J Inf Process Syst, Vol.18, No.3, pp.389~401, June 2022
https://doi.org/10.3745/JIPS.02.0173

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

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Communication Resource Allocation Strategy of Internet of Vehicles Based on MEC

Zhiqiang Ma*

Abstract

The business of Internet of Vehicles (IoV) is growing rapidly, and the large amount of data exchange has caused problems of large mobile network communication delay and large energy loss. A strategy for resource allocation of IoV communication based on mobile edge computing (MEC) is thus proposed. First, a model of the cloud-side collaborative cache and resource allocation system for the IoV is designed. Vehicles can offload tasks to MEC servers or neighboring vehicles for communication. Then, the communication model and the calculation model of IoV system are comprehensively analyzed. The optimization objective of minimizing delay and energy consumption is constructed. Finally, the on-board computing task is coded, and the optimization problem is transformed into a knapsack problem. The optimal resource allocation strategy is obtained through genetic algorithm. The simulation results based on the MATLAB platform show that: The proposed strategy offloads tasks to the MEC server or neighboring vehicles, making full use of system resources. In different situations, the energy consumption does not exceed 300 J and 180 J, with an average delay of 210 ms, effectively reducing system overhead and improving response speed.

Keywords

Allocation of Communication Resources, Genetic Algorithm, Internet of Vehicles, Knapsack Problem, MEC

1. Introduction

The Internet of Vehicles (IoV) is one of the main application scenarios of 5G. In practical applications, task offloading and resource allocation between multiple devices and multiple servers are usually affected by multiple factors such as limited server resources, interference between different devices, and the location and mobility of IoV devices [1-3]. Therefore, reasonable optimization of task offloading and resource allocation is an important challenge to ensure the stable and efficient operation of IoV communication [4-6].

At present, certain research results have been obtained for the problem of vehicle networking communication resource allocation, such as time delay, optimization of energy consumption, and limited resource allocation [7,8]. As one of the key 5G technologies, mobile edge computing (MEC) can dynamically obtain real-time information in the wireless access network, and promote the development of caching technology in a more intelligent direction [9,10]. Jiang et al. [11] proposed a new bandwidth-link resource cooperative allocation strategy. Based on the predictive characteristics of the relative position of the mobile transceiver, a collaborative strategy for mobile cellular networks and dedicated

Manuscript received January 21, 2022; first revision March 21, 2022; second revision April 29, 2022; accepted May 23, 2022.
* **Corresponding Author:** Zhiqiang Ma (mazhiqiang1687@126.com)
Dept. School of Information Engineering, Henan Vocational College of Agricultural, Zhengzhou, Henan, China

short-range communications is formulated. Chen et al. [12] uses asynchronous actor critical learning algorithm to deal with the problem of computing offload in the scenario of IoVs. The experimental results show that this method is effective and can effectively improve the performance of virtual wireless network. Wang et al. [13] proposed a solution based on deep reinforcement learning to deal with the problem of computing unloading in the scenario of IoVs. The scheme maximizes the average cumulative reward, that is, minimizing the total cost, but it has poor adaptability to complex networks. Tang et al. [14] proposed a general framework for sharing and utilizing communication, computing and storage resources in the Internet of Things environment, which realizes more efficient resource allocation to a certain extent, but the scheme ignores the data transmission delay. Guan et al. [15] integrates the radar cross section of the small-scale structure into the simulator based on the high-frequency prediction technology framework. Li et al. [16] proposed a two-layer algorithm based on independent power greedy outer approximation and unified power to ensure the communication rate and service quality of the IoVs. Wei et al. [17] used a deep reinforcement learning algorithm to solve the computational offloading problem of mobile users in wireless cellular networks with MEC. The experimental results are realistic, and the strategy has achieved good results. Wang et al. [18] proposed a federated DRL-based collaborative edge caching framework to deal with the content delivery latency and the cache hit ratio when offloading repetitive traffic. However, in practical applications, due to the complexity of state-action, the Q-learning algorithm takes a long time to converge.

The processing power of the IoV equipment is difficult to meet the needs of delay-sensitive and computationally intensive applications. And most resource allocation strategies have the problem of uneven load distribution. This paper proposes a communication resource allocation strategy for the IoV based on edge computing. The main innovations of this study are summarized as follows:

(1) Due to the limited computing power of the in-vehicle cloud processor and the less power storage, the MEC server is introduced to build a cloud-side collaborative cache and resource allocation system model to strengthen the communication capabilities of the system.

(2) In order to improve the utilization of resources, the proposed strategy uses coding communication tasks and uses genetic algorithms (GA) to find the optimal allocation plan. The offloading of computing tasks to a suitable MEC server or neighboring vehicles also further reduces the total energy consumption.

The rest of this paper is arranged as follows. The second section introduces the system model and modeling process. The third section introduces the resource allocation strategy of IoV based on GA. The fourth section designs experiments to verify the strategy. The fifth section is the conclusion.

2. System Model and Modeling

For ease of reading, abbreviations of some nouns are shown in Table 1.

Table 1. Abbreviate table

Full name	Abbreviation
Internet of Vehicles	IoV
Mobile edge computing	MEC
Road side units	RSU
Genetic algorithm	GA

2.1 System Model

The system scenario is shown in Fig. 1.

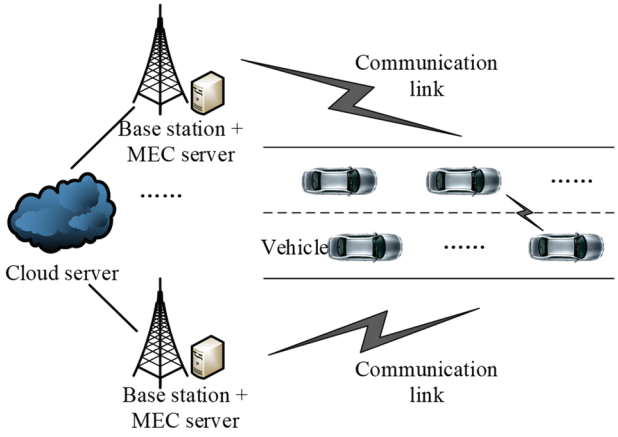

Fig. 1. Model architecture of cloud-edge collaboration for Internet of Vehicles.

We can see from Fig. 1 that in this model, L roadside units (RSU) are deployed around the road, denoted as $R = \{r_1, r_2, \cdots, r_L\}$. Each MEC server is placed next to the RSU. There are N vehicles, and their vehicle identification is $n = \{1,2, \cdots, N\}$; thus, the vehicle is denoted as $V = \{v_1, v_2, \cdots, v_N\}$. Both the MEC server and neighboring vehicles are collectively referred to as the service node $\psi = \{\varphi_1, \varphi_2, \cdots, \varphi_M\}$.

The vehicles are randomly distributed, and the collection of vehicles in cell j is $\phi_j = \{v_1, v_2, \cdots, v_n\}$. The spectrum is equally divided into K sub-channels, denoted as $k = \{1,2, \cdots, K\}$, and the bandwidth of each sub-channel is B. The set of vehicle unloading strategies is denoted as $A = \{a_1, a_2, \cdots, a_N\}$. If $a_i = 1$, it means that v_i will offload the task to the service node φ_m for calculation. If $a_i = 0$, it means that v_i is performing calculation tasks locally.

2.2 Communication Model

Communication consumption occurs when the task is offloaded to the server and the server downloads the calculation content required to complete the task. Communication consumption is not only related to the amount of data transmitted but also is related to the bandwidth and the actual transmission rate of the wireless network. It is assumed that the communication link between the vehicle and the MEC node is a frequency-flattened fast fading Rayleigh channel. Then the vehicle offloads the task to the MEC node and the MEC node returns the calculation result to the vehicle speed v_{V2I}, which can be expressed as:

$$v_{V2I} = B_{V2I}\, log_2\left(1 + \frac{P_s d^{-\delta} \tau^2}{N_0}\right) \tag{1}$$

where, B_{V2I} represents the bandwidth between the vehicle and the MEC node. P_s represents the transmit power of the transmitting device. $d^{-\delta}$ represents the path loss between the vehicle and the server node.

δ represents the path loss factor. τ^2 represents the channel fading factor of the upload link.N_0 represents the Gaussian white noise power. Therefore, the required upload delay T_i^{up} when offloading the task to the server is:

$$T_i^{up} = \frac{s_i}{v_{V2I}} \tag{2}$$

where, s_i represents the size of the task.

Assuming that the distance between the MEC server nodes is d_{MEC2}, and the distance between the edge server and the cloud server is d_{CMEC}. It can be assumed that the transmission speed between the MEC server nodes is v_{MEC2}, and the transmission speed from the cloud service to the edge server is v_{CMEC}. When task Ω_i is sent to the MEC server numbered l and has cached G_s^i cached content, the edge server needs to initiate a request for downloading missing computing content to the cloud. Therefore, the time T_i^{down} required for the service server to download the missing content can be approximately expressed as:

$$T_i^{down} = \frac{\vartheta_\Sigma^i - \vartheta_m^i}{v_\Sigma} \tag{3}$$

In the formula, ϑ_Σ^i represents the calculation content involved in the calculation process.ϑ_m^i represents the calculated content that is offloaded to the MEC server. v_Σ is the sum of the transmission speed between the MEC server nodes and the cloud service to the edge server.

Assuming that the energy consumption due to the transmission per unit time is E_{tran}, the total energy consumption E_{tran}^Σ due to the transmission is:

$$E_{tran}^\Sigma = E_{tran}\left(T_i^{down} + T_i^{up}\right) \tag{4}$$

2.3 Calculation Model

Assuming that each vehicle has a calculation task $\Omega = \{s_i, c_i, t_i^{max}\{\}\}, i \in N$ needs to be processed. Among them, s_i is the input size of task Ω_i. c_i is the number of CPU cycles required to calculate task Ω_i. t_i^{max} is the maximum time delay that can be tolerated by the calculation task Ω_i. The task will be offloaded to the MEC server or neighboring vehicles through the RSU. It can also be executed on the local vehicle.

Offload calculation

Define the cost of the calculation process of task request vehicle v_i offloading task as the service node φ_m, which is the weighted combination of delay and energy consumption, expressed as:

$$u_i^{off} = \alpha t_i^{off} + \beta e_i^{off} \tag{5}$$

where, α represents the weighting factors of delay, and β represents the energy consumption, and satisfies $\alpha + \beta \leq 1$. $t_i^{off} = \frac{s_i}{v_{V2I}} + \frac{c_i}{f_m^i}$ is the sum of the unloading delay and the calculation delay.f_m^i is

the computing resource allocated by the service node φ_m to the vehicle v_i.$e_i^{off} = P_s \frac{s_i}{v_{V2I}}$ is the energy consumption of the transmission process.

Local calculation

Assuming that the computing power of vehicle v_i is F_i^1, different vehicles have different computing powers. When Ω_i is calculated locally, the delay is represented by t_i^1, $t_i^1 = c_i/F_i^1$; the cost that the vehicle v_i needs to bear is:

$$u_i^1 = \alpha t_i^1 + \beta e_i^1 \tag{6}$$

2.4 Optimization

The binary-based task offloading scheme is adopted, that is, tasks are only allowed to be processed locally or offloaded to the server. In the network, the role of the vehicle is only to collect data or offload tasks to the server. Therefore, all tasks can only be processed in cloud servers or in edge servers. If the total number of tasks requested from vehicles is N, there is a task set $\Omega = \{\Omega_i | i = 1,2,\cdots,N\}$. For each task, Ω_i has $\Omega_i = \{a_i, Q_c^i, \vartheta_\Sigma^i, t_{max}^{i_i}\{\}\}$, where Q_c^i is the CPU resource required for execution, t_{max}^i is the maximum execution time of the task, A_i is the unloading decision of the task, ϑ_Σ^i is the calculation content involved in the calculation process and s_i represents the size of the task.

The main concern in this paper is how to realize a joint optimization of the time delay and the energy consumption required by the offloading task. For any task Ω_i, its energy consumption E_Σ is composed of transmission energy consumption E_{tran}^Σ and calculation energy consumption E_{cal}. The delay consumption T_Σ is composed of calculation delay T_{cal} and transmission delay T_{tran}. Then the optimization objective function can be expressed as:

$$\begin{aligned} &min\ O = \min_{\forall a_i}\left(\lambda E_\Sigma + (1-\lambda)T_\Sigma\right) \\ &s.t.\ \ \mathrm{C1}: T_\Sigma^i < t_i^{max} \\ &\quad \mathrm{C2}: a_i \in \{0,1\} \\ &\quad \mathrm{C3}: Q_c^i \le Q_{max}^m \end{aligned} \tag{7}$$

where, λ represents the weight of energy consumption and delay consumption. Constraint C1 represents that the delay consumption of the task cannot be greater than the maximum tolerable delay of the task. Constraint C2 means that tasks can only be offloaded to cloud servers or MEC servers. Constraint C3 represents that the computing resources required by the task cannot exceed the maximum computing resources of the MEC server.

3. Resource Allocation Strategy of IoVs Based on GA

The optimization problem is transformed into a knapsack problem, and a GA is used to solve it to meet the computational unloading requirements. Different MEC servers have different computing capabilities.

When the MEC server with insufficient computing resources undertakes a large number of computing tasks, it cannot guarantee the completion of computing tasks, and will also cause the MEC server to be overloaded, affecting the processing of other services. At the same time, the random offloading of computing tasks will lead to a lengthy number of iterations when searching for the optimal computing offloading strategy. Therefore, a task offloading strategy based on GA is used to reserve suitable initial chromosomes. We combine the greedy algorithm with the GA to find the optimal offloading strategy for each MEC server; thus, speeding up the iterative process.

3.1 Coding and Initial Population Setting

Using binary coding, the coding of each chromosome is an offloading strategy A. $A_{(q)}$ represents the value of the q bit in the strategy A, and $A_{(q)}$ takes 0 or 1. Let $[X_{iM}, X_{iM+1}, \cdots, X_{(i+1)M}]$ denote the unloading result of computing task Ω_i, the number a_i of the uninstalled MEC server is:

$$a_i = \sum_{q=i\kappa}^{(i+1)\kappa} A_{(q)} \times 2^{(i+1)\kappa - q} \tag{8}$$

where, κ represents the number of code bits.

Fig. 2 shows the coding process of computing task strategy for no more than 8 MEC servers.

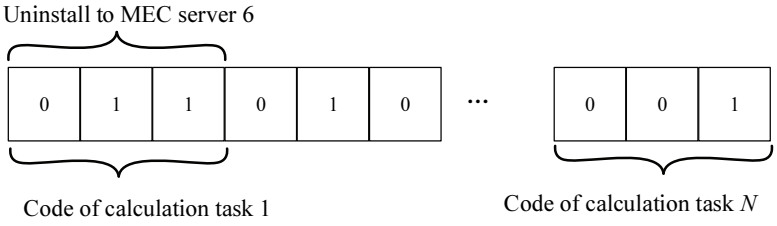

Fig. 2. Mapping of strategy code and MEC server.

Initialize the relevant parameters of the GA, including the maximum number of iterations I, chromosome length $N \times M$, mutation probability P_m, crossover probability P_c, population size Z and reserved population Z'. Most GA select the initial population randomly. The proposed strategy combines pre-defined chromosomes and random chromosome algorithms for population initialization. In the process of pre-defining chromosomes, classify the chromosomes with the highest weight. That is, the on-board computing task encoding is set as follows:

$$\sum_{q=i\kappa}^{(i+1)\kappa} A_{(q)} \times 2^{(i+1)\kappa - q} = \Omega_i \tag{9}$$

3.2 Fitness Function

The fitness f_i of each unloading scheme in the proposed strategy is the reciprocal of the objective function. The higher the fitness function value, the smaller the value of the objective function model. It

means that when the overall cost of the computational offloading scheme decreases, the offloading scheme presents better performances. The specific calculation formula of f_i is as follows:

$$f_i = \frac{1}{\min\limits_{\forall a_i}\left(\lambda E_{\sum} + (1-\lambda)T_{\sum}\right)} \tag{10}$$

After the fitness function is calculated, the general chromosomes with poor fitness are eliminated according to the calculated value, (i.e., the unloading plan), and the chromosomes with higher fitness are retained. In this way, the quality of the chromosomes after several iterations is increasingly higher. In other words, the overall cost of the uninstall scheme is becoming less.

3.3 Genetic Operator Operation

(1) Select operation: The chromosomes are selected according to the selection probability, and the above-mentioned individuals are regarded as the first generation. Using the roulette random selection method proportional to fitness, assuming the fitness of the individual is f_i, the probability of i being selected is:

$$P_i = f_i / \sum_{i=1}^{n} f_i \tag{11}$$

For the initialized population, calculate the fitness value of each chromosome and its probability of being selected for comparison, and eliminate the chromosome with the lowest probability. The chromosome with the highest probability is selected for copying, instead of the chromosome that has been eliminated.

(2) Cross operation: Using a one-point crossover method, the crossover probability is P_c.

(3) Mutation operation: For the proposed optimization problem, the mutation operation is to change the mutation bit 1 of the chromosome to 0, and 0 to 1. The other bits remain unchanged, and the probability of mutation is P_m. Therefore, select a variant position for mutation. Then calculate whether its fitness is greater than or equal to its original fitness. If not, re-select the variant position for mutation.

3.4 Termination Condition

The GA requires the setting of termination conditions. Therefore, the fitness function value needs to be calculated after each chromosome mutation. If the set maximum number of iterations has been reached or the fitness value has remained unchanged for a long time, the entire GA process is terminated. That is, the global optimal solution that meets the conditions is obtained; otherwise, the iteration continues.

4. Experiment and Analysis

The simulation experiment of the proposed strategy is carried out on the MATLAB platform (MATLAB2012a). Considering that there are 5 cells on the roadside, each cell is configured with RSU and MEC server. The specific simulation parameters are shown in Table 2.

Table 2. Simulation parameters

Parameter	Value
Maximum transmission power of vehicle	30 dBm
System bandwidth	30 dBm
Task calculation size	15–30 MB
Calculate the CPU cycles required for the task	1500–3000
Gaussian white noise power	-75 dBm
Weight factor $\alpha = \beta$	0.45
Coverage diameter of RSU	1 km
Vehicle computing power	2–4 GHz
Computing power of MEC server	15 GHz
Number of uplink transmission channels	15 strip
Number of vehicles	100
Vehicle moving speed	40–70 km/hr
Vehicle buffer capacity	100 MB
Cache capacity of MEC server	500 MB

4.1 Effect of Iteration Number on the Task Completion Rate

First, verify the impact of the number of iterations of the proposed strategy on the task completion rate. The task completion rate is expressed as the proportion of successfully calculated computing tasks to the total computing tasks. Through repeated experiments, the effect of the number of iterations on the task completion rate under different vehicle-mounted terminals is studied. The result is shown in Fig. 3.

It can be seen from Fig. 3 that the proposed strategy can reach a plateau after effective iterations. It tends to converge about 250 times. And the larger the number N of vehicle-mounted terminals, the lower the completion rate of the task. When N increases from 10 to 50, the task completion rate drops from 100% to 73%. As the number of vehicle-mounted terminals increases, the amount of computing tasks increases. In the case of limited computing resources, data processing capabilities will decline. As a result, the task completion rate has dropped.

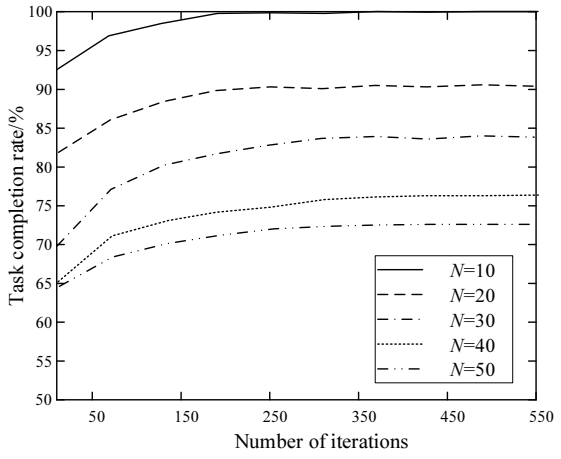

Fig. 3. Influence of iteration times on task completion rate.

4.2 Performance Comparison with Other Methods

In order to demonstrate the performance of the proposed strategy, it is compared with [11,12,17]. The relationship between the number of vehicles and the total cost is shown in Fig. 4.

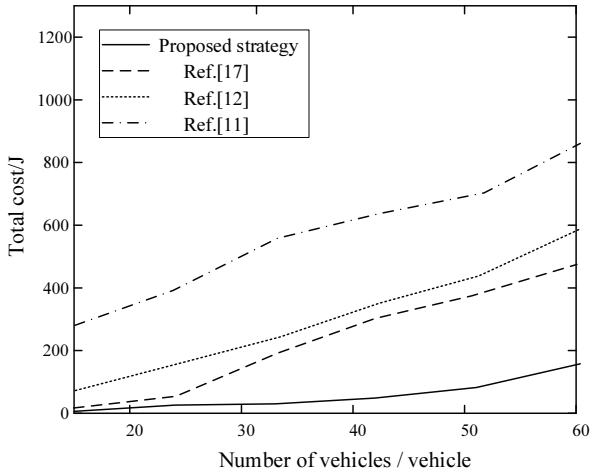

Fig. 4. Relationship between the number of vehicles and the total cost.

It can be seen from Fig. 4 that as the number of vehicles increases, the total cost increases. The proposed strategy makes full use of the surrounding idle resources, reasonably allocating computing and communication resources, and the cloud-side collaborative offloading can effectively improve resource utilization. The caching mechanism can avoid double calculations and reduce network congestion, thereby reducing system overhead. When the number of vehicles is 60, the total calculation cost is only about 180 J.

Similarly, the relationship between the amount of task data and the total cost of different strategies is shown in Fig. 5.

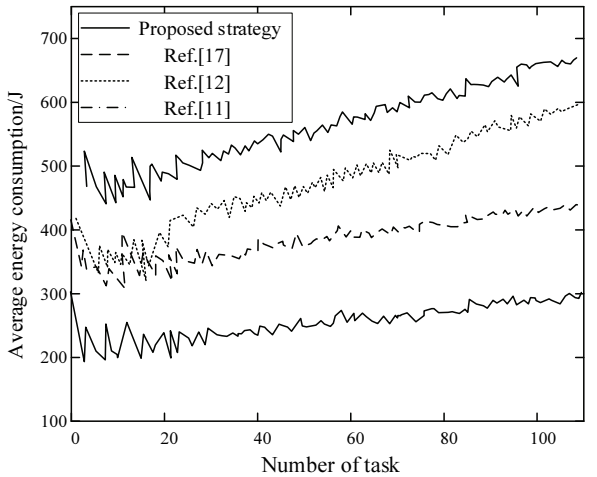

Fig. 5. Relationship between task data volume and total cost.

As can be seen from Fig. 5, the total overhead increases with the increase of the amount of task data. Among several comparison strategies, the total cost of the proposed strategy is the smallest, and the cost growth rate is relatively slow. When the task volume is 30 MB, the total overhead is about 300 J. Because it adopts the cloud-side collaborative offloading method, the GA selects the best resource allocation mode, which can provide resource utilization. In the case of a large number of tasks, low computational overhead can be guaranteed. Wei et al. [17] uses a deep reinforcement learning algorithm to solve the communication resource allocation problem of mobile users in a wireless cellular network with MEC. The learning algorithm used is relatively complicated. When the number of tasks increases, the computational overhead will increase significantly, exceeding 600 J. Chen et al. [12] only uses the asynchronous advantage actor-critic learning algorithm for communication resource allocation. But it relies too much on the processing power of the vehicle itself. Therefore, the total overhead is higher when the number of tasks increases. Jiang et al. [11] allocates resources through the cooperation of mobile cellular network and is dedicated to short-range communication, lacking the optimization of the allocation scheme. Therefore, the overall cost remains high, exceeding 800 J.

In order to verify the delay and energy consumption performance of the proposed model, an experiment is designed to verify the performance of different strategies with the number of tasks as the variable. The experimental results are shown in Fig. 6.

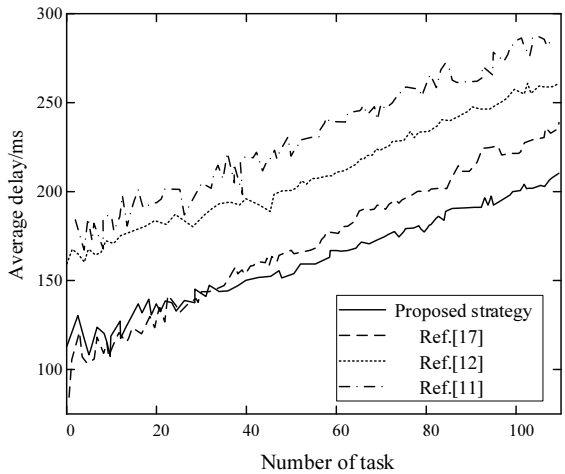

Fig. 6. Comparison of task number and average delay of different strategies.

It can be seen from Fig. 6 that the delay consumption of the strategy in [11] is higher than that of other strategies. It relies on a dedicated short-range communication network, but the vehicle has a wide range of movement, and this network architecture cannot meet the communication needs of the IoVs. Therefore, the time delay is as high as 280 ms. In [12,17], both use learning algorithms to optimize the communication resource allocation plan. But the strategy in [17] is based on MEC deployment, offloading tasks to the MEC server with stronger processing capability, which can reduce the delay by about 20 ms. The proposed strategy is based on the cloud-edge coordinated car networking and uses GA to solve the optimal resource allocation plan. It not only makes full use of the powerful processing capabilities of MEC, but also combines with the optimization capabilities of GA. Therefore, the average delay is relatively small, the highest is only 210 ms.

Similarly, for different number of tasks, the average energy consumption of the four strategies is shown in Fig. 7.

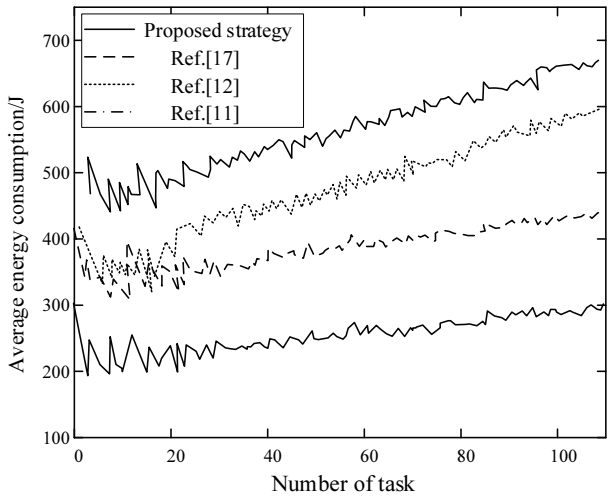

Fig. 7. Comparison of task number and average energy consumption of different strategies.

It can be seen from Fig. 7 that compared with other strategies, the average energy consumption of the proposed strategy is the smallest. When the number of tasks is 60, its average energy consumption is 300 J. Its use of cloud-side collaborative offloading can effectively improve MEC resource utilization. And the caching mechanism can avoid repeated calculations and reduce network congestion, thereby reducing system energy consumption. The authors of [11] only uses changes in network to optimize resource allocation, unable to adapt to the IoVs environment with a large amount of data. Therefore, the energy loss is large. In [12,17], both use learning algorithms to achieve communication resource allocation. However, the learning algorithm is more complicated, and the computational energy loss is large.

5. Conclusion

The IoVs is facing the impact of massive data traffic. In order to realize the real-time response of the IoVs business and to reduce energy consumption. A strategy for resource allocation of IoV communication based on edge computing is proposed. Based on the cloud-side collaborative cache and the resource allocation system of the IoVs, the GA is used to solve the optimization objective function that minimizes time delay and energy consumption. In order to find the best resource allocation plan, experimental analysis based on the built IoV model shows that under different system parameters, the system overhead and the average delay of our proposed strategy are 300J and 210ms, respectively. They are superior to other strategies and provide a certain theoretical basis for practical applications. In future work, we will continue to study the MEC-based V2X (Vehicle-to-Everything) communication resource allocation strategy. At the same time, the situation where the link is disconnected due to the high-speed movement of the vehicle and the task backhaul fails should also be addressed. It is suggested to build a V2X resource allocation framework based on vehicle location prediction.

Acknowledgement

This work is supported by application and research of teaching diagnosis and improvement based on information system fragmentation in the construction of "Shuang Gao Xiao" (No. 2019SJGLX704).

References

[1] B. He and T. Li, "An offloading scheduling strategy with minimized power overhead for Internet of vehicles based on mobile edge computing," *Journal of Information Processing Systems*, vol. 17, no. 3, pp. 489-504, 2021.

[2] S. Ullah, K. Kim, A. Manzoor, L. U. Khan, S. A. Kazmi, and C. S. Hong, "Quality adaptation and resource allocation for scalable video in D2D communication networks," *IEEE Access*, vol. 8, pp. 48060-48073, 2020.

[3] T. Park and K. I. Hwang, "Receiver protection from electrical shock in vehicle wireless charging environments," *Journal of Information Processing Systems*, vol. 16, no. 3, pp. 677-687, 2020.

[4] M. Wen, J. Park, and K. Cho, "A scenario generation pipeline for autonomous vehicle simulators," *Human-centric Computing and Information Sciences*, vol. 10, article no. 24, 2020. https://doi.org/10.1186/s13673-020-00231-z

[5] L. Ferdouse, A. Anpalagan, and S. Erkucuk, "Joint communication and computing resource allocation in 5G cloud radio access networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 9122-9135, 2019.

[6] S. Kim and Y. Yoon, "ACP model for vehicle monitoring based on CPS," *Human-centric Computing and Information Sciences*, vol. 11, article no. 5, 2021. https://doi.org/10.22967/HCIS.2021.11.005

[7] L. Yang, C. Zhong, Q. Yang, W. Zou, and A. Fathalla, "Task offloading for directed acyclic graph applications based on edge computing in industrial Internet," *Information Sciences*, vol. 540, pp. 51-68, 2020.

[8] 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

[9] Y. Chen, N. Zhang, Y. Zhang, and X. Chen, "Dynamic computation offloading in edge computing for Internet of Things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4242-4251, 2019.

[10] X. Zhang, J. Zhang, Z. Liu, Q. Cui, X. Tao, and S. Wang, "MDP-based task offloading for vehicular edge computing under certain and uncertain transition probabilities," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3296-3309, 2020.

[11] X. Jiang, K. Li, H. Jiang, N. Zhu, and X. Tong, "A bandwidth-link resources cooperative allocation strategy of data communication in intelligent transportation systems," *China Communications*, vol. 16, no. 4, pp. 234-249, 2019.

[12] 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.

[13] H. Wang, H. Ke, G. Liu, and W. Sun, "Computation migration and resource allocation in heterogeneous vehicular networks: a deep reinforcement learning approach," *IEEE Access*, vol. 8, pp. 171140-171153, 2020.

[14] M. Tang, L. Gao, and J. Huang, "Communication, computation, and caching resource sharing for the Internet of Things," *IEEE Communications Magazine*, vol. 58, no. 4, pp. 75-80, 2020.

[15] K. Guan, D. He, B. Ai, D. W. Matolak, Q. Wang, Z. Zhong, and T. Kurner, "5-GHz obstructed vehicle-to-vehicle channel characterization for Internet of intelligent vehicles," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 100-110, 2019.

[16] R. Li, P. Hong, K. Xue, M. Zhang, and T. Yang, "Energy-efficient resource allocation for high-rate underlay D2D communications with statistical CSI: a one-to-many strategy," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4006-4018, 2020.

[17] Y. Wei, Z. Wang, D. Guo, and F. R. Yu, "Deep Q-learning based computation offloading strategy for mobile edge computing," *Computers, Materials & Continua*, vol. 59, no. 1, pp. 89-104, 2019.

[18] X. Wang, C. Wang, X. Li, V. C. Leung, and T. Taleb, "Federated deep reinforcement learning for Internet of Things with decentralized cooperative edge caching," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9441-9455, 2020.

Zhiqiang Ma **https://orcid.org/0000-0002-6181-4281**

He has got Master of Software Engineering and graduated from Zhengzhou University in 2005. He is an associate professor at Henan Vocational College of Agricultural. His research interests include Internet of things and cloud computing.