J Inf Process Syst, Vol.18, No.3, pp.443~456, June 2022
https://doi.org/10.3745/JIPS.02.0175

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Resource Allocation Strategy of Internet of Vehicles Using Reinforcement Learning

Hongqi Xi* and Huijuan Sun

Abstract

An efficient and reasonable resource allocation strategy can greatly improve the service quality of Internet of Vehicles (IoV). However, most of the current allocation methods have overestimation problem, and it is difficult to provide high-performance IoV network services. To solve this problem, this paper proposes a network resource allocation strategy based on deep learning network model DDQN. Firstly, the method implements the refined modeling of IoV model, including communication model, user layer computing model, edge layer offloading model, mobile model, etc., similar to the actual complex IoV application scenario. Then, the DDQN network model is used to calculate and solve the mathematical model of resource allocation. By decoupling the selection of target Q value action and the calculation of target Q value, the phenomenon of overestimation is avoided. It can provide higher-quality network services and ensure superior computing and processing performance in actual complex scenarios. Finally, simulation results show that the proposed method can maintain the network delay within 65 ms and show excellent network performance in high concurrency and complex scenes with task data volume of 500 kbits.

Keywords

DDQN Model, Internet of Vehicles, Markov Decision Model, Mobile Edge Computing, System Model

1. Introduction

In today's rapid economic development, vehicles are one of the indispensable and important transportation methods for human travel [1,2]. As the end of 2020, the scale of motor vehicles in China has increased to 365 million [3]. The traffic problem has thus become one of urgent problems. Internet of Vehicles (IoV) is an important starting point for transportation power, and its emergence provides a new solution for smart transportation [4-6]. IoV realizes the mutual communication between vehicles, vehicles and equipment, as well as vehicles and everything [7,8]. The application of IoV in actual traffic can improve the diversity of information services, the safety of travel and the efficiency of urban traffic [9]. However, regardless of the convenience brought by application services, they cause geometric growth of data, increase the corresponding network load and put forward higher demands on network bandwidth [10].

The enormous abundance of in-vehicle applications puts higher demands on computing and storage resources [11,12]. The traditional centralized IoV processing mode cannot efficiently support the high-efficiency requirements of IoV applications. The emergence of mobile edge computing (MEC) places the

Manuscript received November 16, 2021; first revision March 10, 2022; accepted March 15, 2022.
* **Corresponding Author:** Hongqi Xi (ourbigdata@126.com)
College of Computer and Information Technology, Henan Finance University, Zhengzhou, Henan, China (ourbigdata@126.com, 20488481@qq.com)

computing process at the edge of network, which can be close to the data terminal and terminal side. It uses local analysis and decision-making to achieve accurate and reliable network resource allocation and task offloading [13].

MEC is an effective method to cope with the increasing demand for computing and caching [14]. Roadside unit (RSU) and MEC are effectively integrated to make the deployment of computing resources closer to the location of terminal vehicles, so that computing-intensive in-vehicle applications can be transferred from vehicles to MEC servers on various adjacent RSU with the help of computing offloading technology [15], real-time data processing on the server and complete feedback. However, the MEC system of IoV has inherent characteristics such as fast vehicle movement, frequent network topology changes and short interaction between devices [16,17]. This poses unprecedented challenges for the research of computing offloading in IoV MEC system [18]. Therefore, a task offloading scheme that adapts to the dynamic changing environment must be proposed to ensure the reliability and efficiency of vehicle task offloading.

Scholars have carried out various research on resource allocation in IoV. Wang et al. [19] used the convex optimization standard method to plan resource allocation. Zhang et al. [20] adopted Lyapunov optimization theory to weigh the average energy consumption and the average delay to realize the effective allocation of IoV network resources. In [21], the authors used the branch and bound method to improve the search tree algorithm, and solved the problem of minimizing time delay, jointly considering computational offloading, content caching and resource allocation. However, it should be noted that most IoV resource allocation solutions use mathematical solving algorithms to directly solve the optimal problem. But the requirements are too high for the computing power and the algorithm. In addition, the current network is becoming more complex, and the requirements for real-time communication are also higher. The traditional solving method using mathematical algorithms is difficult to realize the complete dynamic perception of network state, which affects the efficiency of resource allocation in IoV network.

The multi-layer learning network can continuously improve the training learning model and formulate corresponding optimal strategy by interacting with the dynamic environment and relying on the state data at the last moment. This can meet the different service quality requirements of vehicle network, and then support the stable operation of IoV network. The authors of [22] considered the mobility characteristics of urban vehicles, and used a reinforcement learning network to realize the research on resource allocation of IoV in MEC mode. Cui et al. [23] combined empirical model decomposition and long and short-term memory network to achieve effective network resource allocation. Chen et al. [24] proposed a high-performance asynchronous advantage actor-critic resource allocation method based on software-defined network and information-centric IoV system to improve the performance of virtual wireless networks. Zhao et al. [25] used deep reinforcement learning DQN (deep Q network) model to allocate resources and to reduce system complexity. However, it is limited by the number of vehicles in the network, which affects the efficiency of task offloading. However, it needs to be noticed that the reinforcement learning method adopted by current resource allocation methods uses the greedy idea to achieve continuous calculation and solutions of the objective function [26,27]. However, the problem of overfitting leads to the overestimation of resource allocation risk for optimal solution, making it difficult for IoV network to provide efficient and reliable application service requirements.

In response to the above problems, this paper proposes an efficient IoV resource allocation method based on deep reinforcement learning network to ensure the high-quality performance of network. The innovative points of this paper are:

1) Introduce the user-level task offloading model and edge-level offloading model to construct the IoV system model. This can realize the horizontal penetration and vertical collaboration of resource allocation in IoV network to provide faster task processing services.
2) Use double DQN (DDQN) network model to effectively solve the IoV resource allocation problem, solve the problem of overestimation in current methods and provide better quality for IoV network services.

2. System Model

2.1 Network Model

The MEC-based IoV network model proposed in this paper is shown in Fig. 1. Since both the MEC server and neighboring vehicles have computing and caching capabilities, they are collectively referred to as the service node $L = \{l_1, l_2, l_3, \dots\dots\}$. In this model, RSU A is deployed around the road, denoted as $A = \{a_1, a_2, a_3, \dots\dots\}$, and each RSU is equipped with an MEC server. There are J vehicles on the road in Poisson distribution, and their vehicle identification is $J = \{j_1, j_2, j_3, \dots\dots\}$, so the vehicles are represented as $K = \{k_1, k_2, k_3, \dots\dots\}$. n vehicles are randomly distributed within the coverage of each RSU, and the vehicle set of cells i is $J_i = \{j_{i1}, j_{i2}, j_{i3}, \dots\dots, j_{in}\}$.

The spectrum is equally divided into λ sub-channels, and the bandwidth of each sub-channel is B. The vehicle offloading strategy set is denoted as $\mu = \{\mu_1, \mu_2, \mu_3, \dots\dots, \mu_k\}$. If $\mu_i = 1$, it means that k_i will offload tasks to the service node l_i for calculation; if $\mu_i = 0$, it means that k_i will perform the computing task locally. The set of caching strategies of the service node is denoted as $S = \{s_1, s_2, s_3, \dots\dots, s_k\}$. If $s_i = 1$, it means that the service node l_i will cache the computing task for the next request.

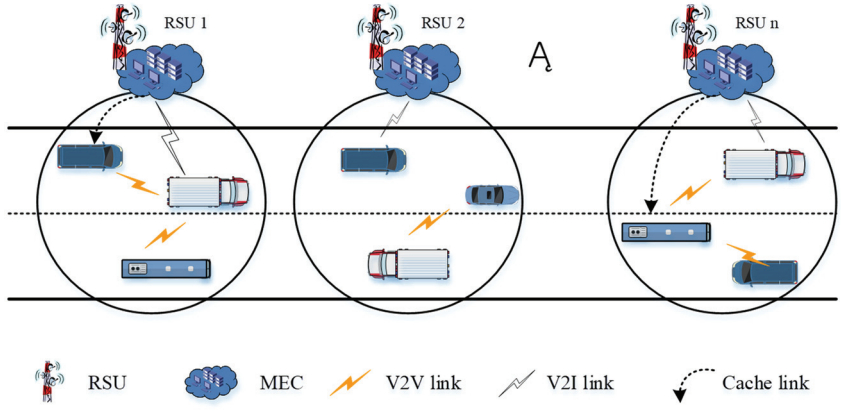

Fig. 1. MEC-based IoV network model.

2.2 Communication Model

V2V communication adopts IEEE 802.11p protocol based on distributed coordination function (DCF) [28]. Vehicles use the CSMA/CA mechanism to compete for channels. Let R_a denote the average number of time slots required to successfully transmit a piece of data, and R_s denote the average length of each

time slot, t. Then, the average delay required to successfully transmit a piece of data between vehicle i and neighboring vehicle k is

$$t_{ik} = R_a \cdot R_s \tag{1}$$

When task vehicles are performing edge offloading, they upload tasks to roadside units by V2I communication, where the communication uses LTE-V2X protocol. Define g_i and B_i as the channel gain and the channel bandwidth between the vehicle and the roadside unit respectively. Let τ_i denote the transmission power of users, and σ^2 denote the transmission noise. Then, according to Shannon's theorem, the data upload rate is:

$$v_i = B_i \text{lb}(1 + \frac{\tau_i g_i}{\sigma^2}) \tag{2}$$

2.3 Offloading Model

For IoV calculation model in MEC mode, it is divided into two ways of processing tasks: user layer calculation and edge layer offloading [29].

2.3.1 User layer calculation

In order to make full use of vehicle resources, in addition to processing tasks locally, the task vehicle can also use neighboring vehicles within its communication range to implement task calculations. Define f_i as the computing power of task vehicle i itself. Then, the time required for vehicle i to process its own task through local calculation can be expressed as

$$t_i = \frac{m_i}{f_i} \tag{3}$$

Defined as a collection of vehicles within the communication range of vehicle i. When the task vehicle uses its neighbor vehicle k to perform computing tasks, the delay includes the transmission delay of task between the two vehicles, the calculation delay of tasks in the neighboring vehicle and the feedback delay of results. Here, this paper ignores the feedback delay of results. For the transmission delay, the average transmission delay R_{ik} can be obtained according to formula (1); for the calculation delay, the average calculation delay t_i can be obtained by formula (3).

The constraints are

$$t_{ik} + t_i < \varepsilon_i \tag{4}$$

where the constraint condition is to ensure that the selected neighboring vehicle can complete the task processing and give feedback within effective communication time ε_i of the two vehicles.

2.3.2 Edge layer offloading

When the task vehicle performs edge offloading, it generally includes the following stages: task upload, task execution and result feedback. This paper ignores the time delay of result feedback and assumes that

task vehicle i chooses the roadside unit for offloading to be l, and analyzes the time delays at different stages.

Task upload stage. The vehicle i first uploads tasks to the currently associated roadside unit l'. The transmission delay of this process depends on the size of tasks and the data transmission rate. From Eq. (2), we can get

$$t_{il'} = \frac{d_{ik}}{v_i} \tag{5}$$

where d_{ik} represents the size of input data.

Task execution stage. According to the location of selected offload server, task execution is divided into the following two situations.

Case 1: The roadside unit l and l' are the same. In this case, the task is calculated in the current roadside unit. If the roadside unit stores service applications required to calculate the task, it can directly calculate the task. The required delay depends on the computing power requirements of task and the computing resources allocated by the roadside unit. Otherwise, the additional delay t_{icloud} of downloading the corresponding service application from the cloud must also be considered. In summary, the time required to complete the task calculation is

$$t_{il} = (1 - w_{il})t_{icloud} \tag{6}$$

where w_{il} represents the cache decision coefficient. When the cache decision $w_{il} = 1$, the roadside unit l can handle the task of offloading vehicle i.

Case 2: Roadside units l and l' are different. In this case, the migration delay of tasks between them needs to be considered. The roadside units are connected by wired links. Assuming that there are $\theta_{ll'}$ links between l and l', and the average transmission delay of each link is $t_{l''}$, then the task migration time between two roadside units is $t_{ll'} = t_{l''} \cdot \theta_{ll'}$. Combining formula (6), the time required to complete the task processing can be obtained as

$$t_{il} = (1 - w_{il})t_{icloud} + t_{l''} \cdot \theta_{ll'} \tag{7}$$

Once the selected roadside unit completes the task processing, the result needs to be fed back to vehicles. Due to mobility, it is necessary to consider whether the vehicle may have driven out of the initially associated roadside unit at this time. Therefore, the time T_{il} of tasks in the upload and execution phase can be compared with the time ς_{il} of vehicles in the initial transmission range of associated servers. Among them, $T_{il} = t_i + t_{il}$ depends on the ratio of time for users leaving the server's communication range to the moving speed. If $T_{il} < \varsigma_{il}$, the result can be transmitted to the roadside unit first, and then fed back to vehicles. Otherwise, it is necessary to locate the movement of vehicles and determine which roadside unit is currently located in order to transmit the result to servers and then feed it back to vehicles.

2.4 Mobile Model

The moving speed of vehicles is much faster than that of ordinary mobile equipment terminals (mobile phones, computers, etc.), and its moving range is larger in a short time. Thus, the mobility of vehicle users cannot be ignored when constructing mathematical models [30].

First, suppose that the vehicle is moving in a two-dimensional Euclidean plane, and the parameter group $\kappa(i) = \{pa(i), lt(i), md(i), mv(i)\}$ is used to characterize the mobility of vehicles. In the parameter group, $pa(i)$ is the accuracy of the location of vehicles i, $lt(i)$ is the latitude of the location of vehicles i, $md(i)$ represents the moving direction of vehicles i, and $mv(i)$ represents the moving speed of vehicles i. According to the definition of the parameter tuple, the future position of vehicles can be calculated by the following equation:

$$pa(i') = pa(i) + mv(i)(i' - i)\ sin\ m\ d(i) \tag{8}$$

$$lt(i') = lt(i) + mv(i)(i' - i)\ cos\ m\ d(i) \tag{9}$$

Definition $d_{max\,1}$ is the maximum communication distance from the vehicle user to the relay (MEC server), and at the same time, $d_{max\,2}$ is defined as the maximum communication distance between vehicle user to vehicle users. Only devices within the communication range can establish successful connection communication. In order to implement calculation offloading, vehicle users need to report their own mobility parameter set at the beginning of each time slot to determine the connection status of vehicles to vehicle/relay in the time slot.

3. Resource Allocation Algorithm based on DDQN Network Model

3.1 Objective Function

This paper aims to resolve the contradiction between limited network resources and different user needs in a dynamic, random and time-varying vehicle environment by the joint optimization of computing offloading and service cache resources, and minimize the system on the premise of ensuring user service needs. In view of this, the designed objective function is as follows:

$$\min_{s,f} \sum_{i \in K} (s_{i0} t_i + \sum_{l \in L} s_{il} T_{il}) \tag{10}$$

$$C1: s_{il} \in \{0,1\}, \forall i \in K, \forall l \in L \tag{11}$$

$$C2: \sum_{l \in L} s_{il} = 1\ , \forall i \in K \tag{12}$$

$$C3: \sum_{i} s_{il} f_{il} < F_l\ , \forall l \in L \tag{13}$$

$$C4: T_{il} < \varpi_i, \forall i \in K \tag{14}$$

where according to formulas (3)–(7), t_i and T_{il} can be obtained respectively. Here, it is assumed that the bandwidth resources allocated by vehicles are the same. Restrictive condition C1 indicates that each task has two processing methods: user-level processing and edge-level offloading. C2 means that each task is executed in only one place; C3 refers to the computing resource limit of servers. C4 indicates that the task of offloading vehicle to the roadside unit should be completed before it leaves the transmission range of associated servers. Among them, ϖ_i represents the connection time between the user and its associated

server, which depends on the ratio of time when the user leaves the servers' communication range to the moving speed.

3.2 Markov Decision Model Construction

The problem of computing task offloading in IoV can be modeled as a partially observed Markov decision process (POMDP), which can be defined as $(\alpha, \beta, \gamma, \varepsilon, \theta)$, that is, state α, action space β, state transition probability γ, discount factor ε and reward θ. The definitions of the above components are given below.

State space: The overall state space is composed of state spaces of multiple mission vehicles. The state space α of task vehicle i can be defined as

$$\alpha_i \in \alpha, \alpha_i = \{lt(i), mv(i), ne(i), cr(i), sr(i)\} \tag{15}$$

where $lt(i)$, $mv(i)$, $ne(i)$, $cr(i)$ and $sr(i)$ represent the position, speed, neighborhood vehicle collection, computing resource demand and spectrum resource demand of vehicles, respectively. It can be considered as a feature vector extracted from IoV environment.

Action space: The overall action space of DDQN resource allocation model is composed of the action spaces of multiple task vehicles. The action space is composed of candidate task destination nodes, such as service vehicles, and RSU is defined as follows:

$$E_i \in E, E_i = \{Q_i\}, Q_i = \{{Q_i}^0, {Q_i}^1, {Q_i}^2, \dots \dots\} \tag{16}$$

where ${Q_i}^0$ represents the index of task vehicle i unloaded to RSU, and $\{{Q_i}^0, {Q_i}^1, {Q_i}^2, \dots \dots\}$ represents the index of task vehicle i unloaded to the neighborhood collection vehicle.

Domain action space: In order to improve the convergence speed of the algorithm, in addition to the asynchronous iterative mechanism, the proposed algorithm also introduces the concept of neighborhood action space. The neighborhood action space aims to allow each vehicle to focus its attention on its surrounding environment, that is, to improve the computational efficiency of computing tasks by offloading IoV tasks to the closer task destination node. At the same time, when computing resources are the same, the performance of computing task offloading is mainly dominated by the performance of communication phase. If the service vehicle is far away from the mission vehicle, the transmission rate will inevitably be affected by factors, for example, the path loss. Thus, in order to maximize its own revenue, each vehicle will have a greater probability to choose a closer vehicle or transportation infrastructure to achieve the offloading of computing tasks. For convenience, the size of neighborhood action space proposed in this paper can be dynamically adjusted, which can improve the adaptability of DDQN resource allocation model in a dynamically changing IoV environment.

Reward function: The reward function is crucial to the performance of DDQN algorithm, which can guide the algorithm to continuously approach the optimal value. In DDQN algorithm, the agent will receive an instant reward $\theta(\alpha, \beta)$ after performing action β according to system state α. When the reward is positive, the tendency of agents to choose action β in state α will increase and vice versa. The key role of instant reward is to adjust the training direction of neural network through its own increase and decrease, and then adjust the transition probability between states and actions.

Generally speaking, the reward function usually needs to be designed based on specific problems. θ_{total} can be defined as the overall reward in the model. θ_{total} mainly includes two parts: communication phase θ_{total1} and calculation phase θ_{total2}.

The reward value θ_{total1} in communication phase can be expressed as

$$\theta_{total1} = ben_{comm} - cost_{comm} \tag{17}$$

$$ben_{comm} = C^{V2V} + C^{V2I} \tag{18}$$

$$cost_{comm} = \xi C^{V2I} \tag{19}$$

where the revenue and the expenditure of the communication stage are ben_{comm} and $cost_{comm}$ respectively. ben_{comm} is the total revenue of V2V and V2I connection, $cost_{comm}$ is the cost of V2I connection, and ξ is the unit price of V2I communication connection.

The reward value θ_{total2} in calculation phase can be expressed as

$$\theta_{total2} = ben_{comp} - cost_{comp} \tag{20}$$

$$ben_{comp} = \sum_{i \in K} \sum_{j \in K} U_{ij} C_{ij}{}^{comp} \tag{21}$$

$$cost_{comp} = \zeta \sum_{i \in K} \sum_{l \in L} U_{il} C_{il}{}^{comp} \tag{22}$$

The revenue and expenditure in calculation phase are ben_{comp} and $cost_{comp}$ respectively, and ζ is the price of the unit computing resource block in RSU. $U_{ij} = 1$ represents that task vehicle i will offload its computing tasks to service vehicle j; $U_{il} = 1$ represents that task vehicle i will offload its computing tasks to the infrastructure l.

3.3 Solution Strategy

From the above analysis, it can be seen that the IoV resource allocation task problem is a Markov dynamic decision-making process. The system will perform appropriate actions according to the current state and offloading strategy to maximize long-term benefits or achieve a certain goal. In order to meet the research goals, as shown in Algorithm 1, the offloading is calculated based on DDQN network model. DDQN task allocation algorithm uses the multi-threaded asynchronous processing technology and uses multiple sub-neural networks to update the global neural network asynchronously, which can accelerate the convergence of neural network.

4. Experiment and Analysis

This paper is based on Python simulation software to simulate and verify the offloading algorithm of in-vehicle edge computing. The pros and cons of different algorithms are evaluated by comparing the performance of each algorithm in terms of delay and rewards with changes in the number of vehicles, the roadside unit computing power and the storage capacity. In addition to the algorithm in this paper, the implemented algorithms also include the ones in [23,24]. All algorithms run in the same software and hardware environment.

Algorithm 1. Resource allocation task algorithm

Input: State matrix α
Output: Assigning actions β
1: Get status information from vehicle and MEC server
2: Create initial state matrix α
3: The number of initial training steps is $x \leftarrow 1$
4: Repeat
5: DDQN gradient parameter zeroing: $d\varphi \leftarrow 0$, $d\varphi_N \leftarrow 0$
6: Synchronization thread characteristic parameters
7: Obtain the status information from the vehicle and MEC server and construct the status matrix α'
8: Repeat
9: Generate offloading action β' according to policy $\pi(\beta'|\alpha';\varphi)$
10: Vehicle and MEC server execute action β'
11: Get new status information from vehicle and MEC server
12: Generate a new state matrix α''
13: $x \leftarrow x + 1$
14: If training ends then
15: $\theta = 0$
16: else
17: $\theta = \theta_{total1} + \theta_{total2}$
18: Until training ends
19: for $x \in [1,2,3, \ldots \ldots, X]$ do
20: Gradient function φ update
21: End for
22: End

The simulation scenario is set to a one-way straight road. The computing power of vehicles is distributed in $[150, 600]$ Mcycle/s, the computing power of edge servers is distributed in $[1, 7]$ Gcycle/s, and the caching capacity of edge servers is distributed in $[250, 1000]$ MB. The computing power of vehicles is distributed in $[150, 500]$ Mcycle/s, and the computing intensity of each task is 280 cycle/bit. The specific simulation parameters of IoV vehicle environment are shown in Table 1.

DDQN algorithm needs to be trained and can only be put into practical use after the neural network converges. The training curve of DDQN algorithm is shown in Fig. 2. The training process of the algorithm consists of 750 training cycles to ensure that the DDQN multi-layer network has converged before the end of training.

Table 1. Simulation parameter setting of IoV model

Parameter	Value
Edge server computing power (Gcycle/s)	$[1, 7]$
Edge server cache capacity (MB)	$[250, 1000]$
Number of vehicles (vehicle)	$[30, 60]$
Vehicle computing capacity (Mcycle/s)	$[150, 600]$
Task calculation intensity (cycle/bit)	280

As shown in Fig. 2, the task offloading algorithm proposed in this paper enters a stable state after being trained for 300 cycles, after which the reward of the algorithm only fluctuates slightly. This shows that the neural network has converged and has superior training and learning efficiency.

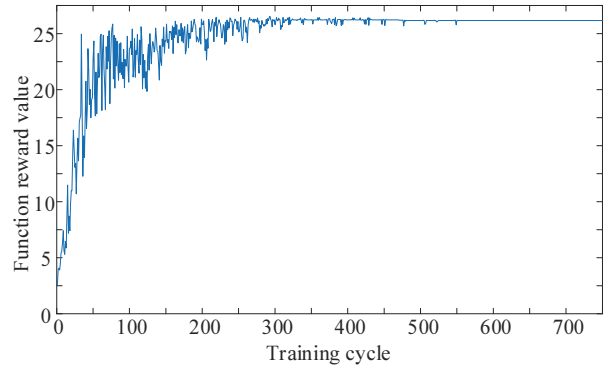

Fig. 2. IoV resource allocation algorithm based on DDQN network.

In this paper, two studies [23,24] are referred as comparable simulation experiments to analyze and discuss the performance of IoV, in order to verify that the proposed method has efficient task offloading capabilities. The analysis includes the discussion of network performance on factors such as RSU computing power, the number of vehicles, and the volume of task data.

Fig. 3 shows the impact of RSU computing power on the delay of different resource allocation methods.

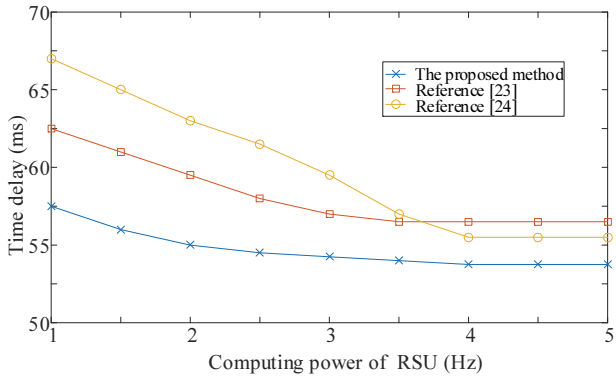

Fig. 3. Network delay variation under different computing power of RSU.

As shown in Fig. 3, the proposed method and the comparable method have the same trend, because the calculation of the task is positively correlated with the resources of the RSU. As the computing power of RSU increases, the time delay for processing tasks of different algorithms decreases. However, DDQN algorithm proposed in this paper presents a faster analyzing and processing speed than the comparative method with different computing capabilities. When the RSU computing power is 1 Hz, the proposed method reduces the network delay by 10.0 ms compared to that in [23], and reduces the delay by 18.95 ms compared to that in [24].

The reason is that the proposed method fully considers the resource offloading mode of the user layer and the edge layer in the IoV when establishing the mathematical model, and realizes the horizontal and vertical collaboration of resource allocation. However, the comparative reference lacks corresponding analysis and consideration for IoV model modeling. Therefore, in the case of the same RSU computing power, the proposed method can realize the optimal allocation of network resources; thereby, ensuring the rapid processing of tasks.

Fig. 4 shows the IoV communication performance with different vehicle numbers. The simulation results show that as the number of cars increases, the average delay of the resource allocation methods used has increased.

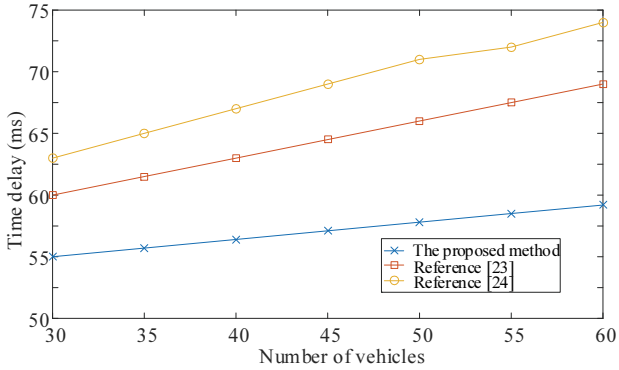

Fig. 4. Network delay variation under different numbers of vehicles.

At the same time, it can be seen from Fig. 4 that the proposed method has a smaller delay increment than comparable methods. In the simulation experiment cycle, the network delay of proposed method increases by about 4.2 ms; while methods in [23] and [24] both increase the network delay by 9 ms. It can be speculated that as the volume of IoV network further increases, [23] and [24] may cause the collapse of IoV due to excessive network delay.

Because the proposed method applies DDQN network model to IoV task offloading, DDQN can be combined with the Q-learning network based on the value function approximation of the neural network. This can avoid the overestimation and improve the efficiency of resource allocation tasks. At the same time, this paper also designs corresponding and reasonable reward functions for different stages of IoV resource allocation, further ensuring the effectiveness of resource allocation. On the other hand, there is no targeted analysis of IoV characteristics in [23] and [24]. They simply migrate the multi-layer network model to the IoV scene, which is not enough for complex and diverse actual scenes.

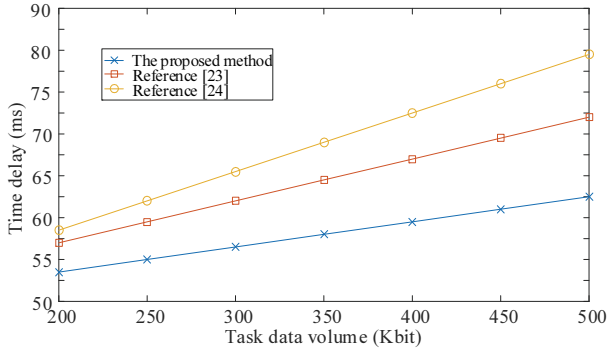

Fig. 5. Network delay variation under different task data volume.

Fig. 5 shows the network analysis efficiency with different task data volumes, and its change trend is similar to Fig. 4. As shown in Fig. 5, as the volume of IoV data increases, the network delay increases

accordingly. Due to the combined effect of DDQN multi-layer network and improved reward function, the proposed method maintains a lower network delay during the simulation cycle than comparative methods. When the network task data volume is 500 kbits, the network delay of proposed method is 62.47 ms, which is 9.85 ms and 16.46 ms lower than that in [23] and [24], respectively. This confirms that the proposed method can meet high-efficiency processing requirements of IoV in datasets of different sizes.

5. Conclusion

This paper studies the resource allocation problem in the context of IoV using deep reinforcement learning network and proposes a new allocation strategy. Besides, simulation experiments have proved the proposed strategy's superiority compared with current methods, which can support the high-quality processing demand of IoV network service applications. When constructing the IoV system model, this paper further refines the horizontal penetration and vertical cooperation structure of offloading model, and introduces the movement characteristics of vehicles to achieve a more refined model mapping. Furthermore, DDQN algorithm is used to realize the optimal allocation of IoV network resources, to avoid overestimation and to provide superior utility performance for network services.

Due to the open nature of IoV environment, many insecure factors are inevitably faced with. Vehicles can continuously send resource requests or feed false computing task execution results and other behaviors back. This has an impact on the performance of offloading computing tasks, and even poses a threat to driving safety in severe cases. The decentralization, openness, autonomy, immutability of information and the anonymity of blockchain technology are powerful solutions to network information security. In future research work, the blockchain technology can be introduced into the resource allocation task of IoV while ensuring the quality of network services, which is expected to achieve more reliable and secure network communication services.

References

[1] S. K. Singh, P. K. Sharma, Y. Pan, and J. H. Park, "BIIoVT: blockchain-based secure storage architecture for intelligent Internet of Vehicular Things," *IEEE Consumer Electronics Magazine*, 2021. https://doi.org/10.1109/MCE.2021.3089992

[2] P. K. Sharma and J. H. Park, "Blockchain-based secure mist computing network architecture for intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 5168-5177, 2021.

[3] Y. Xiao, H. Liu, and X. Cheng, "Key technologies of Internet of vehicles and their development trends and challenges," *Communications Technology*, vol. 54, no. 1, pp. 1-8, 2021.

[4] L. Pu, X. Chen, G. Mao, Q. Xie, and J. Xu, "Chimera: an energy-efficient and deadline-aware hybrid edge computing framework for vehicular crowdsensing applications," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 84-99, 2019.

[5] S. K. Singh, Y. S. Jeong, and J. H. Park, "A deep learning-based IoT-oriented infrastructure for secure smart city," *Sustainable Cities and Society*, vol. 60, article no. 102252, 2020. https://doi.org/10.1016/j.scs.2020.102252

[6] B. Fan, H. Tian, S. Zhu, Y. Chen, and X. Zhu, "Traffic-aware relay vehicle selection in millimeter-wave vehicle-to-vehicle communication," *IEEE Wireless Communications Letters*, vol. 8, no. 2, pp. 400-403, 2019.

[7] H. Zhang, Y. Cheng, K. Liu, and X. He, "The mobility management strategies by integrating mobile edge computing and CDN in vehicular networks," *Journal of Electronics & Information Technology*, vol. 42, no. 6, pp. 1444-1451, 2020.

[8] M. Sohail, L. Wang, R. Ali, S. Rahim, and J. Yao, "Efficient data handover and intelligent information assessment in software-defined vehicular social networks," *IET Intelligent Transport Systems*, vol. 13, no. 12, pp. 1814-1821, 2019.

[9] H. Yang, X. Xie, and M. Kadoch, "Intelligent resource management based on reinforcement learning for ultra-reliable and low-latency IoV communication networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4157-4169, 2019.

[10] X. Huang, K. Xu, Q. Chen, and J. Zhang, "Delay-aware caching in Internet-of-Vehicles networks," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10911-10921, 2021.

[11] J. Cheng, J. Cheng, M. Zhou, F. Liu, S. Gao, and C. Liu, "Routing in internet of vehicles: a review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 5, pp. 2339-2352, 2015.

[12] X. Wang, Z. Ning, X. Hu, L. Wang, B. Hu, J. Cheng, and V. C. Leung, "Optimizing content dissemination for real-time traffic management in large-scale Internet of Vehicle systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1093-1105, 2019.

[13] C. Chen, L. Chen, L. Liu, S. He, X. Yuan, D. Lan, and Z. Chen, "Delay-optimized V2V-based computation offloading in urban vehicular edge computing and networks," *IEEE Access*, vol. 8, pp. 18863-18873, 2020.

[14] W. Zhuang, Q. Ye, F. Lyu, N. Cheng, and J. Ren, "SDN/NFV-empowered future IoV with enhanced communication, computing, and caching," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 274-291, 2020.

[15] L. Zhang, Z. Zhao, Q. Wu, H. Zhao, H. Xu, and X. Wu, "Energy-aware dynamic resource allocation in UAV assisted mobile edge computing over social Internet of Vehicles," *IEEE Access*, vol. 6, pp. 56700-56715, 2018.

[16] J. Kim, J. Lee, S. Moon, and I. Hwang, "A position-based resource allocation scheme for V2V communication," *Wireless Personal Communications*, vol. 98, no. 1, pp. 1569-1586, 2018.

[17] F. Abbas, P. Fan, and Z. Khan, "A novel low-latency V2V resource allocation scheme based on cellular V2X communications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2185-2197, 2019.

[18] J. Wang, Q. Qi, S. Qing, and J. Liao, "Elastic vehicular resource providing based on service function-group resource mapping of smart identify network," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1897-1908, 2018.

[19] S. Wang, F. Huang, and C. Wang, "Adaptive proportional fairness resource allocation for OFDM-based cognitive radio networks," *Wireless Networks*, vol. 19, no. 3, pp. 273-284, 2013.

[20] Q. Zhang, L. Gui, F. Hou, J. Chen, S. Zhu, and F. Tian, "Dynamic task offloading and resource allocation for mobile-edge computing in dense cloud RAN," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3282-3299, 2020.

[21] J. Zhang, X. Hu, Z. Ning, E. C. H. Ngai, L. Zhou, J. Wei, J. Cheng, B. Hum and V. C. Leung, "Joint resource allocation for latency-sensitive services over mobile edge computing networks with caching," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4283-4294, 2019.

[22] S. S. Lee and S. Lee, "Resource allocation for vehicular fog computing using reinforcement learning combined with heuristic information," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10450-10464, 2020.

[23] C. Cui, M. Zhao, and K. Wong, "An LSTM-method-based availability prediction for optimized offloading in mobile edges," *Sensors*, vol. 19, no. 20, article no. 4467, 2019. https://doi.org/10.3390/s19204467

[24] M. Chen, T. Wang, K. Ota, M. Dong, M. Zhao, and A. Liu, "Intelligent resource allocation management for vehicles network: an A3C learning approach," *Computer Communications*, vol. 151, pp. 485-494, 2020.

[25] J. Zhao, M. Kong, Q. Li, and X. Sun, "Contract-based computing resource management via deep reinforcement learning in vehicular fog computing," *IEEE Access*, vol. 8, pp. 3319-3329, 2019.

[26] X. Xu, Y. Xue, X. Li, L. Qi, and S. Wan, "A computation offloading method for edge computing with vehicle-to-everything," *IEEE Access*, vol. 7, pp. 131068-131077, 2019.

[27] J. Kim, Y. Han, and I. Kim, "Efficient groupcast schemes for vehicle platooning in V2V network," *IEEE Access*, vol. 7, pp. 171333-171345, 2019.

[28] H. Peng, L. Liang, X. Shen, and G. Y. Li, "Vehicular communications: a network layer perspective," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1064-1078, 2019.

[29] Y. Liu, H. Yu, S. Xie, and Y. Zhang, "Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 11, pp. 11158-11168, 2019.

[30] M. S. Demir, H. B. Eldeeb, and M. Uysal, "Comp-based dynamic handover for vehicular VLC networks," *IEEE Communications Letters*, vol. 24, no. 9, pp. 2024-2028, 2020.

Hongqi Xi **https://orcid.org/0000-0003-2277-2768**

He has got Master of engineering, and graduated from Zhegnzhou University in 2008. He is an associate professor at Henan Finance University. His research interests include communication technology and big data.

Huijuan Sun **https://orcid.org/0000-0002-5039-8585**

She has got Master of science, and graduated from Henan University in 2008. She is an associate professor at Henan Finance University. Her research interests include optimization algorithm and big data analysis.