

## Research Article

# A Migration Method for Vehicle Mobility Services Based on Road Segmentation Markov Model

Haotian Zhang,<sup>1</sup> Yan Wang<sup>1,2,3</sup>, Jian-Tao Zhou,<sup>1,2,3</sup> Jianwei Gao,<sup>1</sup> and Jing Liu<sup>1,2,3</sup>

<sup>1</sup>Inner Mongolia Engineering Lab of Cloud Computing and Service Software,  
College of Computer Science Inner Mongolia University, China

<sup>2</sup>Inner Mongolia Big Data Laboratory for Discipline Inspection and Supervision, China

<sup>3</sup>National and Local Joint Engineering Research Center of Mongolian Intelligent Information Processing, Hohhot, China

Correspondence should be addressed to Yan Wang; 55234043@qq.com

Received 18 November 2021; Accepted 7 May 2022; Published 27 May 2022

Academic Editor: Bithas Petros

Copyright © 2022 Haotian Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The emergence of edge computing provides a new solution to solve resource-intensive applications, but how to formulate the most efficient service migration strategy according to users' real-time location to ensure users to receive low latency services has become a new challenge. For the vehicle terminals running on the road, mobility of vehicle terminals is a factor that cannot be ignored. Ensuring the continuity of vehicle mobility service is an urgent problem to be solved. For the vehicles running in the roads, we focus on the establishment of service migration model based on Markov decision process according to road zoning, taking the loss and delay caused by service migration as the index to measure the advantages and disadvantages of service migration strategies. Then, a reinforcement learning algorithm is used to transform the service migration model into MDP model, and the measurement index is transformed into the reward function to formulate the service migration strategies for service migration process of the vehicle's travel. Finally, through the simulation experiments on different scenarios, we compared our method with other methods in terms of loss and reward from service migration, which proves the effectiveness of our method.

## 1. Introduction

With the rapid development of mobile Internet technology, a large number of intelligent terminals have emerged, and various resource-intensive applications, such as telematics and virtual reality, have emerged and played an important role in people's life. But the applications often have ultralow latency and massive data processing needs, which brings great challenges to the vehicle intelligent terminals with limited storage, computing, and caching capabilities. Cloud computing, as a new computing model with unlimited supply of resources, can make up for the shortcomings of mobile terminals in terms of computing power and storage capacity. The mobile user terminal transmits the task data to the cloud and uses its powerful computing resources for efficient computing. However, with the development of Internet technology and its integration with the transporta-

tion field, cloud computing can no longer meet the user terminals with mobility and ultralow latency like vehicles and other kinds of transportation. On the one hand, in the traditional cloud computing architecture, the distance between user devices and cloud data centers is generally far away, and the resulting network latency is generally large. On the other hand, because of the mobility of user terminals, the service may be delayed or even interrupted during the process of the terminals moving, which makes it impossible to provide services to users in real time. The mobile edge computing network combines the respective advantages of mobile edge computing (MEC) and mobile cloud computing (MCC) to provide a new solution to the above problems. In the mobile edge computing network, by deploying some delay sensitive applications on the mobile edge computing server, the computing tasks offloaded by the vehicle user terminal only need to be offloaded to a nearby mobile edge

server to obtain the results, which greatly reduces the delay and the data transmission volume of the core network and improves the user's quality of service. In the mobile edge computing network for Internet of vehicles, in order to enable each vehicle user to enjoy the convenience brought by edge cloud computing, we need to deploy MEC servers along the road to complete the coverage of the whole driving road. However, when the location of vehicle terminals in mobile edge computing networks constantly changes, how to solve the mobility management problem to ensure the user's service quality and service continuity is one of the bottlenecks that restrict mobile edge computing networks to further improve the user's service quality.

There are currently three main approaches to mobility management in mobile edge computing networks, including increasing the transmit power of the base station, continuing communication through the backhaul network, and performing service migration. The former two methods are constrained by the cost and communication distance and are not applicable to the scenarios like telematics where the users are far away from the MEC servers and have high mobility. So we solve the service continuity problem in mobile edge computing network by service migration technique. Service migration refers to migrating all/part of the computing tasks in the application to the MEC server for execution, which reduces the energy consumption of mobile devices for executing computing tasks. In order to design an efficient service migration strategy, we need to consider various dynamic factors such as the location of user vehicles, the amount of task data, the processing capacity of MEC server, and the remaining computing resources of MEC server. We offload vehicle tasks to edge computing devices with rich computing power, reduce transmission delay and loss, and improve the quality of service of vehicle users.

In the process of service migration, the loss and delay caused by service migration are important indicators to measure the efficiency of service migration strategy. In previous studies, the formulation of service migration strategy is often considered only from a single perspective. If we only consider the single aspect of loss, we will ignore the impact of delay on users' vehicles. On the contrary, if we only consider low latency, we may ignore the unnecessary loss caused by redundant service migration. In addition, most studies use time as a continuous variable to establish a Markov decision process, and we divide the vehicle driving sections and take the divided sections as variables to establish a Markov decision process. Taking the road section as a variable, the vehicle user location is more directly connected with the Markov decision-making process, which better solves the mobility problem of vehicle users.

However, in the mobile edge computing network, due to the mobility of vehicle terminals, the diversity of service requests, and the variability of regional requests, it is easy to cause problems such as load imbalance of edge servers and service interruption, thus seriously reducing the service quality of users. Although the edge service migration technology can ensure the continuous service for users and balance the workload between edge servers, due to the limited communication capacity, computing resources,

and storage capacity of the mobile edge computing network, how to quickly and efficiently perform edge service migration based on the real-time location information of vehicle terminals has become one of the key issues to be solved in the dynamic service migration of mobile edge computing network.

In this paper, the service migration process of vehicle terminal is modeled as a Markov decision process based on vehicle terminal locations, and a reinforcement learning-based service migration strategy is proposed in conjunction with the actual situation. Taking into account the mobility of vehicle terminals and the real-time nature of vehicle networking tasks, the service migration strategy of mobile edge computing based on vehicle networking is optimized. The main research contents and contributions of this paper include the following three aspects.

- (1) Establish a service migration model. We simulate the real vehicle driving environment and establish the scene model. Then, the driving section of vehicle users is divided into migration section and nonmigration section through the deployment of base stations along the road. The main contribution of this stage is to establish a Markov decision process model with the migration section as a discrete variable, fully consider the mobility of users and the deployment of base stations, and minimize the overhead caused by service migration
- (2) Develop metrics. From the perspective of service providers and vehicle users, loss and delay are taken as the measurement indicators of service migration decision at the same time. It not only considers that frequent service migration will cause unnecessary migration cost and redundant communication overhead but also considers the low delay characteristics of vehicle user tasks
- (3) Service migration strategy based on reinforcement learning. Taking the migration section as a discrete variable, a Markov decision process model is established, which fully considers the mobility of users and the deployment of base stations and minimizes the overhead caused by service migration. The service migration strategy based on reinforcement learning is designed, the state set and action set are defined, the return function of reinforcement learning is defined through measurement indicators, and finally the service migration strategy is solved based on Q-learning algorithm

The overall structure of this paper is as follows: Section 1 expounds the research significance and the key issues to be solved in this paper. Section 2 introduces the related work on service migration. Section 3 introduces the service migration scenario and problem model. Section 4 transforms the problem of service migration into the MDP problem. Section 5 analyzes the performance of the service migration algorithm through experiments. Section 6 summarizes the paper.

## 2. Related Work

In recent years, there have been many studies on service migration for mobile edge computing. It includes optimizing service migration strategies from different perspectives, such as migration performance, migration, and virtual machine migration.

Some studies propose service migration strategies based on Markov decision models. Taleb and Ksentini analyzed the movement patterns of users in a pair network and the transformation relationships between states in the chain and then generalize them to Markov models [1]. Based on this model, the average distance between each user and the optimal MEC can be obtained, and then, the average latency for the user to obtain the corresponding service can be calculated after the virtual machine is moved to the optimal MEC server. Simulation results show that the overall migration of VMs from an edge computing node will consume more energy compared to the partial migration of VMs but reduce the average latency of responding to intelligent applications. Ksentini et al. proposed a new idea to simplify the cellular network from a two-dimensional model to a one-dimensional model, using a continuous-time Markov decision process defined as VM migration policy, which determines the optimal threshold value at the time of VM boot state [2]. Simulation results show that the service migration strategy can obtain the maximum expected gain. Literature [3] solves the problem of service disruption to mobile vehicles due to the limited service range of edge servers by constructing a process gain function of multiparameter Markov decision, which improves the shortcomings of the service migration scheme based on distance alone. But it does not properly model and estimate the cost of the service provider in the migration decision. In the nonoverlapping coverage scenario with fixed user access nodes, Ouyang et al. proposed a Markov approximation algorithm with probability distribution model. It considered the user mobility and the uncertainty problem of service request arrival and then obtained the optimal service migration strategies for different time periods by constructing irreducible Markov chains [4].

Some studies specify service migration policies based on user behavior prediction. Nadembega et al. proposed a service migration prediction algorithm based on user mobility by predicting user's mobile behavior, compromising economic cost and QoS [5]. The experimental results showed that this scheme can obtain lower transmission delay but requires higher economic cost. Literature [6] proposed a dynamic service migration mechanism based on edge cognitive computing (ECC), which can quickly migrate services according to users' behavioral cognition and solve the problem of large fluctuations in service quality under different user behaviors. In literature [7], a mobility-aware edge service migration algorithm was proposed for the problem of load imbalance among edge servers and degradation of user service quality due to dynamic changes of user location in mobile edge computing networks. The optimization problem was modeled as a mixed integer nonlinear programming problem with the objective of minimizing the perceived

delay of user service requests. Then, the delay optimization problem was decoupled into the edge service migration subproblem and the wireless access subproblem based on Lyapunov optimization method. Next, a fast edge decision algorithm was proposed to solve the optimal resource allocation and edge service migration scheme for a given wireless access policy. Finally, an asynchronous best response algorithm was proposed to iterate the optimal wireless access policy. Simulation results showed that the proposed algorithm can reduce the perceived delay of user service requests while ensuring stable service migration cost compared with existing service migration policies. Literature [8] proposed an amortization algorithm and an inert migration algorithm in edge cognitive computing networks based on user location prediction, which can well balance the extra cost spent on performing service migration and the extra overhead caused by not performing service migration. Literature [9] classified the mobility types of user devices into three categories: random mobility, short-term predictable mobility, and fully known mobility, and different service migration strategies based on different mobility types can reduce the overhead of user devices. Literature [10] investigated service migration in MEC by considering the risk of location privacy leakage. The total cost of the system was defined as the combination of migration cost, user-perceived delay, and location privacy leakage risk.

Other studies jointly considered the development of service migration strategies from migration cost, migration overhead, and migration latency. Literature [11] implemented an enhanced service migration model to solve the user proximity problem. The migration cost, transaction cost, and energy consumption associated with the migration process were formalized, and then, the service migration problem was modeled as a complex optimization problem, and then, a deep reinforcement learning was used to approximate the optimal policy. The results showed that the proposed model can estimate the optimal policy with complex computational requirements. A model of intelligent service migration algorithm based on machine learning algorithms was proposed in literature [12]. The algorithm was aimed at optimizing the latency and energy consumption considering the dynamic nature of network bandwidth and the battery power of mobile devices. Simulations showed that the performance of the proposed service migration algorithm outperformed the compared algorithm. In literature [13], a QoS-aware service migration scheme was proposed in an edge-centric cloud network, in which the best target server was selected and service migration was performed considering service migration, network cost, QoS, and resource utilization in the server. The evaluation results showed that the proposed scheme can reduce the migration cost and network traffic, while the required QoS in terms of latency can be guaranteed.

Literature [14] conducts a detailed survey of artificial intelligence and the Internet of vehicles and discusses the problems faced by artificial intelligence to optimize the edge services of the Internet of vehicles. Literature [15] introduces the development background of smart city and briefly defines the concept of smart city. The framework of a smart

city is described according to the given definitions. Finally, various smart algorithms to make cities smarter are discussed and analyzed, along with specific examples. Literature [16] proposes an anonymous blockchain-based charging-connected EV system, which eliminates third-party platforms through blockchain technology and establishes a multiparty security system between EVs and EVSP, an EV charging service provider. The evaluation results show that the system's user anonymity, information authenticity, and system security meet the necessary requirements. There are also some studies in the field of intelligent transportation, which analyze and predict vehicle behavior, speed, and other information by establishing models. In order to better analyze the prediction of driving intention from a bird's eye view, Huang et al. [17] divided typical vehicle intentions into horizontal intention and vertical intention and predicted both intentions based on the historical movement of vehicles and the surrounding environment information based on the urban road scene. Moreover, a new ConvLSTM-based model is proposed for prediction. Experiments show that the new model is more accurate than the baseline LSTM model. Guezaz et al. [18] proposed a new shopping traffic classifier model based on the existing machine learning algorithm, which could classify the events collected in the network. The model includes an input layer, a hidden layer, and an output layer and uses two databases for training and testing. Then, the model is verified mathematically by recognition algorithm. Malek et al. [19] collected real information such as vehicle driving road ID, vehicle actual speed, road type, and battery SoC as data sets through VSimRTI simulator and trained speed prediction models of univariate and multivariate scenarios based on LSTM model. Simulation results show that multivariate models outperform univariate models in short- and long-term forecasting.

In summary, most of the relevant studies have been conducted to develop and optimize the service migration strategy from the user's location or by calculating the cost overhead caused by the migration. There are relatively few studies based on the changes of vehicle terminal location and task real-time performance in telematics. We mainly study the following two issues in the service migration: (1) total energy consumption of different stages of service migration strategies during long-distance driving and (2) the balance of service delay and migration cost. In this paper, by considering the above two aspects, we design a reward function for service migration strategy by reinforcement learning with measurement metrics. We will formulate the service migration problem as a Markov decision process and propose an efficient algorithm to find the best solution that minimizes the total cost in the long run.

### 3. Scenarios and Problem Models of Service Migration

**3.1. Service Migration Scenario.** The framework of service migration under MEC based on telematics is shown in Figure 1, which consists of an edge computing layer and a vehicle layer. The edge computing layer consists of base stations deployed along the roadside and MEC servers

deployed on the base stations. The MEC servers are connected to each other by a wired channel, and when the service is migrated, the source MEC server transmits a running instance of computing task offloaded from the vehicle to the target MEC server via the wired channel. At the same time, each MEC server is connected to the remote cloud to ensure real-time remote service communication. It is known that some kinds of MEC services are running in each MEC computing server to provide MEC services to vehicle terminals. We assume that the target MEC server contains the types of services required by the task. The vehicle terminals and the base stations satisfy the following conditions.

- (1) Each vehicle terminal can be covered by at least one base station
- (2) Only one MEC server is deployed at each base station
- (3) Each vehicle terminal can only be connected to at most one base station at the same time

The coverage area of each base station is limited, and the vehicle terminal initially connects to the nearest base station to offload its application. As the vehicle travels, the distance between the vehicle and the base station increases and its service delay also increases, so the vehicle terminal needs to decide whether to migrate its service. However, frequent service migration is a great waste of limited resources on the MEC server. So we will design a strategy that the cross-coverage of roadside base stations are defined as the migration sections, and the vehicle terminal decides its service migration strategy according to a service migration algorithm when it drives to the cross-coverage area of multiple base stations. For example, if a vehicle terminal requires a real-time driving safety detection service, the service migration strategy is decided based on various information such as the distance between the vehicle and the server and the remaining computing resources on the server when the vehicle drives into the migration sections. In this case, there are two migration decision options: (1) the vehicle terminal chooses to connect to the target roadside unit to be migrated and migrates its services from the source MEC server to the target MEC server, called strategy S1. (2) The vehicle terminal chooses to connect with the roadside unit to be migrated but does not migrate its service from the source MEC server to the target MEC server. The target roadside unit acts as an intermediate node to receive the task data sent by the vehicle terminal and transmits it to the source roadside unit. And then, the result of the completed task processing by the source MEC server is transmitted from the source roadside unit to the target roadside unit and from the target roadside unit to the user. That is, the source MEC server processes the task, and the result is passed from the source roadside unit to the vehicle terminal through the target roadside unit, called strategy S2.

Assume that the service migration can be complete in one migration section, and a vehicle terminal has two choices of migration decisions in every migration section.



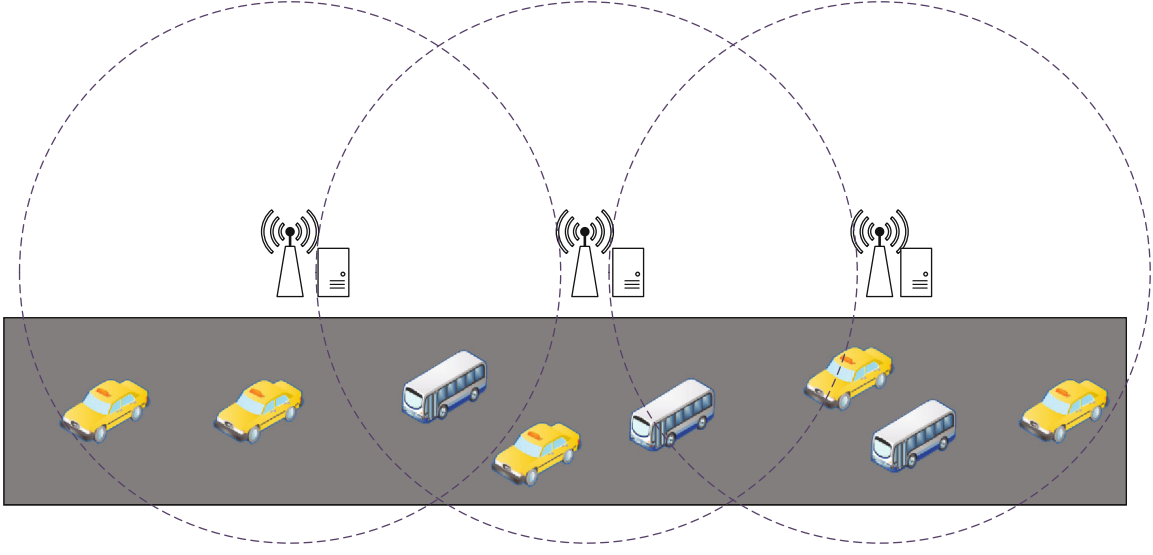


FIGURE 1: Scene model diagram.

Different migration decisions will cause different losses, which are measured by the metrics. And in the process of vehicle driving, the migration decisions at each stage are influenced by each other. The goal of our paper is to find a service migration strategy that minimizes the loss of service migration during the entire journey of the vehicle.

**3.2. Problem Model.** A set of base stations,  $R = \{r_1, r_2, \dots, r_N\}$ , is deployed along roadside, and each base station is connected to a MEC server. All servers have the same execution capability and a limited number of computational resource units (CRUs). A set of vehicle terminals,  $U = \{u_1, u_2, \dots, u_M\}$ , is collected by the base station and transmitted to the MEC server through the base station to which it is connected. After the edge server processes the unloading task, it returns the result to the vehicle user through the base station.

We assume that the service migration can be completed in every migration section. For one migration section  $L$ ,  $s(L)$  is a set of roadside unit that a vehicle terminal on  $L$  can detect the signal of base stations. The distance between vehicle terminal  $u$  and roadside unit  $r$  is denoted by  $\text{Dist}_{u,r}$ , and the distance between roadside unit  $u_i$  and  $u_j$  is represented by  $p_{i,j}$ . The total cost of migration service includes three parts: the cost for migrating a service instance from one edge server to another edge server, the cost of transmission between the roadside unit connected to the vehicle terminal and the roadside unit where its service is located, and the cost of transmission from the user to the roadside unit to which it is directly connected. As shown in Figure 2, the latter two costs of transmission are mainly determined by  $p_{i,j}$  and  $\text{Dist}_{u,r}$ , respectively, which are calculated by exponential cost function according to the existing work [5]. The calculation formula of each part cost in this paper is given below.

- (1) An example of service migration is a virtual machine or container running vehicle user equipment tasks.

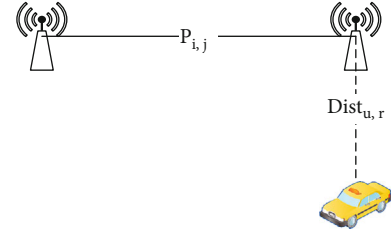


FIGURE 2: Migration communication diagram.

Its data size is different from the task size uploaded to the server and is generally greater than the task data. The cost for migrating a service instance from one edge server to another edge server can be expressed as the loss caused by creating a service instance on the target server, formally

$$\text{Cost}W = \alpha_w p_{i,j} + C, \quad (1)$$

where parameter  $\alpha_w > 0$ ,  $p_{i,j}$  are the distance between server  $i$  and server  $j$ . Because the data packets of computing tasks unloaded by user equipment are generally small, and the delay generated in the transmission process is much less than that generated in the queuing process of data packets, some studies use the network hops between MEC servers to represent the distance between two MEC servers.  $C$  is the cost of creating a vehicle user service instance on the target server

- (2) Because there are various different factors in the actual scene, the simple overhead distance proportional model may not represent all cases. Therefore, in this chapter, the overhead function is defined in the form of exponential model. In addition to being closer to the actual situation, the function form can also obtain a better solution more easily [11]. The

cost of transmission between the roadside unit connected to the vehicle terminal and the roadside unit where its service is located can be calculated by

$$\text{Cost}L_{d1}(x) = \begin{cases} 0, & \text{if } x = 0, \\ \beta_c + \beta_l \sigma^x, & \text{if } x > 0, \end{cases} \quad (2)$$

where  $\beta_c$ ,  $\beta_l$ , and  $\sigma$  are real-valued parameters and  $x = p_{i,j}$

- (3) The cost of transmission from the user to the roadside unit to which it is directly connected can be calculated by

$$\text{Cost}L_{d2}(x) = \begin{cases} 0, & \text{if } x = 0, \\ \delta_c + \delta_l \theta^x, & \text{if } x > 0, \end{cases} \quad (3)$$

where  $\delta_c$ ,  $\delta_l$ , and  $\theta$  are real-valued parameters and  $x = \text{Dist}_{u,r}$

- (4) The total cost of migration service on one road section is shown in

$$C_a(s(1)) = \text{Cost}W + \text{Cost}L_{d1}(p_{i,j}) + \text{Cost}L_{d2}(\text{Dist}_{u,r}) \quad (4)$$

The service delay caused by the terminal from task offloading to getting its execution result includes mainly the following parts: transmission delay, execution delay, and migration delay. The calculation formula of each part cost in this paper is given below:

- (1) The transmission delay includes the transmission delay of task data between vehicle terminals and their connected MEC servers and the transmission delay of task instance between MEC servers. The longer the distance between them, the greater the transmission delay, and the larger the amount of task data or instance, the greater the transmission delay. The calculation formula is as follows:

$$T_{\text{transport}}(x) = D_{\text{task}} * x, \quad (5)$$

where  $D_{\text{task}}$  represents the amount of task data and  $x$  represents the distance  $\text{Dist}_{u,r}$  between the vehicle and the MEC server or the distance  $p_{i,j}$  between the roadside units

- (2) The execution time of a task on a MEC server is related to the amount of data of the task and the computational complexity of the task. Different tasks have different computational complexity. In this paper, the computational complexity of tasks unloaded by vehicle users is defined as  $O(n)$ , and the complexity of all tasks is summarized as follows:

$$O(n) = \begin{cases} O(1), \\ O(n), \\ O(n^2), \\ O(\log n), \\ O(n \log n) \end{cases} \quad (6)$$

The calculation formula of execution delay is as follows:

$$T_{\text{execute}} = \frac{D_{\text{task}} * O(n)}{P_s}, \quad (7)$$

where  $D_{\text{task}}$  represents the amount of task data and  $P_s$  represents the computing power of MEC server.

- (3) The wired channel transmission of data between MEC servers will produce a certain migration delay, which is directly proportional to the distance between servers. In addition, creating a service instance on the target MEC server will also cause delay. Therefore, the migration delay caused by users when selecting service migration consists of two parts.

$$T_{\text{migration}} = \alpha_{Tm} * p_{i,j} + R, \quad (8)$$

where  $\alpha_{Tm} > 0$ ,  $p_{i,j}$  is the distance between MEC server  $i$  and MEC server  $j$ .  $R$  represents the delay caused by creating a task instance on the target MEC server, which is a constant greater than 0

- (4) The total delay can be expressed by

$$T_{\text{sum}} = \begin{cases} T_{\text{transport}} + T_{\text{execute}} + T_{\text{migration}}, & a = 0, \\ T_{\text{transport}} + T_{\text{execute}}, & a = 1. \end{cases} \quad (9)$$

$a = 0$  means to adopt service migration strategy S1, and  $a = 1$  means to adopt service migration strategy S2

- (5) The loss of migration can be defined within a migration section as follows:

$$\text{Loss} = \mu_1 * T_{\text{sum}} + \mu_2 * C_a(s(1)), \quad (10)$$

where  $\mu_1$  and  $\mu_2$  are correction coefficients used to adjust the weight of the terminal's loss value and the MEC system's migration cost

During the long journey of the vehicle, it usually passes through several different roadside units. When the vehicle arrives at a migration section, the vehicle terminal must choose a service migration strategy. Therefore, our optimization objective in this paper is to minimize the total cost of multiple service migrations during vehicle driving, as shown in

$$\min \sum_{i=1}^n \text{Loss}(t_{\text{sum}}, \text{Ca}(s(t))) \quad (11)$$

$$s.t. \sum_{i=1}^n \text{act}_{u,t} * D_i^j < \theta_j, \text{act}_{u,t} \in [0, 1] \quad (12)$$

Formula (12) gives the constraints that should be satisfied to solve (11), where  $\theta_j$  represents the memory capacity threshold of the edge cloud computing node,  $D_i^j$  represents the physical server memory consumed by the  $i$ -th user requesting the  $j$  class service.

#### 4. Service Migration Based on Road Segmentation Markov Model

In the process of vehicle running, the nature of service policy selection process of the vehicle terminal every time when it reaches the intersection coverage area of roadside units is a state transition process. Therefore, we choose Markov decision process to help us solve the problem of service migration strategy, and service migration in every migration section is regarded as a discrete variable. For different vehicle terminal, different migration sections have different states. In this section, the problem of edge computing service migration in the Internet of vehicles is modeled by Markov decision process (MDP) and solved by Q-learning algorithm.

In the MDP modeling process of service migration decision-making by Internet of vehicles, different from most research on solving service migration based on reinforcement learning, we do not use time flow model for MDP modeling. In the road network model driven by vehicle terminals, the road is segmented and the MDP model is established by using the migration segmentation as discrete variables. Through road segmentation, the mobility of vehicles is considered, and the road network model is formed by combining the actual situation of the road. In addition to considering the mobility of vehicle users, the computing resources on the server and the distance between vehicle users and base stations are also considered. The base station set that vehicle terminals can detect signals on the migration section is defined as the system state; the migration strategies that vehicle terminals can adopt are defined as the action set. At the same time, the above indicators of cost and delay are used as the reward function to ensure that vehicle terminals can obtain real-time services while facing the frequent service migration.

After determining the state set, action set, and reward function of MDP, this paper uses Q-learning algorithm to solve the edge computing service migration strategy based on Internet of vehicles. The Q-learning algorithm is a value-based algorithm in the reinforcement learning. The Q-learning algorithm first initializes a Q-table. The rows of the table represent the system state set and the columns represent the system action set in the MDP model. The value of Q-table is the return value of an action in the state set, and then, the Q-table is updated by continuously selecting and executing actions to obtain environmental feedback, and finally a trained Q-table is obtained.

##### 4.1. MDP Model

**4.1.1. System Status.** The computing task offloaded by the vehicle terminal to the edge server is described by a triple  $t = (D_{\text{task}}, T_{dl}, T_{re})$ , where  $D_{\text{task}}$  represents the data volume of the task,  $T_{dl}$  represents the latest completion time of the task, and  $T_{re}$  represents the computing resources occupied by the task. The computing resources of MEC servers deployed on the base station are also limited, and all MEC servers have the same computing resources. Therefore, given the computing task and MEC server, we can clearly know the computing resource demands of the task and the remaining available computing resources of the MEC server.

Multiple base stations are deployed on the upper part of the road section where the vehicle terminal travels. Only a unique MEC server is deployed on each base station, and the distance between the base stations is equal. When the vehicle terminal moves to the migration section, the signals of different base stations can be detected. At this time, the distance between the vehicle and the base station is  $\text{Dist}_{u,r}$ , and the distance between base station  $i$  and base station  $j$  is  $p_{i,j}$ .

The service migration strategy is formulated from the two aspects of migration cost and migration delay. When the service migration is carried out frequently to meet the real-time service requirements of vehicle terminals, additional migration costs will be incurred. At the same time, when the distance between the vehicle and the base station exceeds the service delay requirements of the terminal, the service quality of the vehicle will be affected. Moreover, the computing resources of the MEC server are limited. When the computing resources of the target MEC server to which the vehicle terminal intends to migrate its service do not support service migration, it is necessary to consider other MEC servers or other service migration strategies. Therefore, when implementing the service migration decision, it is necessary to comprehensively consider the computing resource demands of the service and the remaining available computing resources of the target server.

**4.1.2. System Action.** The system makes migration decisions based on the current state, and the action set is defined as  $A = \{0, 1\}$ , where 0 means take service migration strategy S1, that is, migrate the service from the MEC server connected to the source base station to the MEC server connected to the target base station. 1 means take service migration strategy S2, that is, not migrate the service instance of the terminal task, but taking the target base station, to which the terminal connected, as an intermediate node between the terminal and the source base station for communication. The action taken in migration section is noted as  $a_k (a_k \in A)$ .

**4.1.3. Reward Function.** The reward function is based on the migration loss. The smaller the migration cost and communication delay, the greater the reward for the actions taken by the system. The reward function is defined as follows:

$$r(s_l, a_l) = M - \omega_1 * T_{\text{sum}} - \omega_2 * C_a(s(l)). \quad (13)$$

Among them,  $\mu_1$  and  $\mu_2$  are the correction factor,  $\mu_1 + \mu_2 = 1$ .  $M$  is a positive constant to ensure that the reward function will not be negative.  $\mu_1 > \mu_2$  means more emphasis on real-time services for vehicle users;  $\mu_1 < \mu_2$  represents more emphasis on low migration costs.

**4.2. Service Migration Algorithm.** Based on the above MDP model, the service migration decision strategy designed in this chapter is solved by the improved Q-learning algorithm. Q-learning algorithm is a value iterative algorithm in reinforcement learning.  $Q(s, a)$  is the expectation that action  $a$  can get benefit in state  $s$ . Environment will feedback corresponding reward according to agent's action. So the main idea of the algorithm is to construct a Q-table to store Q value with actions and then select an action that can get the most benefit based on Q value. The Q function is updated as follows:

$$Q(s, a) = (1 - \eta) * Q(s, a) + \eta [r + \gamma \max_{a'} Q(s', a)], \quad (14)$$

where  $s$  represents the current system state,  $s'$  represents the next state, and  $a$  represents the current action.  $R$  indicates immediate reward for taking actions at state  $s$ .  $\gamma$  is a discount factor, which varies from 0 to 1, indicating the importance of future returns. The greater the value of  $\gamma$ , the greater the value of future returns.  $\eta$  is the learning rate, which determines the impact of the training structure before each iteration update.

The training result of Q-learning algorithm is usually expressed as a Q-table. As shown in Table 1, the rows of Q-table represent the states, the columns represent the actions, and each element in the table represents  $Q(s, a)$ , which represents the return of the system taking action  $a$  under the state  $s$ .

In this section, according to the MDP model, the size of the action set is 2, which are 0 and 1, respectively. Therefore, there are only two columns of Q-table. The first column is Q value of the service migration policy S1 in each state, and the second column is Q value of the service migration policy S2 in each state. The result of using the Q-learning algorithm to make service migration decisions is a Q-table shown in Table 1. At each migration section, the vehicle terminal selects the action with the highest value in the current status of the Q-table to execute.

One of the issues during selecting actions is to consider how to balance the relationship between utilization and exploration. Utilization refers to how to make decisions based on the currently known information, while exploration refers to making other decisions to collect more information and help subsequent decisions. Reasonably balancing the relationship between utilization and exploration has a very important impact on the learning ability of the system. On the one hand, choosing an action only with the information you have may result in some specific states of the actions not being explored, and the selected action is not optimal. On the other hand, too much random exploration can lead to too long learning time to converge. In this section, the greedy strategy is used to solve this problem,

TABLE 1: Q-table.

| $Q(s, a)$ | $a_0$         | $a_1$         |
|-----------|---------------|---------------|
| $s_0$     | $Q(s_0, a_0)$ | $Q(s_0, a_1)$ |
| $s_1$     | $Q(s_1, a_0)$ | $Q(s_1, a_1)$ |
| ...       | ...           | ...           |
| $s_N$     | $Q(s_N, a_0)$ | $Q(s_N, a_1)$ |

which randomly selects actions at each step with probability, while the remaining 1-probability directly selects the action with the current maximum return based on the Q-table. In the specific implementation, with the training going on, the randomness will gradually decrease; that is, the proportion of exploration will become smaller and the proportion of utilization will become larger and larger.

## 5. Experimental Results and Analysis

**5.1. Simulation Parameter Settings.** In order to evaluate the performance of the service migration decision algorithm proposed in our paper, we perform lots of simulation experiments. This article uses IDEA as the simulation software. Windows version is Win7, and RAM is 12 GB. In practice, base stations deployed along the roadside cover a range of several hundred meters, so we simulated the service migration process of vehicle users in a 5000-meter section. A roadside unit is deployed every 400 meters along the road segment and connected to the MEC server, and the coverage of the roadside unit is achieved. The radius is 700 meters. Each MEC server is assigned 500 CRUs, and the number of CRUs required for each type of service is 5-15. Vehicles travel on the road at random speed  $v$ . The simulation parameters are set as shown in Table 2.

**5.2. Performance Analysis of Service Migration Algorithms.** While verifying the effectiveness of the proposed Q-learning algorithm, we compared it with the following service migration schemes. Among them, the data volume of all algorithms < task data volume and service data volume, so the loss and delay caused by the algorithm are ignored in this paper, which are briefly described below:

- (1) DSP [20]: this solution transforms the problem of service migration into the problem of shortest path solution. Each edge of an adjacent time slice represents a possible combination of service migration. The weight of the edge is the loss of the migration scheme. The algorithm determines the service migration scheme by looking for the sequence of service placements with the lowest average loss over a period of time. This scheme is one of the most classic service migration schemes at present. In order to make the comparison experiment more scientific and persuasive, we take the set of base stations where the vehicle user can detect the signal as the DSP algorithm service migration combination and use the reward value function as the weight value of each



```

Input: System state set  $S$ , System action set  $A$ 
Output: Q-table
1. Initialize Q-table
2. for episode:
3.   Initialization state  $s$ 
4.   for episode <  $thre$  episode++
5.     Read the value from the Q-table and select action  $a$  with the  $\epsilon$ -greedy policy
6.     Get a new state from action  $a$ , get instant reward through reward function  $r$ 
7.     Update Q-table
8.     Update status
9.   end for
10. end for

```

ALGORITHM 1: Service migration algorithm based on Q-learning (MS-Q).

TABLE 2: Parameter setting.

| Parameter name                      | Parameter meaning   | Parameter value     |
|-------------------------------------|---|---------------------|
| $\epsilon$                          | Use the $\epsilon$ -greed policy to get actions                             | 0.8                 |
| $\eta$                              | Learning rate   | 0.2                 |
| $\gamma$                            | Attenuation factor which indicates the importance of future rewards         | 0.8                 |
| $t$                                 | Transmission delay parameter  | 0.5                 |
| $\mu_1, \mu_2$                      | Proportion parameters of energy consumption and time delay in loss function | 0.6, 0.4            |
| $I$                                 | Reward parameter  | 15                  |
| $\delta_c, \delta_l$ , and $\theta$ | Parameters in the conversion cost of exponential models                     | -0.4, 0.4, and 1.03 |
| $\alpha w, C$                       | Migration cost parameters   | 0.4, 5              |
| $\alpha T m, R$                     | Migration delay parameter   | 0.6, 6.5            |
| $v$                                 | Range of vehicle speed  | 10-20 m/s           |
| $i$                                 | Number of algorithm iterations  | 2000                |
| $\beta_c, \beta_l$ , and $\sigma$   | Wireless channel transmission conversion cost parameters                    | -0.4, 0.1, and 0.86 |

combination. The service migration combination with the largest reward value is the optimal service migration combination obtained by using the DSP algorithm

- (2) Always: the vehicle user always migrates the off-loaded service to the MEC server deployed by the nearest base station during driving
- (3) Never: the vehicle user will only perform service migration when it leaves the coverage of the currently connected base station; otherwise, it will always use the source MEC server to provide services
- (4) Low-latency strategy [21]: the low-latency strategy takes the user-perceived delay as the migration standard. If the vehicle user migrates the running service instance to the target MEC server, the user's perception can be reduced, and then, the service is migrated
- (5) Low-cost strategy [21]: the low-cost strategy takes the cost of service migration as the migration standard and minimizes the number of service migra-

tions on the basis of satisfying user-perceived delays to reduce the cost of service migration

As you can see from Figure 3, there are different trends in the loss caused by migrating services using four different algorithms. It can also be seen that the loss value obtained by using Q-learning algorithm is lower than other algorithms in each road segment.

The loss of the vehicle terminal during service migration is related to the distance between the vehicle and the base station. Figure 4 shows the different loss values caused by the vehicle terminal using service migration policy S1 at different coordinates per road segment.

The loss value function used to analyze the quality of the service migration algorithm is composed of the energy consumption and the time delay function. The algorithm with the lowest loss does not necessarily have the lowest total energy consumption or time delay. So we analyze the performance of each algorithm by comparing its total energy consumption and total time delay. Figure 5 shows that the Q-learning algorithm service migration strategy results in the lowest total energy consumption and total latency.

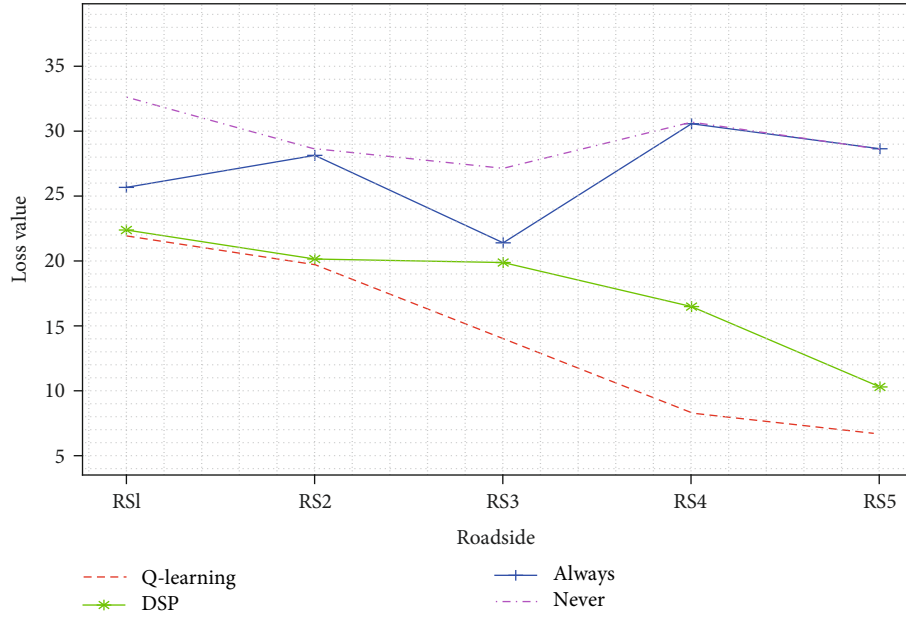


FIGURE 3: Loss values of different algorithms in each segment.

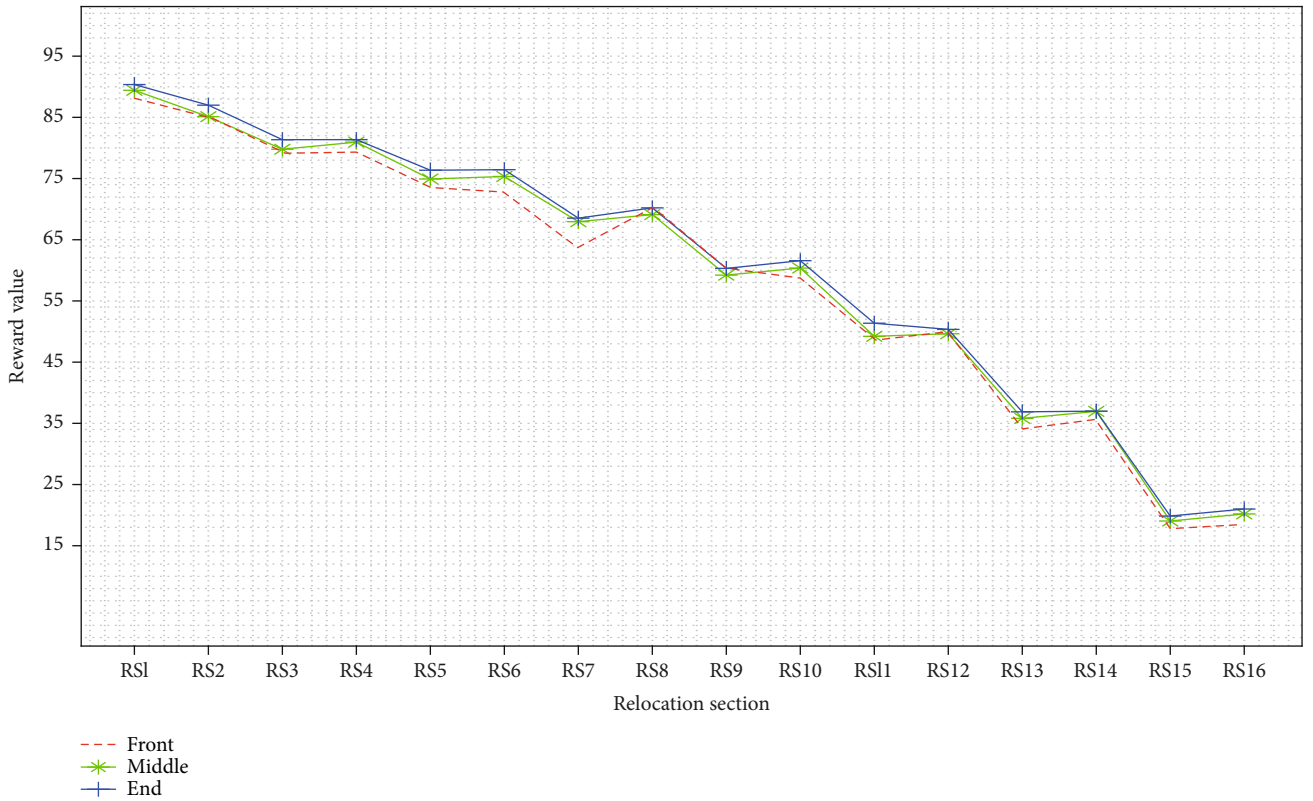


FIGURE 4: Loss values of service migration strategies with different vehicle coordinates.

In edge computing, service migration helps to ensure the continuity of service. However, at the same time, service migration causes some overhead. From Figure 6, we can see that when the number of base stations on the traveled roadway increases, the strategy with low-time cost is basi-

cally not affected by the number of base stations, because it considers the service response time when selecting the migration strategies, thus reducing the number of service migration. The low-latency policy and DSP are more affected by the change in the number of microbase stations.

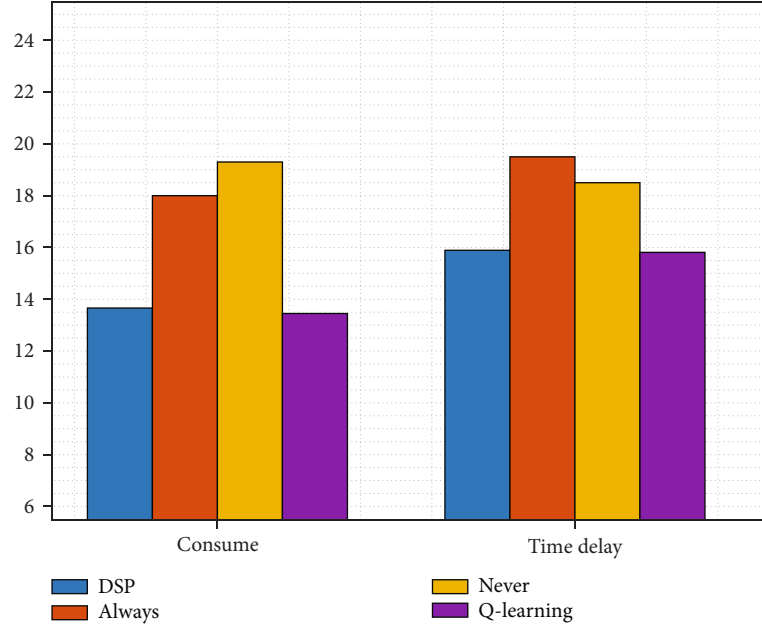


FIGURE 5: Comparison of total energy consumption and total delay for four algorithms.

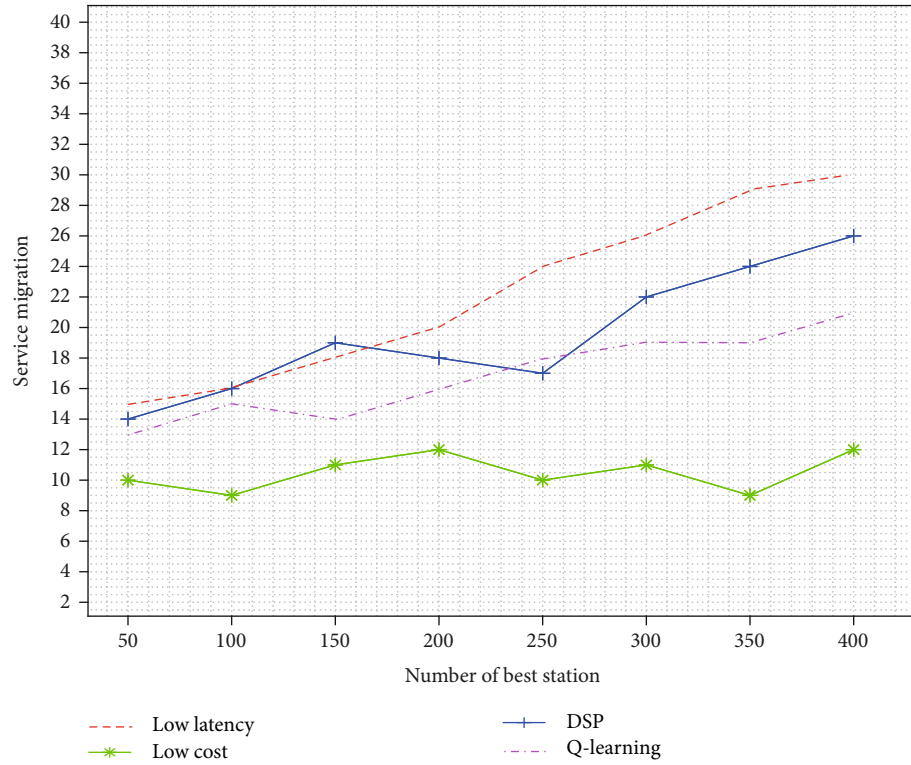


FIGURE 6: The number of service migrations versus the number of base stations.

The Q-learning algorithm is less affected by the change in the number of base stations than the change in the low-latency and DSP.

Figure 7 shows the changes of the Q-learning algorithm and the DSP algorithm as the number of iterations increases. As the number of iterations increases, the reward values of both the Q-learning algorithm and the DSP algorithm con-

verge to their maximum values. In terms of convergence speed, the DSP algorithm is faster than the Q-learning algorithm because the complexity of Q-learning is higher than that of the DSP algorithm and the convergence process is slower. After the number of iterations increases to a certain value, both algorithms stabilize and the reward value of the Q-learning algorithm is larger than that of the DSP

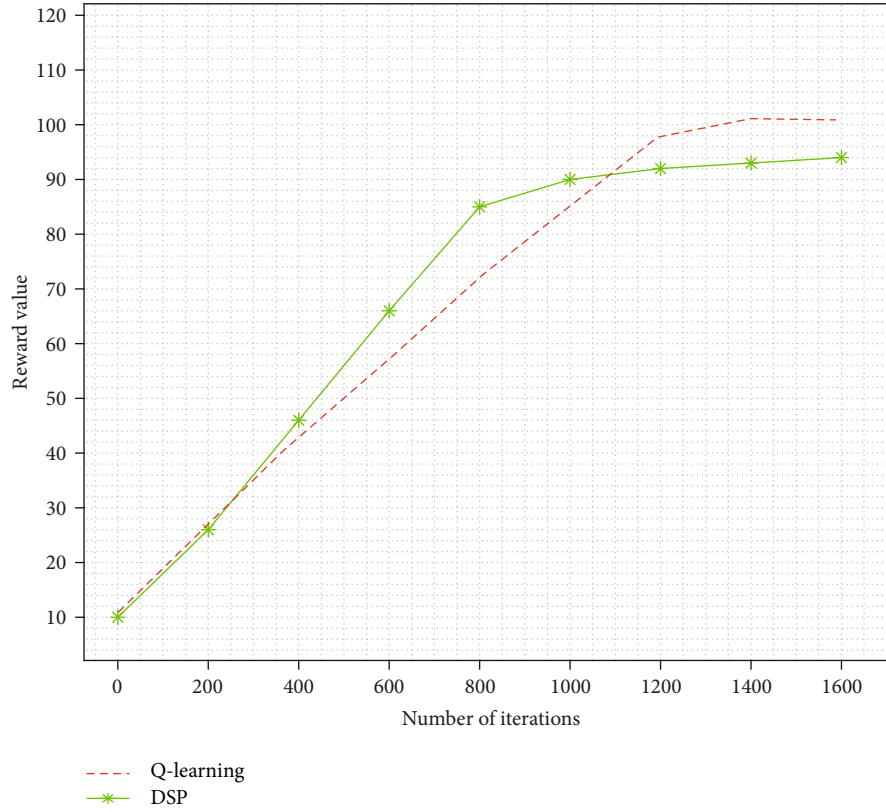


FIGURE 7: The relationship between the return value and the number of iterations of different algorithms.

algorithm. This means that the Q-learning algorithm works better than the DSP algorithm with sufficient iterations.

## 6. Conclusion

In the work of combining telematics with edge computing, we investigate the mobility of vehicle terminals and solve the service migration problem by building a Markov decision process. The road network model is combined with the mobility of vehicle to divide road into migrating segmentation and is modeled using the MDP model. At the same time, the migration cost and communication delay are used as the measurement indexes of the service migration strategy, which can avoid the extra migration overhead caused by the frequent service migration and ensure the real-time performance of vehicle terminals. In the simulation experiments, we compare the Q-learning algorithm with other service migration algorithms in various perspectives to demonstrate the efficiency and advantages of our method. There are still some limitations in this study, which is also the direction of our future work. Firstly, based on the vehicle service migration strategy, straight road scene does not cover all the road scene, such as arc road and angle road, so the direction of future work will integrate all road scene into the road network model and then combining road network model with the user mobility to optimize service migration strategy. Secondly, the task processing process is not refined enough, and the impact of task operation mode and information interaction mode on service migration is not clear.

In the future, the task processing process needs to be refined, and a measurement index is formed to make the service migration strategy more accurate.

## Data Availability

Experimental data and algorithms related to this paper will be shared in the public space.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported in part by the Natural Science Foundation of China under Grant 62162047, Natural Science Foundation of Inner Mongolia under Grants 2019ZD15 and 2019MS06029, Inner Mongolia Science and Technology Plan Project under Grants 2021GG0155 and 2019GG372, the Self-Topic of Engineering Research Center of Ecological Big Data, Ministry of Education, the Open-Topic of Inner Mongolia Big Data Laboratory for Discipline Inspection and Supervision (IMDBD2020012 and IMDBD2021014), Inner Mongolia Engineering Laboratory for Cloud Computing and Service Software, Inner Mongolia Key Laboratory of Social Computing and Data Processing, and Inner Mongolia Engineering Lab of Big Data Analysis Technology.



## References

- [1] T. Taleb and A. Ksentini, "An analytical model for follow me cloud," in *2013 IEEE Global Communications Conference (GLOBECOM)*, pp. 1291–1296, Atlanta, Ga, USA, December 2013.
- [2] A. Ksentini, T. Taleb, and M. Chen, "A Markov decision process-based service migration procedure for follow me cloud," in *2014 IEEE International Conference on Communications (ICC)*, pp. 1350–1354, Sydney, Australia, June 2014.
- [3] H. Guo, L. Rui, and Z. Gao, "Dynamic service migration strategy based on multi parameter MDP model in vehicle edge network [J]," *Journal of Communication*, vol. 41, no. 1, pp. 1–14, 2020.
- [4] T. Ouyang, Z. Zhou, and X. Chen, "Follow me at the edge: mobility-aware dynamic service placement for mobile edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2333–2345, 2018.
- [5] A. Nadembega, A. S. Hafid, and R. Brisebois, "mobility prediction model-based service migration procedure for follow me cloud to support QoS and QoE," in *2016 IEEE International Conference on Communications (ICC)*, pp. 1–6, Kuala Lumpur Malaysia, May 2016.
- [6] M. Chen, W. Li, G. Fortino, Y. Hao, L. Hu, and I. Humar, "A dynamic service migration mechanism in edge cognitive computing," *ACM Transactions on Internet Technology*, vol. 19, no. 2, pp. 1–15, 2019.
- [7] W. Dapeng, J. Lv, Z. Li, and R. Wang, "Edge service migration strategy based on mobile awareness," *Journal of Communication*, vol. 41, no. 4, pp. 1–13, 2020.
- [8] Q. Wu, X. Chen, Z. Zhou, and L. Chen, "Mobile social data learning for user-centric location prediction with application in mobile edge service migration," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7737–7747, 2019.
- [9] Y. Ding, C. Liu, K. Li, Z. Tang, and K. Li, "Task offloading and service migration strategies for user equipments with mobility consideration in mobile edge computing," in *2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pp. 176–183, Xiamen, China, December 2019.
- [10] W. Wang, S. Ge, and X. Zhou, "Location-privacy-aware service migration in mobile edge computing," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, Korea (South), May 2020.
- [11] S. W. Park, A. Boukerche, and S. Guan, "A novel deep reinforcement learning based service migration model for mobile edge computing," in *2020 IEEE/ACM 24th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, pp. 1–8, Prague, Czech Republic, September 2020.
- [12] Y. Cheng and X. Li, "A compute-intensive service migration strategy based on deep reinforcement learning algorithm," in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pp. 1385–1388, Chongqing, China, June 2020.
- [13] J. Lee, J. Kim, Y. Tae, and S. Pack, "QoS-aware service migration in edge cloud networks," in *2018 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, pp. 206–212, Jeju, June 2018.
- [14] X. Xu, H. Li, W. Xu, Z. Liu, L. Yao, and F. Dai, "Artificial intelligence for edge service optimization in Internet of vehicles: a survey," *Tsinghua Science and Technology*, vol. 27, no. 2, pp. 270–287, 2022.
- [15] Z. Tong, F. Ye, M. Yan, H. Liu, and S. Basodi, "A survey on algorithms for intelligent computing and smart city applications," *Big Data Mining and Analytics*, vol. 4, no. 3, pp. 155–172, 2021.
- [16] S. Xu, X. Chen, and Y. He, "EVchain: an anonymous blockchain-based system for charging-connected electric vehicles," *Tsinghua Science and Technology*, vol. 26, no. 6, pp. 845–856, 2021.
- [17] H. Huang, Z. Zeng, D. Yao, X. Pei, and Y. Zhang, "Spatial-temporal ConvLSTM for vehicle driving intention prediction," *Tsinghua Science and Technology*, vol. 27, no. 3, pp. 599–609, 2022.
- [18] A. Guezaz, Y. Asimi, M. Azrour, and A. Asimi, "Mathematical validation of proposed machine learning classifier for heterogeneous traffic and anomaly detection," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 18–24, 2021.
- [19] Y. N. Malek, M. Najib, M. Bakhouya, and M. Essaïdi, "Multivariate deep learning approach for electric vehicle speed forecasting," *Big Data Mining and Analytics*, vol. 4, no. 1, pp. 56–64, 2021.
- [20] S. K. Datta, R. P. F. Da Costa, J. Härrä, and C. Bonnet, "Integrating connected vehicles in Internet of Things ecosystems: challenges and solutions," in *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Portugal, June 2016.
- [21] D. Tang, *The Collaborative Management of Handover and Service Migration in Edge Computing*, Beijing University of Posts and Telecommunications, Beijing, 2020.