

Research Article

Computing Resource Allocation Strategy Based on Mobile Edge Computing in Internet of Vehicles Environment

Deng Gao 

School of Artificial Intelligence, Hunan Vocational College of Science and Technology, Changsha, Hunan 410004, China

Correspondence should be addressed to Deng Gao; gaodeng@hnu.edu.cn

Received 18 April 2022; Revised 23 May 2022; Accepted 31 May 2022; Published 22 June 2022

Academic Editor: Fazli Wahid

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

The rapid growth of the Internet of Vehicles applications leads to unreasonable resource allocation, task computing delay, and large energy consumption. To solve this problem, this paper proposes a computing resource allocation strategy based on mobile edge computing in the Internet of Vehicles environment. Firstly, we analyze the process of mobile edge computing network and resource allocation. Then, it is improved by introducing the Halton sequence into the traditional genetic algorithm, and the difference between populations is reduced by canceling the randomness to generate an initial population with smaller individual differences. Furthermore, by optimizing the parameter setting method of traditional genetic algorithm, the crossover probability and mutation probability that change dynamically with fitness value are given, which improves the accuracy of the algorithm. Finally, the simulation results show that under the premise of the same number of tasks, the average delay and total cost of the proposed strategy are also the smallest. When the number of tasks is 45, the average delay of the proposed method is 0.32 s and the total cost is 0.34, which are better than the comparison method. Simulation results show that the improved algorithm has more advantages in delay and overhead.

1. Introduction

Nowadays, with the rapid development of communication technology, the increasing number of smart devices increases the burden of network transmission and makes the data transmission delay more and more obvious [1, 2]. The proposal of mobile edge computing solves the computationally intensive computing tasks to a certain extent [3–5]. The proposal of the Internet of Vehicles (IoV) system realizes a wireless connection between the drive test unit and pedestrians. The in-vehicle business is composed of various computing tasks, and the calculation results generated by different computing tasks will be sent to in-vehicle terminals, which is the basic basis for ensuring the normal operation of vehicles [6–8]. In process of computing tasks, resource allocation plays an important role. A reasonable resource allocation strategy can accelerate the result generation of computing tasks and improve the security of vehicle-mounted terminals [9, 10].

At present, a mobile edge computing server is usually configured in an IoV system to reduce the delay of vehicle-

to-infrastructure and vehicle-to-vehicle services, and a reasonable resource allocation strategy is used to save the energy consumption of mobile terminals and the delay caused by computing tasks, so as to achieve vehicle safety [11–13]. Reference [14] aimed at the dynamic allocation of IoV resources, and the target problem was transformed into a semi-Markov decision model by introducing resource retention and a secondary allocation mechanism. However, this method did not solve the problem of resource usage and system optimization when vehicle terminal resources were limited. Reference [15] aimed at the resource block conflict in the resource allocation method of the IoV system, based on the characteristics of the driving direction of vehicles, the interference between vehicles in opposite directions is reduced by dividing resource pools. Besides, the collision probability of resource reselection is reduced by having each resource block carry location information. However, it is difficult for this method to obtain accurate channel state information in a high-speed moving vehicle environment. Aiming at the problem that the resource allocation method

in the IoV system is not suitable for video semantics and other services, reference [16] gave a video semantic-driven resource allocation guidance model and a corresponding solution method with target detection as the objective function. They proposed a resource allocation algorithm based on reinforcement Q-learning by constructing a resource allocation optimization model. However, this method has only been studied for services such as video semantics, which has certain limitations. Reference [17] analyzed the optimal channel matching problem with the objective function of maximizing the utilization rate of spectrum resources, aiming at the problem of terminal equipment and insufficient capacity of the IoV system to meet the basic service quality of IoV communication. The problem was solved using KM (Kuhn Munkres) algorithm. But this method did not consider wireless delay, network delay, and computational delay. Reference [18] established a Markov decision process model with the objective function of maximizing total channel capacity and the successful delivery rate of load by considering the V2V link as a separate individual. The SAC reinforcement learning algorithm was used to solve the model, and the optimal allocation of V2V link spectrum resources was realized. However, the convergence rate of this method was slow. Reference [19] divided the tasks generated by vehicles in view of problems such as the differentiated resource allocation mechanism in the in-vehicle cloud system. The divided subtasks were downloaded to an on-board cloud system and executed collaboratively by different vehicles, thereby improving the offloading efficiency of tasks. On this basis, the DiffServ model was used to sort and process tasks according to task pricing, and the optimization of resource allocation was realized. But this method did not reduce the impact of resource mobility on vehicle cloud performance. Reference [20] aimed at the problem of limited Vehicle Fog Computing (VFC) resources in the IoV system, which cannot allow multiple vehicles to simultaneously offloading computing tasks from the cloud to the network edge and formulated a corresponding fog resource allocation strategy by using parked vehicles. The optimization of the resource allocation strategy was achieved by introducing reinforcement learning into the heuristic algorithm. However, this method increased task delay while reducing energy loss, and the success rate of the task needs to be improved [21–25].

Based on the above-given analysis, aiming at the problems of unreasonable resource allocation, large task computing delay, and energy consumption brought about by the rapid growth of IoV applications, this paper proposes a computing resource allocation strategy based on mobile edge computing in the IoV environment. The network model and allocation strategy based on mobile edge computing are constructed, and the model is solved by improving the genetic algorithm. The proposed method is simulated and analyzed by simulation experiments. The experimental results show that the introduction of the Halton sequence into a genetic algorithm can reduce the differences between individuals and strengthen the global search ability of the algorithm. Moreover, dynamically and adaptively changing the crossover probability and mutation

probability can solve the problem of the local optimal solution and improve algorithm accuracy. Compared with traditional computing resource allocation strategies, the innovation of the proposed method lies in the following statements:

- (1) The delay and energy consumption cost of local processing and mobile edge computing server processing are comprehensively considered, and the total task cost and calculation method are quantified
- (2) Halton sequence is introduced in process of generating the initial population by genetic algorithm, which reduces the differences between individuals in the population
- (3) By giving the crossover probability and mutation probability that change dynamically with fitness value, the local optimization problem is solved and calculation accuracy is improved

2. System Model

The mobile edge computing network model based on the drive test unit in the IoV environment is shown in Figure 1.

It is assumed that each drive test unit has R independent computing execution units, and the arrival of mobile terminal request to uninstall the application follows the Poisson formula with an average rate of λ . The specific process is as follows:

- (1) If a move request runs in the drive test unit, the end of the task will follow an exponential distribution with rate η_a , and the average service time of this move request in the drive test unit is $1/\eta_a$. In the model shown in Figure 1, the CPU that each task request is allowed to be assigned in the drive test unit is a fixed value C_a .
- (2) If the system decides to complete tasks locally, the end of tasks will follow an exponential distribution with a rate η_b , and the average service time of this movement request in the drive test unit is $1/\eta_b$. At this time, each vehicle in Figure 1, the execution of mobile applications is supported with a fixed number of C_b CPUs.

The specific process of resource allocation decision-making is shown in Figure 2.

As shown in Figure 2, when a new request arrives, the system decides whether to accept it or not based on the current state. If the request is acceptable, the system assigns a new service request to the drive test unit or offloads it locally. The ultimate goal is to decide how the task is offloaded, i.e., whether to accept mobile service requests. If the request is accepted, it is necessary to finalize how many resources the running mobile application should allocate to maximize system benefits.

3. Problem Definition

3.1. Local Processing Cost. Using (x_u, y_u) to represent the location information of u vehicle and using (x_v, y_v) to represent the location information of v mobile edge

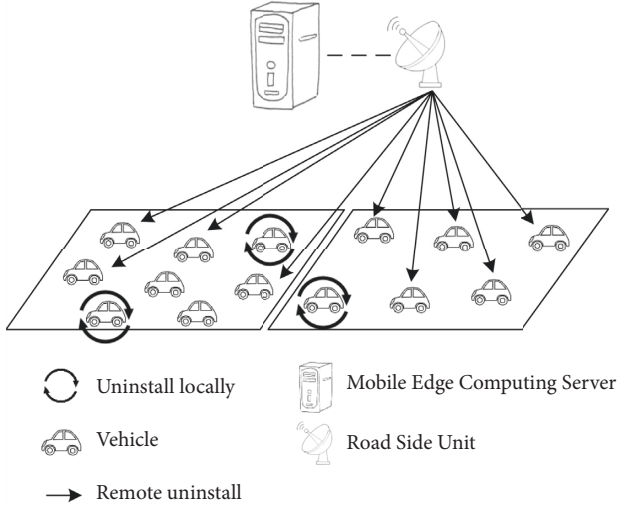


FIGURE 1: Mobile edge computing network based on drive test unit in IoV environment.

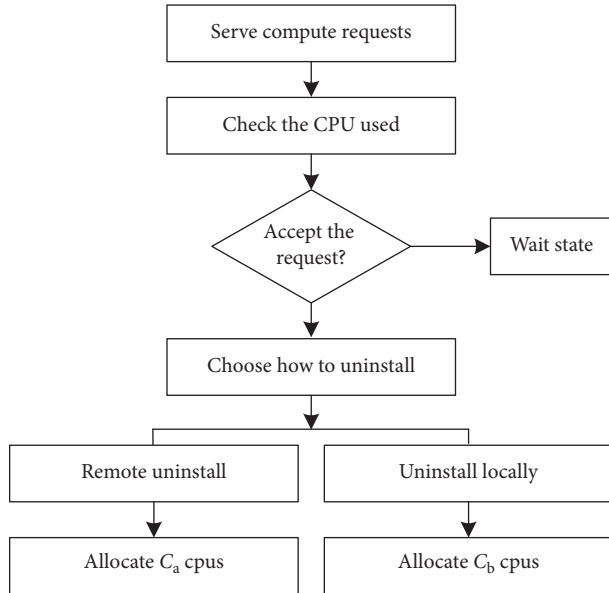


FIGURE 2: Decision-making process for resource allocation.

computing server. For the local offloading request, when $x_{u,v} = 0$, the task is processed by the vehicle itself and the cost of the vehicle task is mainly composed of the following two parts.

3.1.1. The Time It Takes for the Vehicle Itself to Complete Tasks. For local task processing, the processing delay of tasks is mainly generated by the time of the u vehicle itself to process tasks, so the processing delay of local tasks is shown in the following formula:

$$D_{u,v}^b = \frac{S_{u,v} \cdot Q}{J_u^b}. \quad (1)$$

Here, $D_{u,v}^b$ represents the processing delay of the local task of the u vehicle. $S_{u,v}$ represents the data size of the task.

Q represents the number of turns that the CPU rotates to process 1 bit data. J_u^b represents the computing power of the u car itself.

3.1.2. the Energy Consumption Required by Vehicle to Complete Tasks. For the local task processing, the energy consumed by task processing is mainly generated by the process of the u vehicle itself processing tasks, so the processing energy consumption of local task is shown in the following formula:

$$E_{u,v}^b = S_{u,v} \cdot Q \cdot E_c. \quad (2)$$

Here, $E_{u,v}^b$ represents the processing energy consumption of the local task of the u vehicle. E_c represents the energy consumed by the CPU to make one revolution.

3.2. Processing Cost of Mobile Edge Computing Servers. For the mobile edge computing server, when $x_{u,v} \neq 0$; that is, the u vehicle decides to offload some tasks to the v mobile edge computing server for processing. The entire task processing process includes three processes: offloading the task, moving edge computing server to process the task, and feeding back task result. For the u vehicle, the delay and energy consumption are also generated by these following three processes:

- (1) The u vehicle uploads some tasks and related data to the v mobile edge computing server, resulting in a delay.

In process of task offloading, when the u vehicle decides to offload some tasks to the mobile edge computing server, the user needs to upload the relevant tasks and data to the mobile edge computing server. Therefore, the delay of task offloading can be obtained from the task size and transmission rate, and the specific process is shown in the following formula:

$$D_{u,v}^a = \frac{x_{u,v} \cdot S_{u,v}}{\mu_{u,v}^p}. \quad (3)$$

Here, $D_{u,v}^a$ represents the delay of task offloading to the v mobile edge computing server. $x_{u,v}$ represents the task offloading decision for the u vehicle. $\mu_{u,v}^p$ represents the data transmission rate of uplink between the u vehicle and the v mobile edge computing server, and the data transmission rate can be calculated by the following formula:

$$\mu_{u,v}^p = B_{u,v} \log_2(1 + k). \quad (4)$$

Here, $B_{u,v}$ represents the link channel bandwidth allocated between the u vehicle and the v mobile edge computing server. k represents the signal-to-noise ratio, which can be calculated by the following formula:

$$k = \frac{P_R}{P_I + P_N}. \quad (5)$$

Here, P_R represents the received power, P_I represents the interference power, and P_N represents the noise power.

In process of mobile edge computing server processing tasks, after receiving the task and data of the u vehicle, the mobile edge computing server will allocate certain computing and communication resources to process tasks. The time delay is also generated from this, as shown in the following formula:

$$D_{u,v}^c = \frac{x_{u,v} \cdot S_{u,v} \cdot Q}{J_v^c}. \quad (6)$$

Here, $D_{u,v}^c$ represents the delay in allocating computing and communication resources to process tasks. J_v^c represents the computing resources allocated by the mobile edge computing server to the current task, and its value satisfies the constraints shown in the following formula:

$$\sum_{v=1}^V J_v^c \leq J_C. \quad (7)$$

Here, J_C represents the total computing resources of the mobile edge computing server.

After the mobile edge computing server processes the task, the task result is fed back to the u vehicle. Thus, the resulting delay is the time for the u vehicle to receive task results from the downlink, which can be expressed by the following formula:

$$D_{u,v}^d = \frac{\lambda \cdot x_{u,v} \cdot S_{u,v}}{\mu_{u,v}^d}. \quad (8)$$

Here, $D_{u,v}^d$ represents the time delay for the u vehicle to receive task results from the downlink. λ represents the ratio of output data to input data, which can generally be set to 0.1. $\mu_{u,v}^d$ represents the data transmission rate of downlink, and the transmission rate of uplink and the downlink can be considered to be the same for the u vehicle.

It can be seen from formula (8) that the total delay of mobile edge computing server to complete tasks is shown in the following formula:

$$D_{u,v}^E = D_{u,v}^a + D_{u,v}^c + D_{u,v}^d. \quad (9)$$

- (2) The v mobile edge computing server allocates part of communication and computing resources to process tasks and generates energy consumption for processing tasks. Since the energy consumption of the mobile edge computing server is not considered in the set vehicle edge network, only the energy consumed by the first vehicle to complete the task needs to be considered. The process of generating energy

consumption is mainly composed of three parts: the energy consumption of task offloading of the u vehicle, the computing energy consumption of the u vehicle completing the local task, and the energy consumption of the u vehicle receiving result feedback.

During the process of offloading tasks from the u vehicle, when the u vehicle decides to offload some tasks to the mobile edge computing server, the relevant data needs to be transmitted to the mobile edge computing server. The energy consumption generated by this process is shown in the following formula:

$$E_{u,v}^a = x_{u,v} \cdot S_{u,v} \cdot E_0. \quad (10)$$

Here, $E_{u,v}^a$ represents the energy consumed by the transmission of related data to a mobile edge computing server. E_0 represents the energy consumed to transmit 1 bit data.

After the u vehicle uploads part of the tasks to the mobile edge computing server, the remaining part of the tasks needs to be executed locally, thus generating energy consumption. This part of energy consumption can be obtained from the remaining task size and the unit data processing energy consumption, and the specific process is shown in the following formula:

$$E_{u,v}^c = (1 - x_{u,v}) \cdot S_{u,v} \cdot Q \cdot E_c. \quad (11)$$

Here, $E_{u,v}^c$ represents the energy consumption generated by executing the remaining partial tasks locally. E_0 represents the energy consumed to transmit 1 bit data.

When a mobile edge computing server processes the task, it needs to feedback task results to the u vehicle and the u vehicle also needs to consume energy to receive task result. The specific process of solving this part of energy consumption is shown in the following formula:

$$E_{u,v}^d = \delta \cdot x_{u,v} \cdot S_{u,v} \cdot E_0. \quad (12)$$

Here, $E_{u,v}^d$ represents the energy consumed by the u vehicle to receive the feedback task result. δ represents the path loss factor, which can generally be set to -4 .

It can be seen from formula (12) that the total energy consumption of the u vehicle to complete the task is generated by the above-given three processes, and the calculation method of total energy consumption can be obtained as shown in the following formula:

$$E_{u,v} = E_{u,v}^a + E_{u,v}^c + E_{u,v}^d. \quad (13)$$

3.3. Total Cost and Problem Definition. In the proposed mobile edge computing network based on the drive test unit in the IoV environment, the delay cost and energy consumption cost are comprehensively considered. Therefore, the total cost of the system can be defined as the weighted sum of total energy consumption and total delay, thus forming a new cost objective function as shown in the following formula:

$$W_{u,v} = \beta_1 \cdot D_{u,v} + \beta_2 \cdot E_{u,v}. \quad (14)$$

Here, β_1 and β_2 represent the weights of delay and energy consumption, respectively, and have $\beta_1 + \beta_2 = 1$. The proposed network has higher requirements for delay, so it is set as $\beta_1 = 0.8$, $\beta_2 = 0.2$.

For the delay $D_{u,v}$ of the u vehicle, it is necessary to comprehensively consider the processing time of the local task and the processing time of some tasks uploaded to the mobile edge computing server, as shown in the following formula:

$$D_{u,v} = (1 - x_{u,v})D_{u,v}^b + D_{u,v}^E. \quad (15)$$

The purpose of considering delay and energy consumption is to minimize the total system cost by jointly optimizing the task offloading decision $x_{u,v}$ of the u vehicle and the resource allocation strategy $(B_{u,v}, J_v^c)$ of the mobile edge computing server. Therefore, we can consider this joint optimization problem and propose the corresponding objective function as shown in the following formula:

$$\min_{(x_{u,v}, B_{u,v}, J_v^c)} \sum_u \sum_v \{\beta_1 \cdot D_{u,v} + \beta_2 \cdot E_{u,v}\}. \quad (16)$$

The total cost of the system can be minimized by finding the optimal task offloading $x_{u,v}$ and resource allocation strategy $(B_{u,v}, J_v^c)$ of mobile edge computing server, but this needs to consider the constraints of limited communication and computing resources. Considering the proposed constraints comprehensively, it is shown in the following formula:

$$\begin{cases} x_{u,v} \in [0, 1], & \forall u \in U, \\ D_{u,v} \leq \sigma_{\max}, & \forall v \in V, \\ 0 \leq \sum_v J_v^c \leq J_C, & \forall v \in V, \\ 0 \leq \sum_u B_{u,v} \leq B_S, & \forall u \in U. \end{cases} \quad (17)$$

Here, the first constraint indicates that the value of offloading strategy is a value between 0 and 1. The second constraint indicates that the task must meet the delay requirement, and σ_{\max} indicates the maximum tolerated delay. The third constraint shows that the sum of computing resources allocated by all tasks cannot exceed the total computing resources of the mobile edge computing server. The fourth constraint guarantees that the sum of bandwidth allocated by all tasks cannot exceed the total channel bandwidth.

4. Design of Adaptive Improved Genetic Algorithm

4.1. Genetic Algorithm. A genetic algorithm is an intelligent algorithm that simulates natural selection and biological evolution in nature and is a computational model that

simulates Darwin's theory of evolution. According to Darwin's theory of evolution, the evolution of biological species follows the principle of "natural selection, survival of fittest," and there is fierce competition within and between species. Only adaptive individuals can survive. The genetic algorithm searches for the optimal solution by simulating the biological evolution process, which transforms the solving process of the problem into the selection, crossover, and mutation process of chromosomal genes. Only individuals with strong adaptability to the environment can retain it with a high probability.

The genetic algorithm regards the feasible solution set space as a population, and each feasible solution corresponds to an individual. Through continuous selection, crossover, and mutation operations, a group of individuals more adapted to the environment is generated. After several iterations, individuals who do not adapt to the environment are gradually eliminated by the environment. The individual with the largest fitness function value is the optimal solution of the problem. The flow of the genetic algorithm is shown in Figure 3.

4.2. Genetic Algorithm Improvement. In process of solving the traditional standard genetic algorithm, the generation of the initial population of the algorithm and the setting of parameters such as genetic operators are relatively random. After many iterations, excellent individuals will occupy the entire population and it is difficult to generate new individuals, resulting in premature convergence of the algorithm and easy to fall into local optimum. At this time, the characteristics of the population remain basically unchanged, so it is considered to improve the traditional genetic algorithm to prevent the algorithm from falling into the local optimal solution.

Genetic algorithms generate different populations through crossover and mutation operations, so it is considered to generate initial populations with less difference by introducing Halton sequences. Strengthening the global search ability of the algorithm as much as possible and improving the search accuracy, thereby preventing the algorithm from prematurely converging and falling into a local optimum.

The first step of the genetic algorithm is to generate the initial population, which is to be able to obtain information of the target. This step is essential and very important in algorithm operation and often has a great impact on the results of algorithm solution. Generally, many individuals with uneven distribution are randomly generated by a computer when the population is initialized. However, the uniform distribution of individuals in the initial population has more optimization ability than the nonuniform distribution, and it is also more reasonable in algorithm search. Therefore, by introducing the Halton sequence, the individual differences between the populations become smaller, and the population differences are reduced by canceling the randomness.

In the Halton sequence, the generated initial population can be represented by base j , where j is a prime number and

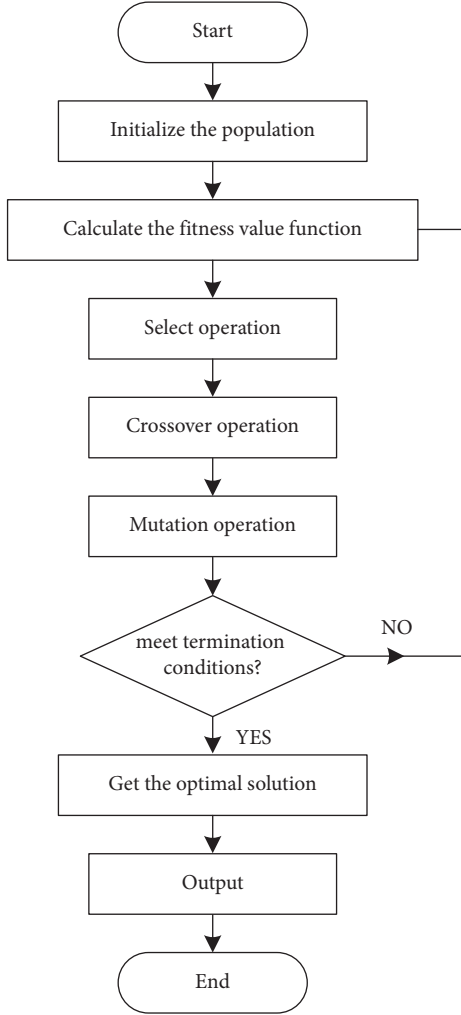


FIGURE 3: The process of genetic algorithm.

$j > 2$, and any integer N can be represented by base j , as shown in the following formula:

$$\begin{cases} N = x_k x_{k-1}, \dots, x_1 x_0 (j), \\ N = x_k j^k + \dots + x_1 j + x_0. \end{cases} \quad (18)$$

Here, j represents the base, which can represent any prime number; j represents a sequence number, which exists and is unique, so that the formula holds. For different integers, the one-dimensional Halton sequence under the corresponding base j can be obtained as shown in the following formula:

$$H_j(N) = \lambda_j(N)N = 1, 2, 3, \dots, N. \quad (19)$$

Halton sequence was introduced to randomly generate 100 points, and an initial population with less difference in initial numbers was generated. The individuals of this population are more evenly distributed on the number line, which greatly improves the quality of the initial population.

4.3. Solving Steps of Adaptive Genetic Algorithm. The following is an analysis of the problem-solving process using an

adaptive genetic algorithm, which is improved on the basis of a traditional algorithm. By giving the crossover probability and mutation probability that change dynamically with the fitness value, the problem of the local optimal solution of the algorithm is solved and the accuracy of the algorithm is improved. The specific steps of the adaptive genetic algorithm are as follows.

- (1) *Chromosome Coding.* For the proposed model, binary coding is adopted, and the coding of each chromosome is an offloading strategy X , and $X^{(h)}$ is used to represent the value of h bit in strategy offloading strategy X , and the value of $X^{(h)}$ is 0 or 1. In order to indicate that all computing tasks can be offloaded to servers, the number of coding bits H should satisfy the constraints shown in the following formula:

$$H \geq \log_2 V. \quad (20)$$

The offloading strategy of the u computing task can be represented as $[X^{(uH)}, X^{(uH+1)}, X^{(uH+2)}, \dots, X^{((u+1)H)}]$, and the number x_u of offloaded mobile edge computing server can be represented by the following formula:

$$x_u = \sum_{h=uH}^{(u+1)H} x^{(h)} \cdot 2^{(u+1)H-h}. \quad (21)$$

The computing task strategy coding of no more than 8 mobile edge computing servers is shown in Figure 4.

- (2) *Chromosome Initialization.* The main purpose is to initialize the relevant parameters of the genetic algorithm, including the maximum number of iterations, chromosome length, mutation probability, crossover probability, population size, and reserved population. In the initial population with less difference generated by introducing the Halton sequence, the encoding of the class with the highest weight in a chromosome is set as the following formula(22). The purpose is to first offload the computing tasks of the highest weight class to the mobile edge computing server to which the vehicle-mounted terminal belongs.

$$\sum_{h=uH}^{(u+1)H} x^{(h)} \cdot 2^{(u+1)H-h} = R_u. \quad (22)$$

If a total of V_c computing tasks are predefined, the codes of remaining $V - V_c$ computing tasks are randomly generated to maintain the randomness of the population.

- (3) *Fitness Evaluation.* The traditional genetic algorithm decides whether an individual inherits to the next generation according to the size of fitness value. Here, the reciprocal of user's total cost is selected as the fitness function evaluation index of the quality of the chromosome individual, and the expression is shown in the following formula:

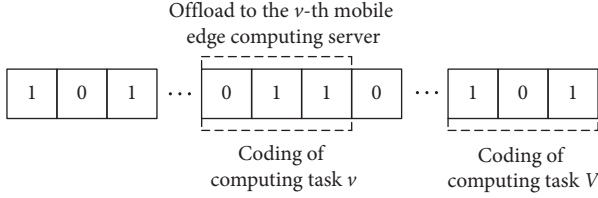


FIGURE 4: Coding of computing task strategy for mobile edge computing servers.

$$F = \frac{1}{\sum_{m=1}^M (1 - r_m) C_M^1 + r_m C_M^e} \quad (23)$$

- (4) *Select an Operation.* The roulette selection method based on fitness proportional selection is used to select some outstanding individuals from the parent individuals.
- (5) *Crossover and Mutation Operations.* Here, single-point crossover and basic bit mutation operations are performed according to the crossover probability and mutation probability, thereby generating chromosomes with new genes. Among them, the crossover probability and mutation probability of dynamically changing according to the fitness value of an individual can be expressed by the following formulas (24) and (25):

$$p_b = \begin{cases} p_{b \max} - \frac{(p_{b \max} - p_{b \min})(s_0 - s_a)}{s_{\max} - s_a}, & s \geq s_a, \\ p_{b \max}, & s < s_a. \end{cases} \quad (24)$$

$$p_c = \begin{cases} p_{c \max} - \frac{(p_{c \max} - p_{c \min})(s_0 - s_a)}{s_{\max} - s_a}, & s \geq s_a, \\ p_{c \max}, & s < s_a. \end{cases} \quad (25)$$

In formulas (24) and (25), s_{\max} represents the maximum value of fitness. s_a represents the mean value of fitness. s_0 represents a feasible solution with crossover probability. $p_{b \max}$ represents the maximum value of crossover probability. $p_{b \min}$ represents the minimum value of crossover probability. $p_{c \max}$ represents the maximum value of variation probability. $p_{c \min}$ represents the minimum value of mutation probability.

5. Experiment and Analysis

5.1. Experimental Parameter Setting. The performance of the proposed scheme is verified by simulation experiments, and the experimental environment settings and experimental results are given, which proves the superiority of the proposed scheme compared with other schemes. Firstly, vehicle nodes and mobile edge computing server nodes are randomly assigned in a simulation area of 200 m × 200 m. Then, parameters such as the number of vehicles and mobile edge computing servers,

TABLE 1: Simulation experimental parameters.

Parameter	Value
Total bandwidth	3×10^6 Hz
Number of vehicles	9
Number of mobile edge servers	3
CPU rate of mobile edge servers	3×10^{15} cycles/s
CPU rate of vehicles	5×10^{12} cycles/s
Task data size	2,6,8 MB
unit processing energy consumption	1/8.2 J/cycle
Energy consumption of vehicles to receive and send data	1.53×10^{-8} J/bit
Tolerable delay for tasks	200,300,500 ms
Received power	0.5 W
Noise power	0 W
Interference power	6 W
Population size	200
The maximum number of iterations	100
Generation gap	0.96

communication resources, and computing resources in experiment are set. The simulation experiment of the proposed strategy is carried out on the MATLAB platform (MATLAB2016a). The relevant parameters of the experiment are shown in Table 1.

5.2. Algorithm Performance Comparison Analysis. The convergence speed of the algorithm, the relationship between the average cost and amount of data, the relationship between the average delay and number of tasks, and the relationship between the total task cost and the total number of tasks are important evaluation indicators to measure the performance of the strategy. In the following, for the above-given four evaluation indicators, the resource allocation strategy based on mobile edge computing in the IoV environment proposed in this paper and the methods in reference [15], reference [16], and reference [19] are compared under the same conditions.

Firstly, the maximum number of iterations to 100 is set and the convergence rates of different algorithms are compared and analyzed. The calculation results of different algorithms are shown in Figure 5.

It can be seen from Figure 5 that the proposed strategy has the fastest convergence speed compared to the other three methods, and it tends to converge when the number of iterations is 20. This is because the Halton sequence is introduced in process of generating the initial population, and the obtained initial population is less different, which improves the speed of the algorithm to find the global optimal solution.

The performance comparison results of several algorithms are shown in Figures 6–8.

As can be seen from Figures 6–8, compared with the other three methods, the average overhead of the strategy proposed in this paper is the smallest in the case of the same task data size. And, as the amount of task data gradually increases, its average overhead grows the slowest. The average overhead of the proposed method is 0.28 when the task

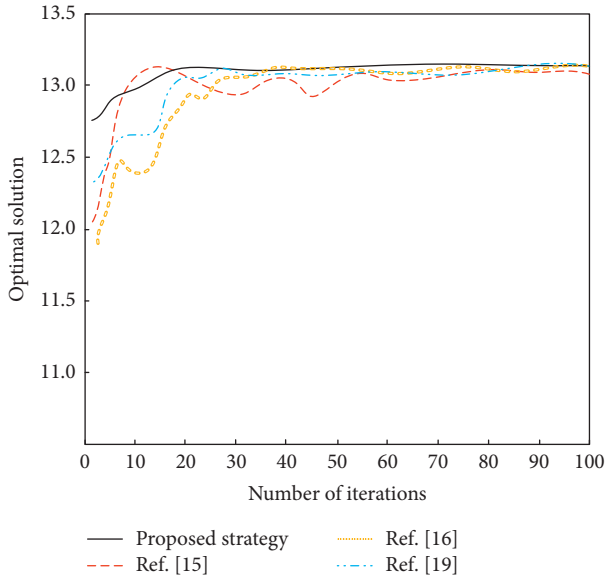


FIGURE 5: Convergence of different methods.

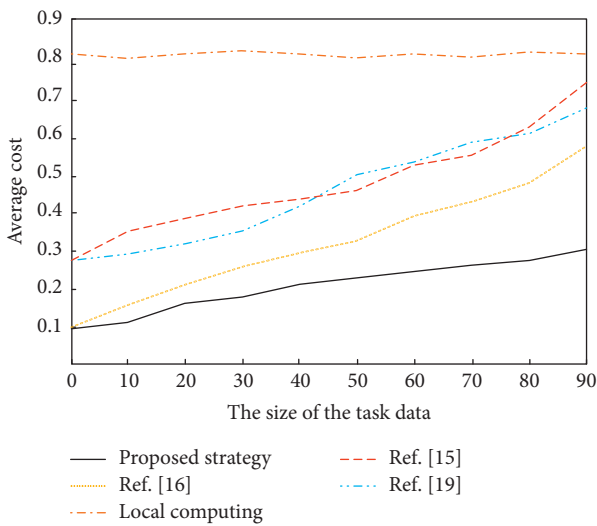


FIGURE 6: The relationship between the size of task data and the average cost of different methods.

data size is 90, which is much smaller than 0.74, 0.67, and 0.58 of the other three methods. Under the premise of the same number of tasks, the average delay and total cost of the proposed strategy are also the smallest. And, as the number of tasks increases, the growth rate of average delay and total overhead is also the smallest. When the number of tasks is 45, the average delay of the proposed method is 0.32 s, and the total cost is 0.34. The experimental results show that the introduction of the Halton sequence into a genetic algorithm can reduce the differences between individuals and strengthen the global search ability of the algorithm. Moreover, dynamically and adaptively changing the crossover probability and mutation probability can solve the problem of the local optimal solution and improve algorithm accuracy.

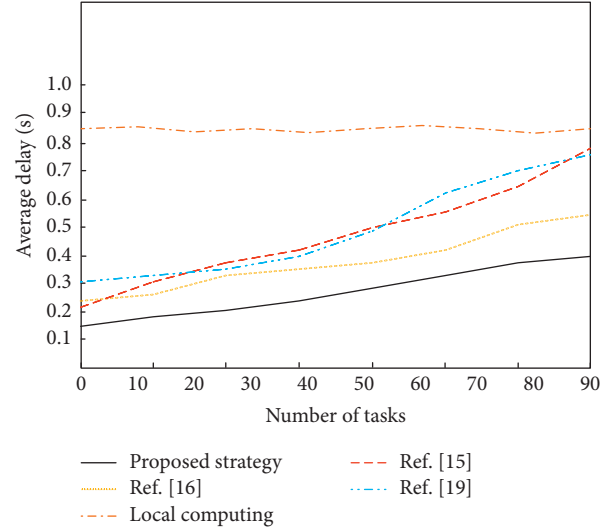


FIGURE 7: The relationship between the number of tasks and the average delay for different methods.

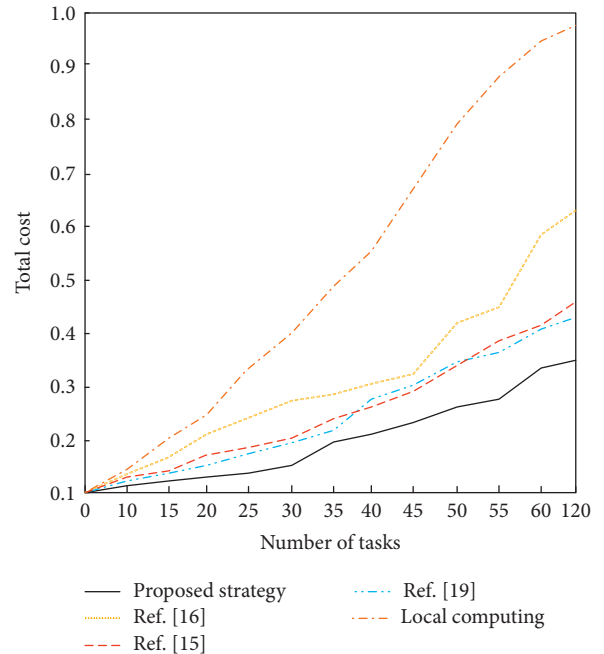


FIGURE 8: The relationship between the total number of tasks and the total cost of tasks for different methods.

6. Conclusion

Aiming at the problems of unreasonable resource allocation, large task computing delay, and energy consumption brought about by the rapid growth of IoV applications, this paper proposes a computing resource allocation strategy based on mobile edge computing in the IoV environment. The basic idea of this paper is as follows: (1) the problems that need to be solved in the current IoV environment and their quantitative calculation methods are analyzed. (2) The mobile edge computing network and resource allocation decision-making framework in the IoV environment are

designed. (3) The applicability of genetic algorithm in research problem is improved by making corresponding improvements. The simulation results show that the performance of the proposed method is better than that of the comparison method.

Future work will further study the applicability and improvement methods of the proposed algorithm in complex vehicle behaviors such as vehicle acceleration, deceleration, overtaking, and lane change. In addition, the adaptive genetic algorithm designed in this paper still has some shortcomings in computational accuracy. In future research, specific problems are analyzed to improve the population structure and further improve the quality and efficiency of the solution.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The author declares that there are no conflicts of interest regarding the publication of this paper.

References

- [1] N. Xue, H. Zhang, C. Zhang, T. Li, and D. Yuan, "Data-Driven Energy Efficient Predictive Resource Allocation in Internet of Vehicles," in *Proceedings of the 12th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 56–61, Nanjing, PEOPLES R CHINA, October 2020.
- [2] Y. Zhang, M. Zhang, C. Fan, F. Li, and B. Li, "Computing resource allocation scheme of IOV using deep reinforcement learning in edge computing environment," *EURASIP Journal on Applied Signal Processing*, vol. 2021, no. 1, pp. 12–19, 2021.
- [3] J. Huang, J. Liang, and S. Ali, "A simulation-based optimization approach for reliability-aware service composition in edge computing," *IEEE Access*, vol. 8, pp. 50355–50366, 2020.
- [4] A. Bouaziz, A. Saddoud, L. Chaari Fourati, and H. Chaouchi, "Adaptive V2X user selection and resource allocation for ultra-dense 5G HetNet network," in *Proceedings of the 17th IEEE International Wireless Communications and Mobile Computing Conference*, pp. 1454–1459, IEEE IWCMC), ELECTR NETWORK, Harbin City, China, July 2021.
- [5] S. Mao, S. He, and J. Wu, "Joint UAV position optimization and resource scheduling in space-air-ground integrated networks with mixed cloud-edge computing," *IEEE Systems Journal*, vol. 15, no. 3, pp. 3992–4002, 2021.
- [6] P. Wang, N. Xu, W. Sun, G. Wang, and Y. Zhang, "distributed incentives and digital twin for resource allocation in air-assisted internet of vehicles," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 345–349, Nanjing, Peoples R China, April 2021.
- [7] B. Liu, H. Xu, and X. Zhou, "Resource allocation in unmanned aerial vehicle (UAV)-Assisted wireless-powered internet of things," *Sensors*, vol. 19, no. 8, pp. 18–26, 2019.
- [8] S. Feki, A. Masmoudi, A. Belghith, F. Zarai, and M. S. Obaidat, "Swarm intelligence-based radio resource management for V2V-based D2D communication," *International Journal of Communication Systems*, vol. 32, no. 17, pp. 225–231, 2019.
- [9] M. Zhang, S. Wang, and Q. Gao, "A joint optimization scheme of content caching and resource allocation for internet of vehicles in mobile edge computing," *Journal Of Cloud Computing-Advances Systems and Applications*, vol. 9, no. 1, pp. 312–319, 2020.
- [10] G. Wang and F. Xu, "Regional intelligent resource allocation in mobile edge computing based vehicular network," *IEEE Access*, vol. 8, no. 12, pp. 7173–7182, 2020.
- [11] Y. Lee, S. Jeong, A. Masood, L. Park, N.-N. Dao, and S. Cho, "Trustful resource management for service allocation in fog-enabled intelligent transportation systems," *IEEE Access*, vol. 8, no. 42, pp. 147313–147322, 2020.
- [12] X. Song, Y. Tan, L. Dong, J. Wang, J. Hu, and T. Song, "Resource allocation algorithm with energy efficiency priority based on quasi-Newton interior point method in cognitive internet of vehicles," *Journal of Southeast University. Natural Science Edition*, vol. 49, no. 2, pp. 213–218, 2019.
- [13] S. Y. Wang, F. F. Liu, and H. L. Xia, "Content-based vehicle selection and resource allocation for federated learning in IoV," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 485–488, Nanjing, Peoples R China, March 2021.
- [14] H. Liang, X. Zhang, X. Hong et al., "Reinforcement learning enabled dynamic resource allocation in the internet of vehicles," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4957–4967, 2021.
- [15] Y. U. Xiang, X. Chen, Z. Wang, and X. Shi, "Resource allocation algorithm for internet of vehicles based on LTE-V2X," *Computer Engineering*, vol. 47, no. 2, pp. 188–193, 2021.
- [16] J. Cheng, G. Yuan, J. Cui, A. Zhou, L. Y. U. Bo, and L. I. Guangyao, "Video semantics-driven resource allocation algorithm in Internet of vehicles," *Journal on Communications*, vol. 42, no. 7, pp. 12–18, 2021.
- [17] J. Zhao, Z. Dong, H. Qu, and L. Lu, "Allocation algorithm for maximizing spectrum resource utilization on D2D communication in Internet of Vehicles," *Application Research of Computers*, vol. 38, no. 7, pp. 2144–2148, 2021.
- [18] Y. Huang, N. Peng, L. Yan, J. Fan, Y. Zhang, and Y. Yu, "Dynamic spectrum resource allocation in internet of vehicles based on SAC reinforcement learning," *Computer Engineering*, vol. 47, no. 9, pp. 34–43, 2021.
- [19] Y. Kang, Z. Liu, Q. Chen, and Y. Dai, "Joint task offloading and resource allocation strategy for DiffServ in vehicular cloud system," *Wireless Communications and Mobile Computing*, vol. 2020, no. 12, Article ID 8823173, 15 pages, 2020.
- [20] 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.
- [21] J. L. Zhang, M. H. Dai, and Z. Su, "Task allocation with unmanned surface vehicles in smart ocean IoT," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 443–450, 2020.
- [22] F. Jiang, W. Liu, J. Wang, and X. Liu, "Q-learning based task offloading and resource allocation scheme for internet of vehicles," in *Proceedings of the IEEE/CIC International Conference on Communications in China (ICCC)*, pp. 460–465, Electr Network, Chongqing, China, August 2020.
- [23] X. Z. Li, "A computing offloading resource allocation scheme using deep reinforcement learning in mobile edge computing systems," *Journal of Grid Computing*, vol. 19, no. 3, pp. 18–26, 2021.

- [24] T. Xiao, C. Chen, T. Qiu, C. He, Q. Pei, and H. Cao, "Joint computation resource allocation using mobile-edge-platooning-cloud in the internet of vehicles," in *Proceedings of the IEEE International Conference on Communications (ICC)*, pp. 512–515, Electr Network, Montreal, QC, Canada, June 2021.
- [25] B. Cao, Z. H. Sun, and Y. Gu, "Resource allocation in 5G IoV architecture based on SDN and fog-cloud computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 273–279, 2021.