Multi-Objective Optimized Immune Algorithm For Computing Offloading Problem In Edge Computing Scenes of Internet of Vehicles

Zhu Si-feng

Tianjin Chengjian University

Cai Jiang-hao (✉ 807079588@qq.com)

Tianjin Chengjian University <https://orcid.org/0000-0001-7123-6211>

Sun En-lin

Tianjin Chengjian University

Zhang Qing-hua

Tianjin Chengjian University

Research Article

Keywords: IoV, Mobile Edge Computing, Computing Offloading, Energy consumption, Workload balancing, offloading delay, NSGA - III, Immune algorithm

Posted Date: July 13th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-656949/v1>

License: This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Multi-objective Optimized Immune Algorithm For Computing Offloading

Problem in Edge Computing Scenes of Internet of Vehicles

ZHU Si-feng^{1a*}, CAI Jiang-hao^{1a}, SUN En-lin^{1a}, ZHANG Qing-hua^{1b}

(1a. School of Computer and Information Engineering, Tianjin Chengjian University, Tianjin, 300384, China

1b. Library, Tianjin Chengjian University, Tianjin, 300384, China)

ABSTRACT: With the development of 5G technology, the Internet of Vehicles (IoV) has also received worldwide attention. IoV edge computing achieves the goal of low latency by offloading tasks to the Mobile Edge Computing Server (MECS). However, it is still a challenge to reduce the computing delay of mobile terminal devices while ensuring the low energy consumption and load balancing of servers. In order to solve this problem, the system model, delay model, load balancing model, energy consumption model and objective optimization model are established in this paper. A computational unloading scheme based on multi-objective immune optimization algorithm is proposed. Finally, this scheme is compared with the reference scheme and the literature scheme. Simulation experiments show that the proposed scheme can effectively reduce the average unload delay of users, optimize the workload between servers, and effectively reduce energy consumption. Its performance is better than the schemes in the literature.

Keywords: IoV; Mobile Edge Computing; Computing Offloading; Energy consumption; Workload balancing; offloading delay; NSGA - III; Immune algorithm

1.Introduction

In recent years, with the rapid development of the economy, the number of vehicles has also exploded, and traffic problems often appear. With the emergence of 5G technology, terminal devices, especially vehicles, will have more urgent demands on resources, and edge computing of the Internet of Vehicles has gradually become a research hotspot [1]. Internet of vehicles means that vehicles perceive the status information inside the vehicle through sensing technology, and obtain all dynamic information of vehicles within the regional platform through the infinite communication network, and make effective use of it. The existing Internet of vehicles reduces the probability of vehicle collisions by providing vehicles with space between vehicles. Or through real-time navigation to improve the user's driving experience while improving the efficiency of traffic operation. In addition, the number of on-board multimedia devices is also increasing, the addition of a variety of entertainment facilities also makes the

vehicle needs a large increase in computing resources, which makes the vehicle to unload the task of a large increase in demand. Cloud computing allows mobile terminals will be sent to the cloud, to effectively reduce the load and energy consumption of mobile terminals, but the car networking tasks are very strict for the time delay, high delay brought by the upload task to the cloud is car networking scenario is difficult to accept, the edge, it can be added to cloud computing, as a small cloud computing network, Edge computing devices can better meet the needs of Internet of Vehicles [2]. The main difference between edge computing and cloud computing is that edge computing transfers the task to the edge server nearby, which can greatly reduce the delay of the task and reduce the load on the cloud terminal, and at the same time has better security. In the unloading process, the unbalanced assignment of tasks may lead to excessive load of some servers. When the load of MEC server is unstable, it will affect the server performance and even lead to server failure. On the other hand, the energy consumed in the operation process is also one of the measurement criteria [3]. Therefore, when allocating resources, it is necessary to optimize the load and energy consumption of a large number of deployed MEC servers to reduce the cost of edge equipment [4].

Therefore, the design of a computing mode that can guarantee load balancing and low energy consumption on the premise of low delay is a problem worth studying. At present, the unloading decision problem in the IoV edge computing scenario has attracted extensive attention from scholars at home and abroad. Literature [3] designed a vehicle-to-vehicle unloading model for cross-regional task transmission, and used NSGA-III to optimize the load balancing rate, server energy consumption and computational task processing delay. Literature [4] proposes a blockchain-based computing offloading method to ensure data integrity, and adopts NSGA-III to generate resource allocation strategy, in view of the phenomenon of unbalanced server load and insecure information transmission. Literature [5] designed a three-layers architecture, including the vehicles cloud, roadside cloud and cloud, effective for a large number of tasks on the road to the request for distribution, reasonable use of the resources of three between clouds, and combining genetic algorithm and particle swarm algorithm, designs an adaptive particle swarm optimization algorithm, effectively reduce the task response time delay and energy consumption. In literature [6], the trade-off relationship between energy consumption and task processing delay was studied in the fog cloud scenario, and the NSGA-II algorithm was used to conduct experiments on it. Literature [7] solves the problem of workload distribution in the Internet of Things - fog - cloud architecture, and makes a tradeoff between energy consumption and propagation delay. NSGA-II algorithm is used to process the multi-objective model, and simulation calculation is carried out in three

scenarios of fog, pure cloud and fog cloud, which proves that this scheme has significant differences. Literature [8] adopts a hybrid task scheduling algorithm, which has the maximum completion time of farmland and better performance than the single ant colony algorithm and genetic algorithm in a multi-processor environment. Literature [9] studied resource allocation in a multi-task environment. In order to make effective use of resources, a multi-task evolution scheme of dynamic resource allocation was designed, which could be adaptively allocated to each task according to the demands of the task and had the ability of cross-regional allocation. Literature [10] introduces an adaptive unloading method based on genetic algorithm to deal with effective traffic in the Internet of Things, which effectively reduces unnecessary delay and complexity in the request process and improves the success rate of Internet of Things requests. In view of the lack of set interpretation of the weighting coefficient, Literature [11] proposed a MMD method with rich geometric interpretation, which could generate the weighting coefficient with a systematic method without prior preference. Literature [12] designed a new architecture to achieve efficient data scheduling in mixed V2I and V2V communication, proposed the problem of time data uploading and publishing, and proposed an evolutionary multi-objective algorithm, MO-TDUD, to improve data quality and delivery rate. Literature [13] has designed a vehicular delay-tolerant networks (VDTNs) to transfer data to vehicles on the edge of the network to process and switch the large data set generated by the vehicles. Literature [14] proposed a resource allocation method named MRAM in the Internet of Things scenario, and proposed multi-criteria decision making and order preference technology based on ideal solution similarity, which effectively solved the requirements of low delay, low load balancing and low energy consumption. Literature [15] aims at shortening communication and processing delay, and adopts distributed alpha algorithm and optimal signal-to-noise ratio to distribute Internet of Things traffic to appropriate workstations, so as to achieve traffic load balance between workstations and fog nodes. Literature [16] proposed a general framework based on the application of the Internet of Things, in which tasks were assigned by setting a threshold. If the response time was less than the threshold, it would be accepted by the fog node; otherwise, it would be transmitted to other fog nodes or clouds. Literature [17] proposed an optimization method based on stochastic gradient descending (SGD), designed an efficient distributed unloading model to solve the nonlinear multi-objective optimization problem, and found the optimal balance between energy consumption delay and cost.

Our main contributions are following:

- Propose a multi-objective immune algorithm(MOIA) to realize multi-objective optimization of unloading delay, server energy consumption and load balancing.

- Design a vehicle-to-vehicle optimization model to solve the task unloading target problem when the vehicle moved across the region, and the server energy consumption and load balancing were optimized under the premise of ensuring low delay.
- Conduct extensive experimental evaluations to prove the effectiveness of the proposed algorithm and model

The remaining of this paper is organized as follows. Section 2 establishes the system model, mathematical model and problem model. Section 3 describes the design and improvement of the algorithm. Section 4 designs related experiments. Section 5 summarizes the article and looks into the future.

2 Offloading decision model in MEC scenarios of IoV

2.1 System model

The system model designed in this paper is shown in Figure 1 without loss of generality. Suppose there are M cars in the system, denoted as $\text{Vehicles} = \{v_1, v_2, \dots, v_M\}$, and the vehicle v_i have a $task_i$ that needs to be unloaded from the server. The task needs storage resources and computing resources, which are jointly expressed as $\langle z_i, c_i \rangle, 1 \leq i \leq M$. A sufficient number of base stations are available for unloading, and vehicles will submit tasks to the base stations according to where they are located. Suppose there are N edge computing servers in the system, denoted as $\text{MECS} = \{MEC_1, MEC_2, \dots, MEC_N\}$, and each edge computing server has independent storage resources and computing resources, which are jointly expressed as $\langle Z_j, C_j \rangle, 1 \leq j \leq N$. Each server has V virtual machine instances, denoted as $\text{VMS} = \{vm_1, vm_2, \dots, vm_v\}$, and can handle multiple tasks at the same time. The server allocates computing resources according to the number of virtual machine instances.

Fig. 1 System model

In order to simplify the analysis process and obtain valuable rules, this paper assumes that the process is quasi-static, that is, the task set in a single time period is unchanged, but the task set will change in different time periods. When at the beginning of each time period, the vehicles will be submit to the server of the area needed for the task of computing resources and storage resources, represented as binary group $\langle z_i, c_i \rangle$, after the server upload resources to the cloud, the cloud will target server according to the current server status for task allocation, task when a vehicle unloading area target server is not in the vehicle, The vehicle is transmitted based on vehicle to vehicle (V2V) technology, and the task is first passed to the vehicle in the next area, which is then forwarded to the server or the vehicle in the next area. Similarly, when the vehicle is crossing the region, if the vehicle has already left the region, the unloading target information will be sent by the original server to the nearest vehicle in the boundary area, and transmitted by the vehicle. Ensure that the server has enough storage space to accept tasks during task assignment. Set as the set of unfinished tasks that have been assigned to the above, then Equation (1) shall be satisfied during the assignment process.

$$z_i + \sum_{k \in K_j} z_k^j < Z_{MEC}^j \quad (1)$$

2.2 Workload balancing model

In order to better evaluate the load degree of this model, the usage degree of all MEC servers and the load task amount assigned to a single server are combined to discuss. The usage rate u_j of VM instances on MEC_j is shown in Equation (2).

$$u_j = \frac{1}{V} \sum_{i \in M} \text{tag}_j^i \quad (2)$$

Where V represents the number of virtual machine instances on the server, and tag_j^i is the binary variable indicating whether task_i appears on MEC_j . Let the vector group $\text{ATN} = \{\text{atn}_1, \text{atn}_2, \dots, \text{atn}_N\}, 1 \leq j \leq N$ represent the usage of the server, where atn_j is a binary variable, as shown in Equation (3).

$$\text{atn}_j = \begin{cases} 1, & \text{if } \text{MEC}_j \text{ is occupied} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Then, the average utilization of virtual machine instances on all servers is represented by MSU as shown in Equation (4).

$$\text{MSU} = \frac{\sum_{j=1}^N u_j}{\sum \text{ATN}} \quad (4)$$

Then, the resource balance rate MSB of all servers is shown in Equation (5).

$$\text{MSB} = \frac{\sum_{j=1}^N |u_j - \text{MSU}|}{\sum \text{ATN}} \quad (5)$$

The load balancing rate L of the system can be obtained as shown in Equation (6).

$$L = \text{MSB} / \text{MSU} \quad (6)$$

2.3 Time consumption model

Unloading delay is the most important criterion in this model, and the size of unloading delay will directly affect the user experience. In this model, unloading delay is mainly composed of four parts: 1. Upload delay, 2. Processing delay, 3. Feedback delay, 4. V2V propagation delay. Where, the upload delay of task_i can be expressed as Equation (7).

$$T_{\text{offl}}^i = c_i / \theta_{V2E} \quad (7)$$

Where θ_{V2E} is the byte transfer rate from the vehicle to ECD (V2E). The processing delay of a task is directly related to the number of virtual machine instances on the server and the computing resources required for the task. The processing delay of task_i is expressed in Equation (8).

$$T_{\text{exec}}^i = z_i / \frac{Z_j}{V * u_j} \quad (8)$$

After the task processing is completed, it needs to be sent back to the vehicle, so the feedback time of task_i is expressed in Equation (9).

$$T_{\text{feed}}^i = c_i' / \theta_{V2E} \quad (9)$$

Then, the unloading delay $\tau^{(1)}$ of task_i is expressed in Equation (10).

$$\tau^{(1)} = T_{offl}^i + T_{exec}^i + T_{feed}^i \quad (10)$$

When the vehicle drives along the road, it may pass through two or more server areas, then the execution server of the task is judged to be the focus of the model. If the vehicle in the handover area needs to unload the task to the next regional server, the changed task will be transmitted by the vehicle in the next region. Vector group $H = \{h_1, h_2, \dots, h_m\}, 1 \leq k \leq M$ is used to indicate whether the task will be unloaded to other servers, where h_i is shown in Equation (11).

$$h_i = \begin{cases} 1, & \text{The task}_i \text{ is executed on the original server} \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

Then, the time-delay T_{V2V}^i of the cross-region transmission of $task_i$ is expressed in Equation (12).

$$T_{V2V}^i = \frac{c_i}{\theta_{V2V}} (1 - h_i) \rho \quad (12)$$

Where, ρ is the number of vehicle-to-vehicle transmission, then the unloading delay $\tau^{(2)}$ of $task_i$ is shown in Equation (13).

$$\tau^{(2)} = T_{offl}^i + T_{exec}^i + T_{feed}^i + T_{V2V}^i \quad (13)$$

Then, the total unloading delay τ can be expressed as Equation (14).

$$\tau = \sum_{i \in M} h_i * \tau^{(1)} + (1 - h_i) * \tau^{(2)} \quad (14)$$

2.3 Energy consumption model

The server energy consumption is divided into three parts: server base energy consumption, virtual machine instance energy consumption and unused virtual machine instance energy consumption. The power consumption of the server mainly depends on the power of the server and the virtual machine and the processing time of the task. Is expressed as Equation (15).

$$T_{serv}^j = \text{MAX}(T_{exec}^i), i \in K_j \quad (15)$$

Then the basic energy consumption E_{base} of all servers is expressed as Equation (16).

$$E_{base} = \sum_{j \in N} atn_j * T_{serv}^j * \alpha \quad (16)$$

Where α is the power of the server. The energy consumption of virtual machine instances in use, S , is shown in Equation (17).

$$E_{used} = \sum_{i \in M} T_{exec}^i * \beta \quad (17)$$

Where β is the power of the virtual machine in use, and E_{idle} is the energy consumption of the idle virtual machine, as shown in Equation(18).

$$E_{idle} = (\sum_{j \in N} W * T_{serv}^j - \sum_{i \in M} T_{exec}^i) * \gamma \quad (18)$$

Where γ is the idle virtual power, then the total energy consumption of all servers E is shown in Equation (19).

$$E = E_{base} + E_{used} + E_{idle} \quad (19)$$

2.4 Optimization model

The model designed in the Internet of Vehicles scenario in this paper takes minimizing the delay as the main goal, and ensures the load balance of each server while reducing the total energy consumption of the server. Therefore, the problem to be solved in this paper is defined as follows

$$\min \tau \quad L \quad E \quad (20)$$

$$st. \sum_{k \in K_j} z_k^j \leq Z_{MEC}^j, 1 \leq j \leq N \quad (21)$$

$$\sum_{i \in M} tag_j^i \leq W, 1 \leq j \leq N \quad (22)$$

Equations (21) and (22) indicate that the server shall not be overloaded during task assignment, and the number of allocated tasks shall not cause the number of idle VM Instances of the server.

3. Offloading decision scheme based on multi-objective immune algorithm

In this paper, a computational unloading scheme under the scenario of the Internet of Vehicles is proposed. The computational unloading problem is a multi-objective optimization problem, which needs to ensure the minimum vehicle loading time, while ensuring the low energy consumption of the server and the load balance of each server. NSGA-III is an excellent solution to multi-objective optimization problems. Based on NSGA-III, a multi-objective optimization immune algorithm (MOIA) is proposed in this paper. First, the vehicle and the edge server were coded, then the fitness function and constraint conditions were proposed based on the multi-objective optimization problem, and the immune operation was used to search. Finally, the antibody group was updated by using the dominant principle and reference point-based selection. Compared with the conventional NSGA-III, the proposed algorithm has better multi-peak searching ability.

Non-Dominant Sorting Genetic Algorithm (NSGA-III) is a population-based algorithm based on the theory of individual natural evolution. An individual represents a point in the search space, and non-dominant sorting techniques and genetic operators (selection, crossover, and mutation) are used to obtain the best solution and create a new population. Immune algorithm is an intelligent search algorithm inspired by biological immune system. It is a heuristic random search algorithm that combines certainty and random selection. In the immune algorithm, the antigen represents the problem to be optimized, each antibody represents a feasible solution, and each gene represents a computational

unloading strategy for a task. In this paper, we will adopt decimal encoding and unload decision will be made using vector group $X = (x_1, x_2, \dots, x_m), 1 \leq x_i \leq N$, where each gene bit represents the unload target server of a task. In the immune algorithm, the affinity function is used to evaluate the adaptability of antibodies to antigens, that is, to evaluate the pros and cons of antibody solutions. The affinity function in this paper consists of three parts: time delay, energy consumption and load balance, as shown in Equations (6), (14) and (19). Constraints (21) and (22) are also used to make the server run normally.

Algorithm 1 gives the pseudocode of NSGA-III. The first step is to generate an initial population and select the activation edge devices for each solution. For each solution of the population, the unload server is selected for all tasks, the solution is compared with other solutions in the new population to ensure the uniqueness of the solution, and then random selection is made on the basis of the current population to simulate two-point crossover and polynomial mutation to generate the offspring population. Fusion of parent and child populations. In order to select the next generation population, the non-dominant order based on Pareto dominance was used to divide it into several different non-dominant layers, and a new population was constructed by adding the solutions of each non-dominant layer to the new population to the maximum size successively. In the proposed NSGA-III, the maximum number of iterations is taken as the termination criterion to determine when the algorithm stops, The computational complexity of the NSGA-III is $O(Mn^2)$, where M is the number of objective functions and n is the population size [18].

Algorithm 2 gives the pseudocode of MOIA. In the model design, the distance allocated to the target will directly lead to the increase of the delay, so the first step is to randomly generate the initial population according to the initial state of the allocation. Meanwhile, functions Eq.(21) and Eq.(22) are used to constrain the population. After completing the constraint, the objective functions Eq.(6), Eq.(14) and Eq.(19) are used to calculate its fitness value. In order to ensure the diversity of generated solutions, the same reference-centered scheme as NSGA-III was adopted in MOIA. After the standardization and association of the current population, the former NP/2 antibodies were selected according to the non-dominant layer for immunization operation. The computational complexity of the MOIA is $O((n_d + n_c)^2)$, where n_d is the maximum size of the dominant population and n_c is the clone population size [19].

Algorithm 3 produces the pseudocode of the immune operation. When the selected antibodies are

cloned and mutated, the roulette strategy is used to calculate the mutation of the individual, which makes the task on the server with lower load have higher unloading probability. After the completion of the mutation operation, excellent individuals should be selected from the mutation population. In the improved MOIA, the role played by the reference line is deepened. The relevant parameters will be retained after the non-dominant ordering of the individual. In the screening of the immune population, the individual will be retained if one of the following two conditions is met: 1. This individual dominates the original individual 2. This individual uses a new guide in the set of guides used by the original population. After the completion of the iteration, the size of the new population will be greater than $NP/2$. In this case, the population will be completed to NP through the population refresh operation.

Algorithm 1: Pseudo Code for NSGA-III

Input: H structured reference points Z^s or supplied aspiration point Z^a , parent population P_t

Output: P_{t+1}

begin

```

1   $S_t = \phi, i = 1$ 
2   $Q_t = \text{Recombination} + \text{Mutation}(P_t)$ 
3   $R_t = P_t \cup Q_t$ 
4   $(F_1, F_2, \dots) = \text{Non-dominated-sort}(R_t)$ 
5  Repeat
6     $S_t = S_t \cup F_i$  and  $i = i + 1$ 
7  Until  $|S_t| \geq N$ 
8  Last front to be included:  $F_l = F_i$ 
9  If  $|S_t| \geq N$  then
10    $P_{t+1} = S_t$ , break
11 else
12    $P_{t+1} = \bigcup_{j=1}^{l-1} F_j$ 
13   Points to be chosen from  $F_l$ :  $K = N - |P_{t+1}|$ 
14   Normalize objectives and create reference set  $Z^r$ :  $\text{normalize}(f^n, S_t, Z^r, Z^s, Z^a)$ 
15   Associate each member  $s$  of  $S_t$  with a reference point:  $[\pi(s), d(s)] > \text{Associate}(S_t, Z^r)$ 
16   %  $\pi(s)$ : closest reference point,  $d$ : distance between  $s$  and  $\pi(s)$ 
17   Compute niche count of reference point  $j \in Z^r$ :  $\rho_j = \sum_{s \in S_t / F_l} ((\pi(s) = j) ? 1 : 0)$ 
18   Choose  $K$  members one at a time from  $F_l$  to construct  $P_{t+1}$ :  $\text{Niching}(K, \rho_j, \pi, d, Z^r, F_l, P_{t+1})$ 
19 End if
end begin

```

Algorithm 2: Pseudo Code for MOIA

Input: $|M|, |N|, NP, g \max, p m, z_j, c_j, Z_j, C_j, V, \theta_{V2E}, \theta_{V2V}, \alpha, \beta, \gamma$

Output: The Optimal solution set $L \quad E$

begin

```
1 AssignTasksToServers(M, N) /* assign tasks to target servers */
2  $P_{(0)} \leftarrow \text{Initialize Population}()$  /* initialize population with initial unloading state and NP*/
3 CheckConstraints()/* check constraints given by formulations (Eq. (21), Eq. (22)).*/
4 EvaluateFunctions(Eq. (6), Eq. (14), Eq. (19))/* evaluate the objective functions */
5  $P(1) \leftarrow \text{NondominatedSort}(P(0))$  /* perform a non-dominant sort on the population  $P(0)$  */
6 for  $i=1$  to  $gmax$  do
7    $Q_{(i)} \leftarrow \text{SelectOperation}(P(i), NP/2)$ 
8   /* select the former NP/2 individuals according to nondominated layer/
9    $T_{(i)} \leftarrow \text{ImmuneOperation}(Q_{(i)})$ /* solution is evaluated through the Cloning,mutation and
10  clonal inhibition operation and another population is generated */
11   $P(i+1) \leftarrow \text{RefreshThePopulation}(T_{(i)})$ /* complete the population size to NP */
12  CheckConstraints()/* check constraints given by formulations (Eq. (21), Eq. (22)).*/
13  EvaluateFunctions (Eq. (6), Eq. (14), Eq. (19))/* evaluate the objective functions */
14   $R_{(i)} \leftarrow \text{NondominatedSort}(P(i+1))$ 
15  /* the population is sorted in fronts and  $R_{(i)}$  is Reference line set */
16 end for
end begin
```

Algorithm 3: Pseudo Code for immune operation

Input: $Q_{(i)}, NP, R_{(i)}, Ncl, g, g \max, p m, z_j, c_j, Z_j, C_j, V, \theta_{V2E}, \theta_{V2V}, \alpha, \beta, \gamma$

Output: $T_{(i)}$

begin

```
1 for  $np=1$  to  $NP$  do
2    $L(np) \leftarrow \text{CloneOperation}(\text{clone}(Q(i)(np)), Ncl)$ /* Generate a clone population of size Ncl */
3   MutationOperation( $L(np)$ )/* Perform mutation operation according to roulette */
4   CheckConstraints()/* check constraints given by formulations (Eq. (21), Eq. (22)).*/
5   EvaluateFunctions(Eq. (6), Eq. (14), Eq. (19))/* evaluate the objective functions */
6   for  $NC=1$  to  $Ncl$  do
7     1.replace:  $L(np)(nc)$  dominate original individual
8     2.save:  $L(np)(nc)$  use a new reference line
9     3.remove:otherwise
10    /* It is saved only if individual dominates the original individual or
11    if the individual uses a new reference line */
12  end for
13 end for
14  $T_{(i)} = [T_{(i)}, L(np)]$ /* generation of new populations */
end begin
```

4. Performance evaluation

In order to verify the effectiveness of the proposed MOIA-based scheme, it can effectively reduce the unloading delay of mobile users, ensure the load balance of the server and effectively reduce the energy consumption of the server. This paper uses MATLAB to carry on the simulation experiment. In this simulation experiment, the effectiveness of MOIA is first verified, and it is compared with the traditional NSGA-III. In order to verify the effectiveness of the scheme, the scheme using MOIA proposed in this paper is compared with other schemes. The following is a description of the unloading scheme adopted:

- **Benchmark:** All tasks are offloaded to the nearest server. However, if it is discovered during the unloading process that the task requires more resources than the server being unloaded has. It will unload to the nearest server and repeat the process until all tasks have been assigned.
- **Computing offload method based on cloud edge computing (MOC):** In literature [3], a solution algorithm based on SAW and MCDM is designed to reduce time and energy consumption and maintain system stability by managing resources in IOV-CEC system. NSGA-III is used to realize the multi-objective optimization.

In this paper, load balancing, time cost and energy consumption are taken as the main evaluation indexes. The comparison between MOIA Benchmark and MOC in the same experimental background is analyzed in detail. Detailed simulation parameters developed in this experiment are shown in Table 1.

Table1 Parameter settings

Parameter description	Value
The total number of ECDs N	20
Task requests computing resources C_i	evenly distributed between 1 and 2/(MIPS)
Task requests the storage resource Z_i	evenly distributed between 10 and 20/ (GB)
MEC server computing resources C_{MEC}	evenly distributed between 100 and 200/(MIPS)
The MEC server stores resources Z_{MEC}	evenly distributed between 1000 and 2000/ (GB)
The data transmission rate based on V2V technology θ_{V2V}	1Gb/s
The data offloading rate based on V2I technology θ_{V2E}	600Mb/s
The power rate of the servers in the ECDs α	300W
The power rate of the employed VM instances β	50W
The power rate of the unemployed VM instances γ	30W

3.1 Evaluation of algorithm effectiveness

In this section, in order to prove the effectiveness of the proposed algorithm, the MOIA of improved NSGA-III proposed in this paper is compared with the traditional NSGA-III. The Pareto frontier was obtained by running the traditional NSGA-III 100 times. A group of representative experimental results will be presented in this paper. Figures 4, 5, and 6 illustrate the non-dominated solution. Fig. 4 illustrates the relationship between task unloading delay and energy consumption. In all schemes, the relationship between time delay and energy consumption is roughly inversely proportional, that is, the energy consumption will show an upward trend with the reduction of time delay. Figure 5 illustrates the relationship between load balancing and energy consumption. As the load balancing decreases, the energy consumption also decreases. Figure 6 illustrates the relationship between task unload delay and load balancing. With the optimization of load balancing, the time delay will also show a decreasing trend. Compared with the NSGA-III, the MOIA designed in this paper is considered to have better performance in multi-peak search than the traditional NSGA-III according to the conclusions in Figure 4, Figure 5 and Figure 6.

Fig.2. Pareto front of Energy consumption vs. Time consumption.

Fig. 3. Pareto front of Time consumption vs. Workload balancing.

Fig. 4. Pareto front of Energy consumption vs. Workload balancing.

3.2 Evaluation of model effectiveness

3.2.1 comparison of load balancing on the server

Load balancing is an important index of the server. This paper estimates the load balancing of the system by discussing the load of a single server and all the servers. Figure 5 shows the load of MOIA Benchmark and MOC, which distribute tasks more evenly across servers than Benchmark and MOIA.

With the increasing number of vehicles, the load balancing performance of the scheme using MOIA is improved compared with that of Benchmark scheme and MOC scheme.

Fig. 5 Comparison of the workload balancing of ECDs of Benchmark, MOIA and MOC with different vehicle scales

3.2.2 comparison of server power consumption

Energy consumption consists of three parts: server base energy consumption, virtual machine instance energy consumption in use and idle virtual machine instance energy consumption. The energy consumption of the server in these three aspects is calculated in Figure 6a, 6b and 6c. In Fig. 6a, it can be seen that under the reasonable allocation of the algorithm, the total running time of each server is reduced and the basic energy consumption is also reduced. In Figure 6b, as the MOIA reduces the processing delay of tasks with more transmission delay, it can be seen that the energy consumption of the virtual machine used by the server to process tasks decreases. In Fig. 6c, when the number of tasks is small, the load of each server is small, and the effect of MOIA is not obvious. When the number of tasks is increased, the scheme using MOIA tends to have obvious advantages over other schemes. Figure 7 shows the total energy consumption of each scheme in the case of different number of vehicles. It can be seen from Figure 7 that this scheme can effectively reduce energy consumption compared with other schemes.

(a) The baseline energy consumption of ECDs

(b) The energy consumption of employed resource units

(c) The energy consumption of unemployed resource units

Fig. 6 Comparison of different components of energy consumption by Benchmark, MOIA and MOC with different vehicle scales

Fig. 7 Comparison of energy consumption of Benchmark, MOIA and MOC with different

3.2.3 Comparison of task unload delays

The unloading delay of the task is the most important criterion. This paper divides the total unloading delay into four parts: upload delay, execution delay, return delay and V2V transmission delay, as shown in Fig. 8a,8b,8c and 8d. The scheme proposed in this paper introduces vehicle-to-vehicle transmission technology and attempts to reduce the processing delay of the server through vehicle-to-vehicle transmission, while optimizing load balancing and low energy consumption. Therefore, it may cost more transmission delay to transfer the task from the high-load server to a suitable MEC server to perform. Figure 8a shows the processing time of the task with different number of vehicles, and Figure 8d shows the transmission time of the task with different number of vehicles. It can be seen that compared with other schemes, the scheme using MOIA has a higher transmission delay, but has a certain reduction in processing delay. Fig. 9 shows the total unloading delay of each scheme. From Fig. 9, we can see that the scheme using MOIA has a slight increase in the total delay compared with other algorithms, but the comparison of comprehensive energy consumption and load balancing shows that it has significant effects on the optimization of load balancing and server energy consumption

(a) Execution Time

(b) Feedback Time

(c) Offloading Time

(d) Transmission Time

Fig. 8 Comparison of the different part of time consumption of Benchmark, MOIA and MOC with different vehicle scales

Fig. 9 Comparison of the time consumption by Benchmark, MOIA and MOC with different vehicle scales

4. Conclusion

In this paper, the parallel unloading problem of multiple vehicles is discussed in the scenario of the Internet of Vehicles, and a MEC system model is designed. Based on this model, the goal is to minimize the mobile user delay and reduce the total energy consumption of the server as much as possible while maintaining the load balance of the server. Firstly, a multi-objective optimization function is constructed to calculate the unloading method, and the idea of immune algorithm is used to improve the traditional NSGA-III, and a lot of simulation experiments are carried out. The simulation results show that the proposed MOIA has better multi-peak optimization ability than the traditional NSGA-III algorithm, and the designed unloading scheme can effectively reduce the unloading delay, reasonably allocate tasks to maintain the load balance of each server and effectively reduce the energy consumption of the server. In addition, compared with the benchmark scheme and the unload scheme in literature [3], the unload scheme based on MOIA proposed in this paper can carry out more efficient unload of tasks with better performance and has better application value.

In future research work, the proposed MOIA will be improved through reinforcement learning and applied to more complex models and scenarios.

Declaration

Acknowledgement

This work was supported by the National Natural Science Foundation of China No.61972456. Research and Innovation Project for Postgraduates in Tianjin (Artificial Intelligence) No.2020YJSZXS25.

Data Availability Statements

The datasets generated during and analysed during the current study are available from the corresponding author on reasonable request.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] Shaik Mazhar Hussain, Kamaludin Mohamad Yusof. Dynamic Q-learning and Fuzzy CNN Based Vertical Handover Decision for Integration of DSRC, mmWave 5G and LTE in Internet of Vehicles (IoV)[J]. Journal of Communications,2021,16(5).
- [2] Huang Mengxing, Zhai Qianhao, Chen Yinjie, Feng Siling, Shu Feng. Multi-Objective Whale Optimization Algorithm for Computation Offloading Optimization in Mobile Edge Computing[J]. Sensors (Basel, Switzerland),2021,21(8).
- [3] Xiaolong Xu,Renhao Gu,Fei Dai,Lianyong Qi,Shaohua Wan. Multi-objective computation offloading for Internet of Vehicles in cloud-edge computing[J]. Wireless Networks,2020, 26(5):1611–1629.
- [4] Xu, X., Zhang, X., Gao, H., Xue, Y., Qi, L., & Dou, W. (2019). Become: Blockchain-enabled computation offloading for IoT in mobile edge computing. IEEE Transactions on Industrial Informatics, 16, 4187–4195.
- [5] Sadip Midya,Asmita Roy,Koushik Majumder,Santanu Phadikar. Multi-objective optimization technique for resource allocation and task scheduling in vehicular cloud architecture: A hybrid adaptive nature inspired approach[J]. Journal of Network and Computer Applications,2018,103.
- [6] S. Vimal,Manju Khari,Rubén González Crespo,L. Kalaivani,Nilanjan Dey,M. Kaliappan. Energy enhancement using Multiobjective Ant colony optimization with Double Q learning algorithm for IoT based cognitive radio networks[J]. Computer Communications,2020,154.
- [7] Mahdi Abbasi,Ehsan Mohammadi Pasand,Mohammad R. Khosravi. Workload Allocation in IoT-Fog-Cloud Architecture Using a Multi-Objective Genetic Algorithm[J]. Journal of Grid Computing,2020(prepublish).
- [8] Sayantani Basu,Marimuthu Karuppiiah,K. Selvakumar,Kuan-Ching Li,S.K. Hafizul Islam,Mohammad Mehedi Hassan,Md. Zakirul Alam Bhuiyan. An intelligent/cognitive model of task scheduling for IoT applications in cloud computing environment[J]. Future Generation Computer Systems,2018,88.
- [9] Computation - Evolutionary Computation; Recent Studies from Xidian University Add New Data to Evolutionary Computation (Evolutionary Multitasking With Dynamic Resource Allocating Strategy)[J]. Journal of Robotics & Machine Learning,2019.

- [10] Azham Hussain, S. V. Manikanthan, T. Padmapriya, Mahendran Nagalingam. Genetic algorithm based adaptive offloading for improving IoT device communication efficiency[J]. *Wireless Networks*, 2019 (prepublish).
- [11] Computation - Evolutionary Computation; Reports Outline Evolutionary Computation Study Findings from Yuan Ze University (Minimum Manhattan Distance Approach to Multiple Criteria Decision Making in Multiobjective Optimization Problems)[J]. *Journal of Robotics & Machine Learning*, 2017.
- [12] Machine Learning; Studies in the Area of Machine Learning Reported from Southwest Jiaotong University (Temporal Information Services in Large-Scale Vehicular Networks Through Evolutionary Multi-Objective Optimization)[J]. *Journal of Robotics & Machine Learning*, 2019.
- [13] Kumar, N., Zeadally, S., & Rodrigues, J. J. (2016). Vehicular delay-tolerant networks for smart grid data management using mobile edge computing. *IEEE Communications Magazine*, 54(10), 60.
- [14] Qi Liu, Ruichao Mo, Xiaolong Xu, Xu Ma. Multi-objective resource allocation in mobile edge computing using PAES for Internet of Things[J]. *Wireless Networks*, 2020 (prepublish).
- [15] Fan, Q., Ansari, N.: Towards workload balancing in fog computing empowered IoT. *IEEE Transactions on Network Science and Engineering*. (2018)
- [16] Yousefpour, A., Ishigaki, G., Jue, J.P.: Fog computing: Towards minimizing delay in the internet of things. In: 2017 IEEE International Conference On Edge Computing (EDGE), pp. 17–24 (2017)
- [17] Engineering; Researchers from South Asian University Report New Studies and Findings in the Area of Engineering (Computation Offloading for Distributed Mobile Edge Computing Network: a Multiobjective Approach)[J]. *Journal of Engineering*, 2020.
- [18] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
- [19] Maoguo Gong, Licheng Jiao, Haifeng Du, Liefeng Bo. Multiobjective Immune Algorithm with Nondominated Neighbor-Based Selection[J]. *Evolutionary Computation*, 2008, 16(2).

Author biographies

ZHU Si-feng earned his Ph.D. degree from school of computer science, Xidian university, Xi'an China, in 2012. Now, he is a professor in Tianjin Chengjian University, Tianjin, China. His research interests include mobile edge computing, internet of things and multi-objective optimization algorithm. He has published more than 20 papers in some international journals. E-mail: zhusifeng@tcu.edu.cn.

CAI Jiang-hao received his B.S. degree from Zhonghuan Information College Tianjin University of Technology, in 2020. Currently, he is pursuing the M.S. degree in software engineering at School of Computer Science and technology, Tianjin Chengjian University, Tianjin, China. His research interests include mobile edge computing and multi-objective optimization algorithm. E-mail: 807079588@qq.com.

SUN En-lin received his B.S. degree from Xingtai University, in 2019. Currently, he is pursuing the M.S. degree in software engineering at School of Computer Science and technology, Tianjin Chengjian University, Tianjin, China. His research interests include mobile edge computing and multi-objective optimization algorithm. E-mail: 1138896661@qq.com.

ZHANG Qing-hua received her B.S. degree from Zhoukou normal University, Henan, China. Now, she works at Tianjin Chengjian University, Tianjin, China. Her research interests include library management informatization, artificial intelligence algorithm. E-mail: 304854105@qq.com.