ISSN 2616-5775 Vol. 6, Issue 4: 34-48, DOI: 10.25236/AJCIS.2023.060405

Multi-Hop Assisted Offloading Strategy Based on Mobile Awareness in the Internet of Vehicles

Zhu Xiaoli[1], Pang Xiaoyan[1,a,*], Gu Kunyuan[1]

[1]*School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, Henan, 454000, China*
[a]*930912812@qq.com*
[*]*Corresponding author*

Abstract: *The rapid development of Internet information technology and automobile economy has spawned a large number of computation-intensive and time-delay sensitive vehicle applications. In the face of high amount of computing data and real-time response requirements of applications, how to improve the utilization of network resources and reduce the computing delay of vehicle applications has become an urgent problem to be solved. Most of the existing research focuses on the problem of vehicle task offloading within the coverage of edge servers, and the idle vehicle resources outside the one-hop range are not fully utilized. In this paper, we formulate a multi-hop offloading scheme for tasks with mobility awareness. Based on the high-speed movement characteristics of vehicles, the real-time location of vehicles and the multi-hop-assisted offloading of tasks are modeled as a Markov decision process, and a multi-hop-assisted task offloading and resource allocation algorithm is proposed to ensure the reliability of each hop link and minimize the computation delay of tasks. Simulation results show that the proposed multi-hop assisted offloading scheme can significantly improve the response delay of the task. When the vehicle speed is high, the proposed scheme can reduce the response delay of the task by at least 17.2% compared with other algorithms (such as local and single-hop offloading algorithm and random multi-hop offloading algorithm).*

Keywords: *Internet of vehicles, Movement awareness, Multi-hop communication, Task offloading, Resource allocation*, *Delay*

1. Introduction

With the popularity of 5G mobile networks and other wireless networks, intelligent connected vehicles have come into reality, which plays an increasingly important role in promoting the construction of intelligent transportation systems and realizing safe and efficient driving [1]. Intelligent vehicles usually utilize their on-board processors and edge infrastructure to make real-time driving decisions, as the transmission delay of tasks uploaded to the cloud is long and prone to security issues [2, 3, 4]. By offloading tasks to the edge servers, vehicular edge computing can effectively reduce the transmission delay of the network backhaul links. At present, the construction of 5G vehicle-road collaborative demonstration application bases has been carried out in many places in China. However, the deployment cost of edge servers is very expensive, and it is difficult to have a high penetration rate [5, 6]. Therefore, it is essential to enhance the utilization of network resources and solve the complex scheduling problem of task computing resources.

With the significant increase in the number of intelligent vehicles and the limitation of edge resources, more and more researches have pay attention to V2V offloading methods. In order to alleviate the problem of mobile data explosion in the vehicular networks, more and more scholars propose to share the vehicle task communication to the remaining neighboring vehicles, so as to reduce the delay or energy cost of the task [7-10]. In [7] and [8], vehicles are regarded as edge computing resources, which can share resources and conduct collaborative computing with edge servers. The authors of [9] and [10] proposed a relay-based task offloading scheme and task allocation algorithm according to the mobility of vehicles to solve the problem of task offloading and resource allocation. Compared with [7] and [8], researches [9] and [10] pay more attention to the impact of vehicle movement on V2V communication and task offloading. However, the above literatures focus on the problem of vehicle task offloading within the coverage of edge servers, and only consider single-hop offloading, which means that the task vehicle can only use the vehicle resources within its one-hop neighboring range, and it is difficult to use other idle

ISSN 2616-5775 Vol. 6, Issue 4: 34-48, DOI: 10.25236/AJCIS.2023.060405

resources.

Different from the above work, this paper proposes a mobility-aware V2V/V2R multi-hop assisted offloading scheme to address the problem that the vehicles outside the coverage range of the edge server cannot effectively utilize the computing resources of vehicles or the edge server outside the single-hop communication range. Specifically, the main contributions of this paper are as follows:

(1) We model the interaction process between the real-time state of the vehicle and the multi-hop offloading of the task as a Markov process, and establish a V2V/V2R multi-hop offloading problem model with the goal of minimizing the delay.

(2) We propose a multi-hop assisted task offloading and resource allocation algorithm. Firstly, the packet loss rate of multi-hop transmission is calculated, and the greedy algorithm is used to solve the multi-hop path search problem. On this basis, a resource allocation algorithm based on priority time reaching and fair proportional allocation is designed to solve the resource allocation problem of edge server.

(3) By designing simulation experiments and analyzing the experimental results, the effectiveness and advantages of the proposed algorithm are verified.

2. Related Work

In this section, we summarize the existing work related to task offloading in mobile edge computing, especially the offloading researches with V2V communication mode.

Compared with the abundant cloud resources, the computing resources of edge servers are relatively limited, coupled with their expensive deployment costs, more and more research has begun to focus on the aggregation and utilization of vehicle computing resources [11-15]. In [11], Wang et al. integrated and clustered idle vehicle resources by analyzing the service capability of vehicles within the coverage range of RSUs. They proposed an online VEC resource scheduling algorithm based on imitation learning, which uses each RSU to maintain the information within its cluster and reduce the communication overhead between vehicles and edge servers. Similar to research [11], the work of [12] proposed a cluster-based cooperative task allocation scheme for C-V2X networks. In order to explore the collaboration between MEC servers and vehicles with idle computing resources, Bute et al. [12] used vehicle-to-vehicle (V2V) communication and vehicle-to-cellular network (V2N) communication to group vehicles and cellular communication base stations into clusters, and developed an improved matching algorithm to improve link reliability between nodes and achieve effective task allocation. The authors of [13], [14] and [15] respectively realized the utilization of vehicle resources based on software-defined network, digital twin technology and block chain technology. In [13], Luo et al. designed a software-defined collaborative data sharing architecture based on 5G-VANET, which uses SDN controller to collect contextual information and share cooperative data between adjacent vehicles. The work of [14] combined the digital twin technology and artificial intelligence into the design of vehicle edge computing network, and concentrated the use of potential edge service matching through edge image evaluation. To promote collaborative computation offloading between vehicles and ensure the security of computation offloading between vehicles, the authors of [15] proposed to use block chain technology to realize efficient data sharing between vehicles. They designed a new data sharing mechanism to improve the efficiency of data sharing and prevent malicious attacks.

High speed movement of vehicles is one of the most significant characteristics of Internet of vehicles. The mobility of vehicles can greatly affect the quality of communication links and interrupt the offloading transmission process of tasks. Therefore, vehicle mobility and communication delay have always been important factors to be considered in vehicle computing offloading [16-20]. In order to cope with the rapid change of network structure caused by vehicle movement, Zhang et al. [16] described the task offloading of vehicles as a multi-armed slot machine problem, and proposed a distributed online decision algorithm. Based on long-term energy consumption constraints, Tang et al. [17] adopted Lyapunov optimization technique and two-stage greedy heuristic algorithm to obtain the approximate optimal solution of task offloading decision. In [18], the authors considering the computation offloading problem for mobile vehicle users in a heterogeneous VEC scenario, and they developed a network relay selection mechanism based on vehicle dwell time to address the network and base station selection problem in a rapidly changing in-vehicle environment. The authors of [19] proposed a task partition migration and offloading method based on multi-access edge computing (MEC) environment to balance the load of vehicles and edge services and improve the throughput and response delay of MEC system. Aiming at the problem of frequent link disconnection and connection caused by fast movement of network nodes, the authors of [20] proposed a bee colony based task allocation algorithm for vehicle edge computing to reliably reduce the execution time of applications in vehicle edge computing systems.

In VEC systems, the movement of vehicles will bring a certain burden to the offloading of tasks. However, it may also give some vehicles the advantage of being located closer to the edge servers, which will bring benefits for the timely processing of tasks. In [21], the authors exploit vehicle motion on urban roads to create vehicle edges and propose a distributed task orchestration framework (DTOF) to support a vehicle-to-vehicle based lightweight task orchestration scheme. The authors of [22] used machine learning algorithms to predict the number of users at each base station, combined the mobility of users with spatial correlation, and proposed an edge controller based cellular network architecture. The authors of [23] comprehensively considered vehicle speed perception, task offloading and resource allocation, established a vehicle utility function based on task execution cost and benefit, and proposed a joint optimization method of task offloading and resource allocation based on Multi-Agent Deep Deterministic Policy Gradient (JORA-MADDPG).In [24], the authors proposed a novel VEC multi-hop task offloading scheme based on the mobility analysis of vehicles. The task vehicle can simultaneously use the vehicle resources in its one-hop range or multi-hop range for task computation. However, their research focused on the collaboration between vehicles and does not consider the computing resource allocation of edge servers.

Inspired by [24], this article from another new beneficial perspective to look at the movement of the vehicle. We consider a vehicular edge computing scenario that the task vehicle is outside the scope of edge services. Due to the mobility of vehicles in the scene, the task vehicle can migrate the task to other service vehicles with richer resources or within the coverage of edge servers through multi-hop transmission to optimize its own offloading delay. Different from [24], except for the multi-hop transmission between vehicles, we also consider the possibility of task vehicle-service vehicle-edge server link. The objective of this paper is to make full use of V2V and V2R communication resources in vehicle networks and minimize task response delay.

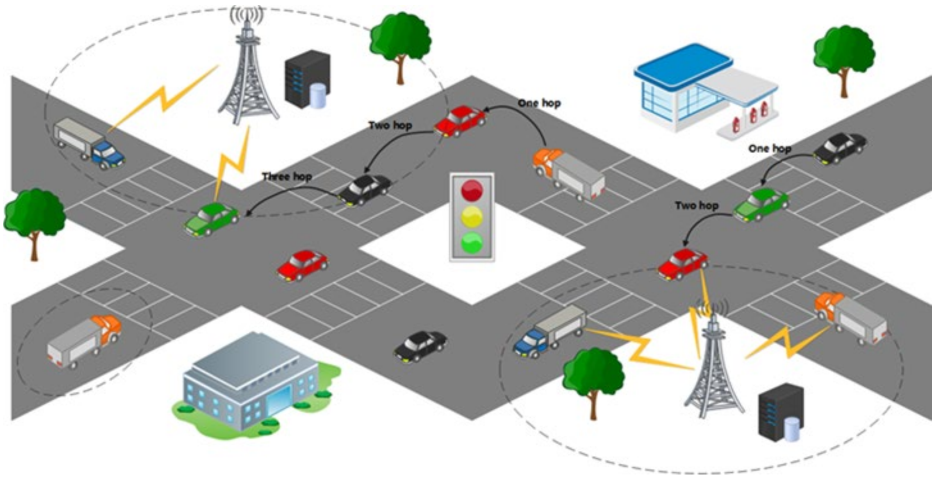


Figure 1: Multi-hop assisted Internet of vehicles computing offloading scenario.

3. System Model

In this paper, we consider a multi-hop assisted urban IoV scenario as shown in *Figure 1*, in which multiple edge servers are set, but the coverage is limited. Vehicles equipped with wireless interfaces and on-board data processing unit, capable of supporting V2R with V2V two types of communication. According to whether or not within the scope of cover is located in the edge server, road vehicles respectively for consumers (CV) and the provider of the vehicle (RV) two types, expressed in set $CV = \{cv_1, cv_2, cv_3 ..., cv_m\}$ and set $RV = \{rv_1, rv_2, rv_3 ..., rv_n\}$ respectively, the set of edge nodes is $R = \{rsu_1, rsu_2, rsu_3 ..., rsu_r\}$. When the consumer vehicle does not enter the communication range of the MEC server, it can carry out multi-hop offloading selectively, use the computing resources of nearby vehicles for task calculation, or use edge servers for calculation after multiple offloading and forwarding through nearby vehicles, so as to maximize the use of existing resources in the network.

ISSN 2616-5775 Vol. 6, Issue 4: 34-48, DOI: 10.25236/AJCIS.2023.060405

3.1. Vehicle Location Model

The position of the vehicle in the scene varies over time, while the edge RSUs are at fixed positions. Mobility of the vehicle will affect the transmission performance of V2V and V2R link. Considering the unsustainability of communication links between vehicles driving in the opposite direction, this paper assumes that the driving direction of consumer and server vehicles in the scenario is the same. The positions of CV, RV, and RSU at a certain time point are shown in *Figure 2*.

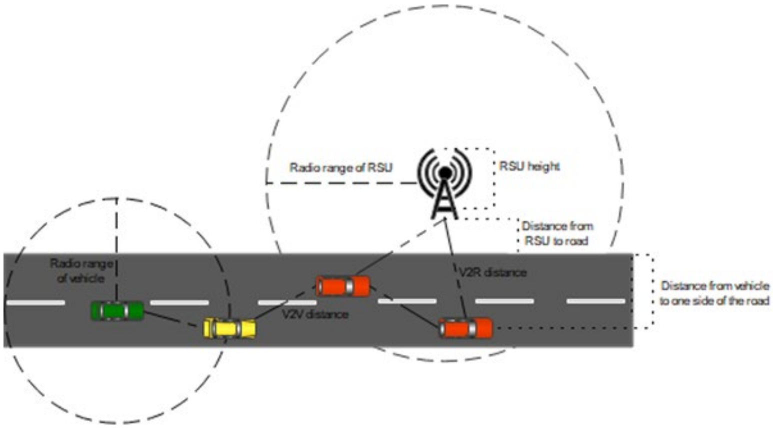


Figure 2: Diagram of the relative positions of vehicle to vehicle and vehicle to edge server.

Let r_{-RSU} denote the RSU coverage radius, h_{-RSU} denote the RSU antenna height, h_{-Veh} denote the vehicle antenna height, and d_{rr} denote the distance from the RSU to the road. According to the spatial distance formula, the distance between the vehicle and the RSU at each time d_{vr} is obtained as follows:

$$d_{vr}=\sqrt{{d_{rr}}^2+(h_{\_RSU}-h_{\_Veh})^2+(s.t)^2} \tag{1}$$

Where s represents the speed of the vehicle, and $s \cdot t$ is the distance of the vehicle from its position at time $t=0$ to its current position.

Similarly, the relative positions between vehicles at different times d_{vv} are:

$$d_{vv}=\sqrt{[(O_{vc}-O_{vr})+(s_c-s_r)t]^2} \tag{2}$$

Where O_{vc} and O_{vr} respectively denote the starting position coordinates of the consumer and service vehicle, s_c and s_r respectively denote their respective speeds.

3.2. Communication Model

Wireless communication has a great impact on the quality of edge-assisted intelligent driving. It is assumed that the consumer vehicle selects only one of the two communication modes V2R and V2V at a time for task offloading and computation.V2R and V2V links use different frequency bands to avoid interference with each other [25, 30].

(1)V2V communication

When the consumer vehicle cv_m is within the communication range of the service vehicle rv_n, the communication rate between vehicles R_t^{V2V} can be calculated by the following equation:

$$R_t^{V2V}=b_{V2V}^t \log_2(1+\frac{p*g_{V2V}}{N_0 b_{V2V}})(t) \tag{3}$$

Where p is transmission power, b_{V2V} is channel bandwidth, g_{V2V} is channel gain between CV and RV, and N_0 is the noise power density.

(2)V2R communication

The V2R communication link is based on Orthogonal Frequency Division Multiplexing (OFDMA) technology [26]. It is assumed that there are K orthogonal resource blocks without interference, and the bandwidth of each resource block is W . For each RSU, resource blocks are allocated equally to the relevant vehicles. When the vehicle is located within the coverage of the RSU, let R_t^{V2V} denote the communication rate between the vehicle and the RSU, and have:

$$R_t^{V2R} = W\lfloor \frac{K}{E_t} \rfloor \log_2(1+\frac{p*gl_n^{\alpha}}{N_0})(t) \tag{4}$$

Where p is the transmission power, $\lfloor \cdot \rfloor$ is a downward rounding function, E_t is the number of vehicles associated with the RSU, g represents the channel power gain, l_n is the distance from the vehicle to its associated RSU, α is the path loss factor, and N_0 is the noise power.

3.3. Time Delay Model

The computation delay of a task includes three types of delay, namely, the transmission time of each hop, the computation time, and the result collection time. Since the output result of task processing is small, the result collection time is ignored and the transmission time and computation time of the task are mainly considered [10, 12]. According to the calculation position of the task, it is divided into vehicle calculation model and server calculation model.

(1) Vehicle calculation model

The transmission time depends on the transmission rate of the channel and the size of the task. At each hop, the data transmission delay between vehicle and vehicle is as flows:

$$T_{Vi}^{t} = \frac{A_n}{R_t^{V2V}}, \tag{5}$$

Where A_{n} is the data size of the task.

The computation time of a task depends on the computational requirements of the task and the computing power of the serving device. Let C_{n} represent the number of CPU cycles required to process one bit of data, and the time required to complete the task on the volunteer vehicle is shown as follows:

$$T_{V_i}^{comp} = \frac{A_n \times C_n}{f_i^{Veh}} \tag{6}$$

(2) RSU calculation model

The data transmission delay between the RSU and the volunteer vehicle is:

$$T_{Ri}^{t} = \frac{A_n}{R_t^{R2V}} \tag{7}$$

The time required to complete the task computation on the RSU is as follows:

$$T_{R_i}^{comp} = \frac{A_n \times C_n}{F_i^{RSU}} \tag{8}$$

4. Markov Process Transformation and Multi-Hop Offloading Problem Formulation

In this part, we first develop a 2-D Markov process [27-29] to model the interactive process of multi-hop offloading of a moving vehicle and a task, and then formulate the optimization problem with the goal of minimizing the delay.

4.1. Modeling the Interaction Process between of Vehicle Movement and Multi-Hop Offloading

In the process of multi-hop offloading, the number of multi-hop of the vehicle task and the final computing position are uncertain, that is, the consumer vehicle can offload its task to the service vehicle

ISSN 2616-5775 Vol. 6, Issue 4: 34-48, DOI: 10.25236/AJCIS.2023.060405

or the edge server for computing through multi-hop transmission. As shown in *Figure 3*, due to the high-speed movement of the vehicle and the passage of time, the V2V and V2R communication service nodes of the task vehicle may change at an uncertain time, which undoubtedly increases the complexity of the multi-hop offloading process. Considering that the position and speed of the vehicle will affect the time and type transformation of the link between the communication nodes, this paper uses the Markov process model to simulate the interaction process between the vehicle movement and the communication link change in the IoV network.

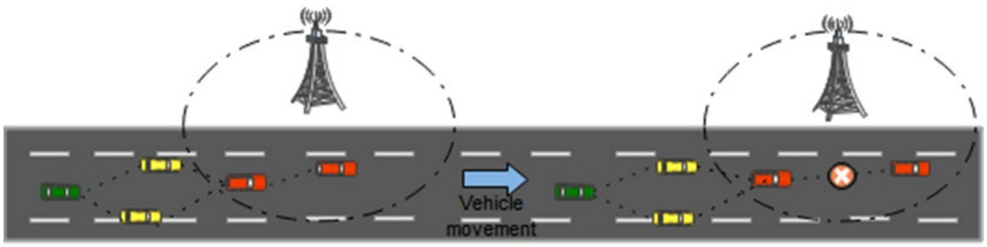


Figure 3: Vehicle movement and communication link disconnection.

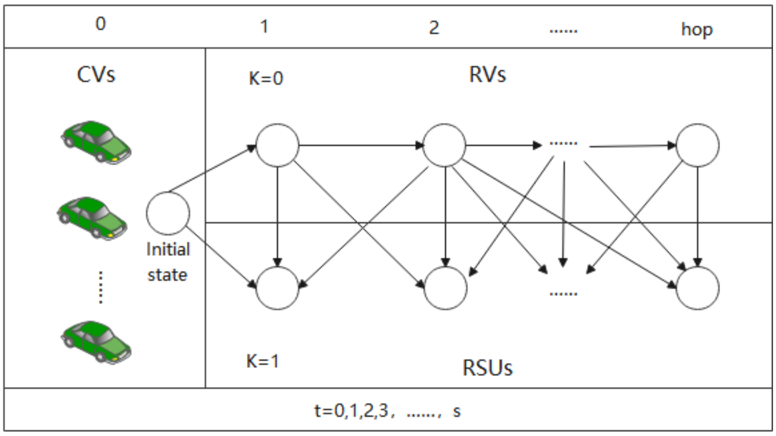


Figure 4: Markov model of vehicle movement and multi-hop unloading process.

As shown in *Figure 4*, let J denote the task set of consumer vehicles in the request queue and K denote the network connection condition between V2V and V2R. Where $K=1$ denotes the V2V connection and $K=0$ denotes the V2R connection. According to the current network performance, this transformation can be occur between two type network connections. Let $p(k, j, hop, n)$ denote the probability that the task j is served by a network connection k and a device node n in each hop, hop represent the number of hops, and n is the vehicle or server number.

Let binary variables α_{CV} represent the offloading or computation status indicator symbol of the task j on the consumer vehicle CV, and the binary variable α_{RV} denote the offloading or computation status indicator symbol of the task j on the service vehicle RV. The specific formula is shown as follows:

$$\alpha_{CV} = \begin{cases} 1 & if\ CV_V\ offloaded \\ 0 & otherwise \end{cases}, \tag{9}$$

$$\alpha_{RV} = \begin{cases} 1 & if\ RV_V\ offloaded \\ 0 & otherwise \end{cases}, \tag{10}$$

Let the binary variable β_{V2R} represent the offloading indicator for the vehicle to offload task j to the base station RSU_r, and the formula is expressed as follows:

$$\beta_{C2R} = \begin{cases} 1 & if\ RSU_r\ offloaded \\ 0 & otherwise \end{cases} \tag{11}$$

Combining the node connection probability $p(k, j, h, n)$ of the task j in each hop, the offloading or computing status indicator α_{RV} on the server vehicle, and the offloading indicator β_{V2R} on the base station,the communication node selection and task offloading situation of each hop can be determined, and then the corresponding delay can be calculated.

4.2. Formulation of Multi-hop Offloading Optimization Problem

Assuming that the total number of tasks is m , and the multi-hop number of task j is hop_j , the total delay after multi-hop transmission and using the resources of service vehicle to complete the calculation is

$$T_j^{Veh} = \sum_0^{hop_j} \frac{A_n}{R_t^{R2V}} + \frac{A_n \times C_n}{f_i^{Veh}} \tag{12}$$

The total delay of the task to complete the calculation by using RSU resources after multi-hop offloading is:

$$T_j^{Veh\_RSU} = \sum_0^{hop_j-1} \frac{A_n}{R_t^{V2V}} + \frac{A_n}{R_t^{V2R}} + \frac{A_n \times C_n}{F_i^{RSU}} \tag{13}$$

Let and denote the number of tasks that complete computation through vehicles and edge servers, respectively. According to formula (12) and Formula (13), the average delay of all tasks is:

$$T_{avg}^{Veh} = \frac{\sum_{j=0}^{m_1} T_j^{Veh} + \sum_{j=0}^{m_2} T_j^{Veh\_RSU}}{m_1 + m_2} \tag{14}$$

$$P: \min_n \sum_{j=0}^{m_1} T_j^{Veh} + \sum_{j=0}^{m_2} T_j^{Veh\_RSU} \tag{15}$$

$$s.t. C1: 0 \le m_1(t) \le m \tag{15a}$$

$$C2: 0 \le m_2 \le m \tag{15b}$$

$$C3: m_1 + m_2 = m \tag{15c}$$

$$C4: T_j^{Veh} \le D_n, \forall j \in m_1 \tag{15d}$$

$$C5: T_j^{Veh\_RSU} \le D_n, \forall j \in m_2 \tag{15e}$$

The constraints in Problem P1 are explained as follows: C1 means that the number of tasks calculated after multi-hop unloading and using vehicle resources to complete tasks does not exceed the total number of tasks; C2 indicates that the number of tasks calculated by edge server resources after multi-hop offloading does not exceed the total number of tasks; C3 indicates that each task can be successfully processed; C4 and C5 respectively indicate that the delay cost after multi-hop offloading and using vehicle resources to complete task computation and after multi-hop offloading and using edge resources to complete task computation does not exceed the constraint delay of the task D_{n} .

5. Multi- hop assisted task offloading and resource allocation algorithm

In this section, we propose a multi-hop assisted task offloading and resource allocation algorithm for optimal task offloading and resource allocation decisions. The algorithm can be divided into greed multiple hops path search and fair resource allocation in two stages.

5.1. Multi- hop path search algorithm based on greedy strategy

Vehicle movement can lead to rapid changes in communication links, and packet loss problems may occur during multi-hop transmission of tasks. The packet loss rate of packet transmission between vehicles is mainly affected by the distance between vehicles and the channel occupancy rate [10, 24, 31]. In order to avoid the problem of communication channel congestion, this paper assumes that the link between each hop pair is a one-to-one connection, so the main cause of packet loss is the change of distance between vehicles. Firstly, the packet loss rate of task multi-hop transmission is defined as follows:

During the time interval of the i th hop Δt, the relative distance variation of the vehicles C_1 and C_2 participating in multi-hop transmission is:

$$D(V_1,V_2)=(\Delta V_2^i-\Delta V_1^i)\times\Delta t=(V_2^i-V_1^i+V_1^{i-1}-V_2^{i-1})\times\Delta t\ , \tag{16}$$

Therefore, the packet loss rate of its data transmission P_{12}^i can be calculated by the following equation:

$$P_{12}^i=1-P\{[(X_2^i-X_1^i)+D(V_1,V_2)]<R\}\ , \tag{17}$$

Where X_1^i and X_2^i represents the initial position of the vehicle C_1 and C_2. During each hop offloading of the task, the packet loss probability of each possible transmission route is firstly calculated, and then the greedy algorithm is used to select the multi-hop relay offloading node with the minimum packet loss probability as the best offloading. Finally, the multi-hop transmission delay of the task is optimized. The time complexity of the greedy algorithm is $O(m\cdot hop\cdot n_h)$. The pseudocode for the algorithm is shown in Algorithm 1.

Algorithm 1:Multi-hop path search algorithm based on greedy strategy

Input:the parameters associated with the task: $Task=\{task_1,task_2,...,task_m\}$,related parameters of the candidate vehicle: $V=\{vehicle_o, vehicle_1, vehicle_2,..., vehicle_n\}$, related parameters of edge sever: $RSU=\{RSU_1,RSU_2,\cdots,RSU_r\}$;

Initialize the starting position and resource status information of all vehicles and edge servers in the scenario;

Output: multi-hop selection scheme: $Result=\{op_1, op_2,..., op_j\}$;

for $t=0$ to $t=T$ do

    for $j=0$ to $j=m$ do

        $h_j=0$;

        while task is not completed do

            update the location and resource status information of vehicles and edge servers in the scene;

            generate information about the nodes to be offloaded and calculate the packet loss probability;

            select the smallest node packet loss probability as the next-hop node, and calculate the corresponding transmission delay;

            while the transmission delay is smaller than the local computing delay do

                $h_j=h_j{+}{+}$;

                If the offloading node is within the coverage of the edge server, add the edge server to the set of next hop nodes, and calculate the respective transmission delay;

                select the node with the lowest transmission delay as the next hop node;

                $h_j=h_j{+}{+}$;

    end for

end for

It is worth noting that within the maximum tolerance delay range of tasks, the criterion for determining whether a hop offloading node of a task is a RSU or a vehicle is the respective transmission delay size.

5.2. Task Resource Allocation Algorithm Based on Priority Time Arrival and Fair Proportion

When the last hop node of multiple tasks is the edge server node, the corresponding computing resources are allocated to the tasks according to the arrival time and number of tasks. The steps of the algorithm are as follows:

Algorithm 2: Task resource allocation algorithm based on priority time arrival and fair proportion

1: **Input**: Task information and its offloading decision set A;

2:Initialize the starting position and resource status information of all vehicles and edge servers in the scene;

3: **Output**: The set of resource allocations for the server F .

4:for $i=1$ to $i=N$ do

5: while $t=i\times\Delta t$ do

6: update the remaining computing resource value of the edge server;

7: obtain all task information and its offloading decision set A during the period $((i-1)\times\Delta t, i\times\Delta t)$, and calculate the number of tasks offloaded to the edge server;

8: allocate the remaining computing resources of the server to the task vehicles in equal proportion, and calculate the resource allocation value of each vehicle;

9: return F ;

10:end for

6. Experiment Simulation and Result Analysis

In this section, we firstly set the main parameters of the experiment, and then set up the vehicle movement and communication link state analysis experiments and delay cost comparison experiments respectively for the proposed multi-hop assisted task offloading scheme, respectively. In the experiment of vehicle moving and communication link state analysis, the reliable link time of V2V and V2R link is analyzed by changing the vehicle moving speed, and the feasibility of multi-hop offloading scheme is determined. In the delay-cost comparison experiment, we compare the proposed multi-hop-assisted task offloading algorithm with several existing classical offloading schemes. By comparing their delay calculation costs, the advantages of the proposed scheme in reducing the task offloading delay were demonstrated. We present the results of the optimization algorithm and evaluate the performance of the proposed algorithm.

6.1. Parameter settings

We built the simulation environment based on open source optimization modeling languages Python and Pyomo [32]. Consider a realistic urban street IoV scenario, where two communication base stations are equidistantly distributed near a street with a length of 3km, the number of consumer vehicles is set to [5,50], and the number of server vehicles is 100. By default, all vehicles travel in the same direction. The vehicle communication network is based on IEEE 802.11p V2I physical /MAC protocol Settings. For V2R communication, the coverage of each base station is set to 0.5km [28], and the service vehicles that are within the coverage of the base station can offload tasks to the edge server for computation. For V2V communication, the communication range of each vehicle is set to 0.15km [10], and the task vehicle can transmit its task to any service vehicle within its communication range. In the same way, the service vehicle can offload the task to other service vehicles, or use its own computing resources to process the task. In addition, the height of all vehicle antennas is 1.5m, and the height of base station antennas is 9m [28]. Table 1 describe the main parameters used in the simulations.

Table 1: Simulation parameters setting.

Parameters	Value
Length of the street L /km	3
The simulation time T /s	30

ISSN 2616-5775 Vol. 6, Issue 4: 34-48, DOI: 10.25236/AJCIS.2023.060405

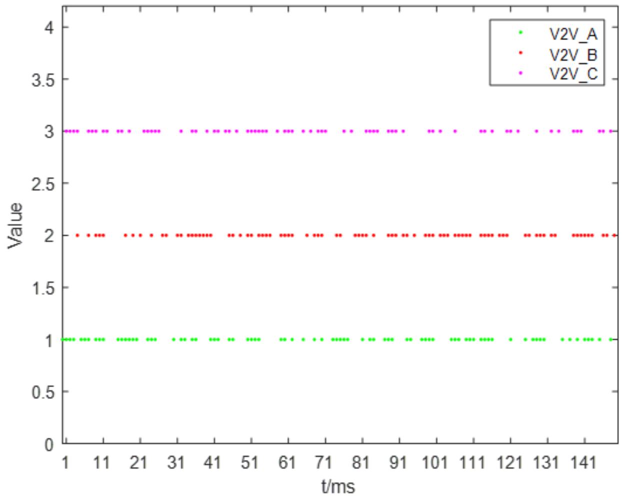
Single time slot length Δt /ms	20
The number of consumer vehicles	[5,50]
The number of service vehicles	100
The speed of the vehicle s /(km/h)	[3,40]
Height of vehicle antenna h_{-Veh} /m	1.5
Height of the base station antenna h_{-RSU} /m	9
The number of base stations	2
The base station coverage r_{-RSU} /m	500
Vehicle communication range r_{-Veh} /m	150
Transmitted power of vehicle p /mw	200
Vehicle to RSU resource block bandwidth b_{V2R} /kHz	180
Total number of V2R resource blocks K	50
Vehicle to vehicle communication bandwidth b_{V2V} /kHz	60
Noise spectral density N_0 /(dBm/Hz)	-174
Path loss coefficient α	3.75
The amount of task input data A_n /kb	[400,1000]
The number of CPU cycles required per unit of data C_n /(cycles/bit)	[200,500]
Maximum tolerable delay of the task D_n /ms	[90,120]
Vehicle computing capacity f /(cycles/s)	[1×10^9,2×10^9]
Edge server computing power F /(cycles/s)	[2×10^{10},8×10^{10}]

6.2. Analysis of experimental results

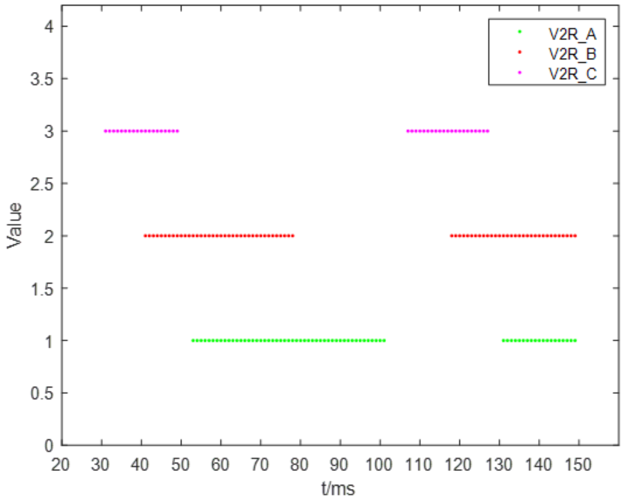
(1) Experiments on vehicle mobility and communication link state analysis

In this article, the reliable link time of a communication link refers to the duration of the state in which a vehicle is within the communication range of other vehicles or edge server nodes. Based on the above analysis, it mainly depends on the moving speed of the vehicle. In order to obtain the mobile communication link of vehicles at different speeds, we set up a time comparison experiment under low speed, medium speed and high speed, in which the moving speeds of consumer and service provider vehicles are (5,10)km/h, (15,20)km/h and (35,40)km/h in turn. The numerical result curves correspond to V2V_A (V2R_A), V2V_B (V2R_B), and V2V_C (V2R_C) in sequence. The simulation duration was 300s, and the distance between vehicles and the distance between vehicles and the edge server was calculated every two seconds. If the distance is within the vehicle communication range or the base station coverage range, the vehicle V2V or V2R communication link is considered to be linkable at that time, and the corresponding value is positive; otherwise, the value is zero. The experimental results are shown in *Figure 5 (a)* and *Figure 5 (b)*.

As for the experimental results in *Figure 5 (a)*, firstly, by comparing the experimental results of three types of ABC V2V communication links, it is found that, at different speeds, the state of V2V communication links will change intermittently in the irrelevant time slot state. The reason for this phenomenon is the irregular change of vehicle speed. Secondly, by analyzing the link duration of each single communication link at different vehicle speeds, it can be found that with the increase of vehicle speed, the probability that the link duration of a single communication link becomes smaller keeps rising, which can be explained by the increase of distance variation caused by the increase of speed. As for the experimental results shown in *Figure 5 (b)*, firstly, the duration of V2R communication links is significantly higher than that of V2V communication links due to the wide coverage of edge servers. Second, due to the limited number of edge servers, there is no global coverage, so vehicles cannot complete V2V communication links at long intervals. This means that vehicles can only use their own computing resources to compute tasks when they do not use V2V communication links. If the computing resources of vehicles are insufficient, it is difficult to ensure that tasks are processed in a timely manner. According to the experimental results in figure 5 (a) and figure 5 (b), when the task vehicle is outside the coverage of the edge server, compared with the mode of waiting for the vehicle to enter the coverage of the server before offloading, enabling the two communication modes of V2V and V2R can improve the utilization of network equipment resources and better ensure the completion time of the task.



(a) V2V communication links



(b) V2V communication links

Figure 5: V2V and V2R communication links in different time slots.

(2) Delay Cost Comparison Experiment

To verify the effectiveness of the proposed MHORA algorithm in reducing the delay cost, it is compared with three other multi-hop offloading algorithms. The first algorithm is Local and Single-hop offloading algorithm (LSHORA), which represents that vehicle tasks can only use local resources or single-hop task offloading vehicles for task processing. The second algorithm is Random Multi-hop offloading algorithm (RMHORA), which represents that each hop offloading node of the consumer vehicle is randomly generated. The third comparison algorithm is the MMTO algorithm proposed in literature [24].

We first studied the impact of the number of consumer vehicles on the delay cost of different algorithms, and the experimental results are shown in *Figure 6*. The simulation time set in the experiment is 1000s, and the number of service vehicles is 100, evenly distributed on the street. Set all the vehicle's speed is (3, 10 km/h). The experimental results show that the delay cost of the four unloading schemes is increasing as the number of consumer vehicles increases. Compared with the other three algorithms, MHORA algorithm proposed in this paper has the best overall performance and the lowest delay cost. Due to the limitations of greedy algorithm, MHORA algorithm does not gain obvious advantages over the other three comparison algorithms in the case of some consumer vehicles. Firstly, when the number of vehicles is 20, the delay cost of MHORA algorithm is higher than that of RMHORA algorithm and MMTO algorithm. This is because each hop target node of the greedy algorithm is selected according to the optimal value under the current situation, which may miss some multi-hop offloading paths with better performance in the future. Secondly, when the number of consumer vehicles increases to 50, the

ISSN 2616-5775 Vol. 6, Issue 4: 34-48, DOI: 10.25236/AJCIS.2023.060405

delay cost of local or single-hop offloading algorithm LSHORA is the lowest compared with other schemes. This is because that, in the case of limited number of server vehicles, vehicle users will compete with each other for limited relay multi-hop offloading nodes, which will increase the completion delay of the task when there are too many vehicles of consumers. However, the local or single-hop offloading algorithm can support local computing mode, which is beneficial to alleviate the problem of multi-hop resource shortage. The results in *Figure 6* illustrate that multi-hop offloading is more advantageous in scenarios with sufficient resources of service vehicles.

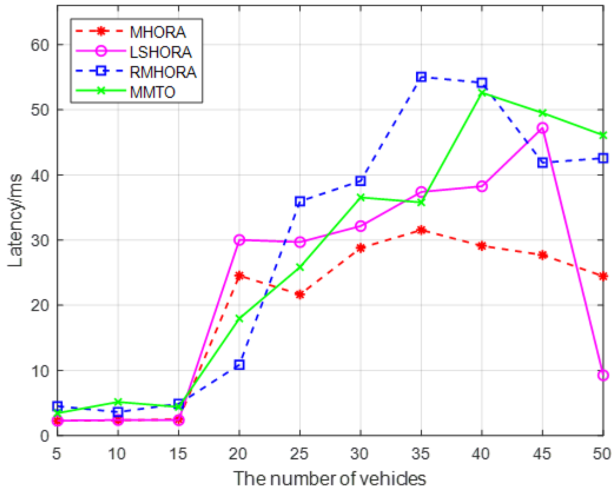


Figure 6: Impact of the number of consumer vehicles on the delay cost of different algorithms.

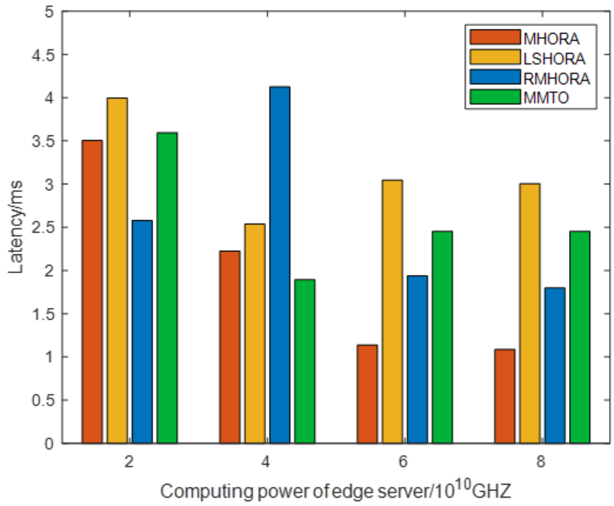


Figure 7: Impact of edge server computing capacity on delay cost of different algorithms.

Figure 7 shows the offload delay cost of different algorithms when the computing power of edge servers is gradually increasing. In the scenario, the number of consumer vehicles is set to 30, and the simulation time, number of server vehicles, vehicle position and driving speed are the same as those in *Figure 7*.It can be seen from the figure that the delay cost of the four algorithms shows a decreasing trend as the computing power of the edge server increases. Compared with the other three algorithms, the LSHORA algorithm has the worst performance in cost reduction. This is because when the consumer vehicle is outside the scope of the edge server, the local or single-hop offloading calculation method is difficult to effectively utilize the resources of the edge server and the service vehicle, which results in the highest delay cost. RMHORA algorithm and MMTO algorithm adopt multi-hop offloading mode, which improves the utilization rate of vehicle resources of the server, so the delay cost is relatively low, and the random multi-hop unloading algorithm shows relatively strong instability. In general, the MHORA algorithm proposed in this paper has the highest resource utilization efficiency for the server vehicles and edge server devices in the network, and when the computing power of the edge server increases, the effect of reducing the delay cost is most significant.

ISSN 2616-5775 Vol. 6, Issue 4: 34-48, DOI: 10.25236/AJCIS.2023.060405

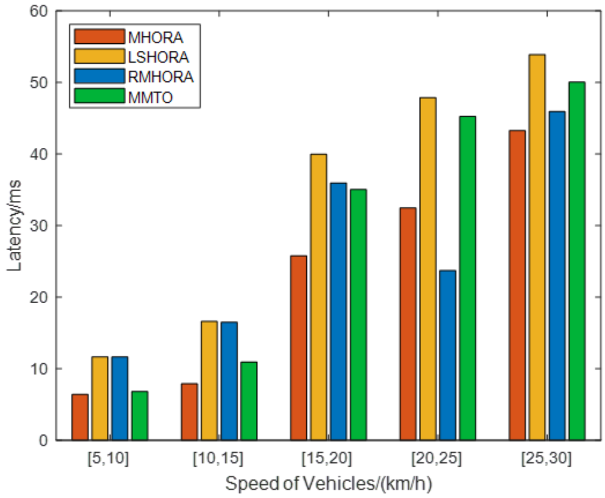


Figure 8: Impact of vehicle speed on delay cost of different algorithms.

The experimental results in figure 8 demonstrate the impact of vehicle speed on the delay cost of different algorithms. The driving speed of the vehicle determines the contact time between the vehicle and its current server vehicle or the edge RSU. The higher the vehicle speed is, the shorter the contact time between the vehicle and its neighboring nodes is. Therefore, the time delay cost of offloading will increase with the increase of vehicle speed. According to the communication link reliability analysis experiment, V2V multi-hop offloading can provide higher reliable link probability for vehicle task offloading in the process of high speed driving. Compared with the other three algorithms, the MHORA algorithm in this paper improves the resource utilization of service vehicles and edge nodes in the network through multi-hop task offloading and server resource allocation, so as to achieve the lowest offloading delay. Although the random multi-hop offloading algorithm RMHORA achieves lower delay cost than the MHORA algorithm when the vehicle speed is (20, 25) km/h, similar experimental results are not obtained in the other four sets of speed-delay experiments. According to the experimental results of figure 6, figure 7 and figure 8, the MHORA algorithm of the proposed scheme can better reduce the task delay cost and improve the satisfaction of vehicle users compared with the other three comparison algorithms in the case of unstable vehicle speed and sufficient edge resources or idle vehicle resources.

7. Conclusion

Aiming at the problems of long transmission distance and long offloading delay of vehicle tasks outside the coverage of edge servers in Internet of vehicles (IoV), this paper proposes a multi-hop assisted offloading scheme based on mobile perception. By using the flexible change characteristics of vehicle location, the vehicle can be offloaded to the edge server or other vehicles with richer computing resources in advance through multi-hop transmission when it does not enter the coverage of the edge server. The multi-hop offloading of vehicle mobility and task was modeled as a Markov decision process, and a multi-hop-assisted task offloading and resource allocation algorithm MHORA was designed to solve the problem iteratively. According to the numerical results of simulation experiments, the proposed scheme has more advantages than some existing task computing offloading schemes when the vehicle is running at a high speed and the edge resources are sufficient. In the future research work, we will focus on the limitations of the greedy algorithm to improve, and strive to find the multi-hop offloading scheme as the optimal solution. At the same time, the dependence or priority characteristics of tasks will be considered to meet the personalized service needs of different vehicle users.

Acknowledgements

This work was supported by the General Program of National Natural Science Foundation of China Grant No. 62276092, Henan Province University Young Key Teacher Funding Plan Grant No. 2019GGJS061.

References

[1] Ning Z, Dong P, Wang X, et al. Partial computation offloading and adaptive task scheduling for 5G-enabled vehicular networks [J]. IEEE Transactions on Mobile Computing, 2020, 21(4): 1319-1333.
[2] Shinde S S, Tarchi D. Collaborative Reinforcement Learning for Multi-Service Internet of Vehicles [J]. IEEE Internet of Things Journal, 2022.
[3] Zhang D, Cao L, Zhu H, et al. Task offloading method of edge computing in internet of vehicles based on deep reinforcement learning[J]. Cluster Computing, 2022, 25(2): 1175-1187.
[4] Chattopadhyay R, Tham C K. Fully and partially distributed incentive mechanism for a mobile edge computing network [J]. IEEE Transactions on Mobile Computing, 2020, 21(1): 139-153.
[5] Bao L, Yang Y, Mao J, et al. Resource Scheduling Optimization under Multi-Access Edge Computing Architecture [M]//CICTP 2022. 2022: 668-678.
[6] Dai P, Hu K, Wu X, et al. A probabilistic approach for cooperative computation offloading in MEC-assisted vehicular networks [J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 23(2): 899-911.
[7] Huang C M, Chiang M S, Dao D T, et al. V2V data offloading for cellular network based on the software defined network (SDN) inside mobile edge computing (MEC) architecture[J]. IEEE Access, 2018, 6: 17741-17755.
[8] Qiao G, Leng S, Zhang K, et al. Collaborative task offloading in vehicular edge multi-access networks [J]. IEEE Communications Magazine, 2018, 56(8): 48-54.
[9] Raza S, Mirza M A, Ahmad S, et al. A vehicle to vehicle relay-based task offloading scheme in Vehicular Communication Networks [J]. Peer J Computer Science, 2021, 7: e486.
[10] Raza S, Liu W, Ahmed M, et al. An efficient task offloading scheme in vehicular edge computing [J]. Journal of Cloud Computing, 2020, 9: 1-14.
[11] Wang X, Ning Z, Guo S, et al. Imitation learning enabled task scheduling for online vehicular edge computing [J]. IEEE Transactions on Mobile Computing, 2020, 21(2): 598-611.
[12] Bute M S, Fan P, Liu G, et al. A cluster-based cooperative computation offloading scheme for C-V2X networks [J]. Ad Hoc Networks, 2022, 132: 102862.
[13] Luo G, Zhou H, Cheng N, et al. Software-defined cooperative data sharing in edge computing assisted 5G-VANET [J]. IEEE Transactions on Mobile Computing, 2019, 20(3): 1212-1229.
[14] Zhang K, Cao J, Zhang Y. Adaptive digital twin and multiagent deep reinforcement learning for vehicular edge computing and networks [J]. IEEE Transactions on Industrial Informatics, 2021, 18(2): 1405-1413.
[15] Lang P, Tian D, Duan X, et al. Cooperative computation offloading in blockchain-based vehicular edge computing networks [J]. IEEE Transactions on Intelligent Vehicles, 2022, 7(3): 783-798.
[16] Zhang R, Cheng P, Chen Z, et al. Online learning enabled task offloading for vehicular edge computing[J]. IEEE Wireless Communications Letters, 2020, 9(7): 928-932.
[17] Tang C, Zhu C, Wu H, et al. Toward response time minimization considering energy consumption in caching-assisted vehicular edge computing [J]. IEEE Internet of Things Journal, 2021, 9(7): 5051-5064.
[18] Bozorgchenani A, Maghsudi S, Tarchi D, et al. Computation offloading in heterogeneous vehicular edge networks: On-line and off-policy bandit solutions[J]. IEEE Transactions on Mobile Computing, 2021, 21(12): 4233-4248.
[19] Moon S, Lim Y. Task migration with partitioning for load balancing in collaborative edge computing [J]. Applied Sciences, 2022, 12(3): 1168.
[20] De Souza A B, Rego P A L, Chamola V, et al. A Bee Colony-Based Algorithm for Task Offloading in Vehicular Edge Computing [J]. IEEE Systems Journal, 2023.
[21] Mittal S, Dudeja R K, Bali R S, et al. A distributed task orchestration scheme in collaborative vehicular cloud edge networks [J]. Computing, 2022: 1-25.
[22] Polese M, Jana R, Kounev V, et al. Machine Learning at the Edge: A Data-Driven Architecture with Applications to 5G Cellular Networks. CoRR abs/1808.07647 (2018)[J]. arXiv preprint arXiv: 1808. 07647, 2018.
[23] Huang X, He L, Chen X, et al. Revenue and energy efficiency-driven delay-constrained computing task offloading and resource allocation in a vehicular edge computing network: A deep reinforcement learning approach [J]. IEEE Internet of Things Journal, 2021, 9(11): 8852-8868.
[24] Liu L, Zhao M, Yu M, et al. Mobility-aware multi-hop task offloading for autonomous driving in vehicular edge computing and networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2022.
[25] Waheed A, Shah M A, Khan A, et al. Hybrid task coordination using multi-hop communication in volunteer computing-based VANETs [J]. Sensors, 2021, 21(8): 2718.

ISSN 2616-5775 Vol. 6, Issue 4: 34-48, DOI: 10.25236/AJCIS.2023.060405

[26] Yuan Q, Li J, Zhou H, et al. A joint service migration and mobility optimization approach for vehicular edge computing [J]. IEEE Transactions on Vehicular Technology, 2020, 69(8): 9041-9052.
[27] Zhang Y, Li C, Luan T H, et al. A mobility-aware vehicular caching scheme in content centric networks: Model and optimization [J]. IEEE Transactions on Vehicular Technology, 2019, 68(4): 3100-3112.
[28] Musa S S, Zennaro M, Libsie M, et al. Mobility-aware proactive edge caching optimization scheme in information-centric iov networks [J]. Sensors, 2022, 22(4): 1387.
[29] Hu R Q. Mobility-aware edge caching and computing in vehicle networks: A deep reinforcement learning [J]. IEEE Transactions on Vehicular Technology, 2018, 67(11): 10190-10203.
[30] Wu Q, Zhao Y, Fan Q, et al. Mobility-Aware Cooperative Caching in Vehicular Edge Computing Based on Asynchronous Federated and Deep Reinforcement Learning [J]. IEEE Journal of Selected Topics in Signal Processing, 2022.
[31] Hu X, Huang Y. Deep reinforcement learning based offloading decision algorithm for vehicular edge computing [J]. Peer J Computer Science, 2022, 8: e1126.
[32] Hart W E, Watson J P, Woodruff D L. Pyomo: modeling and solving mathematical programs in Python [J]. Mathematical Programming Computation, 2011, 3: 219-260.