Journal of Engineering, Science and Mathematics

Volume 02, Number 02, Pages 56-72, Year 2023

https://jesm.in/archives/

Research Paper

Edge Computing-Based Task Offloading Algorithm for 5th Gen Internet of Vehicles Communication

Ismail Keshta[1], Suchitra Bala[2]

[1]Computer Science and Information Systems Department, College of Applied Sciences, AlMaarefa University, Riyadh, Saudi Arabia

[2]ICT & Cognitive Systems, Sri Krishna Arts and Science College, Coimbatore, Kuniyamuthur, Tamil Nadu, 641008, India

Correspondence should be addressed to Ismail Keshta; imohamed@um.edu.sa

Handling Editor: Surbhi Gupta

The new in-vehicle tasks that are emerging are requiring stronger communication and processing capabilities due to the Internet of Vehicles' rapid expansion. Highway car users may now get more dependable and fast services thanks to the widespread installation of 5G millimeter-wave base stations. Simultaneously, mobile edge computing (MEC) technology reduces transmission latency by deploying MEC servers surrounding user terminals that have computation and storage capacity to provide computer services for onboard operations. To address the allocation challenge, the combined optimization problem of computation and communication resources is modeled as a 0-1 mixed-integer linear programming problem. Initially, the primary optimization issue is divided into two smaller issues: allocating resource blocks and making decisions on unloading. Swarm algorithms are used to solve each of the subproblems independently. Finally, using the heuristic approach, the subproblems are solved iteratively to get the best resource block allocation plan and unloading decision vector. The outcomes of the simulation demonstrate that the suggested algorithm is capable of fulfilling the demands of every onboard task. Simultaneously, the system's average latency is reduced.

Keywords: *Mobile Edge Computing, Task Offloading Algorithm, 5 Generation Network, Internet of Vehicles, Linear Programming*

1. Introduction

The Internet of Vehicles (IoV) technology is an integral part of intelligent transportation and is emerging as one of the application scenarios of the Internet of Things [1]. Many new service applications such as unmanned smart driving, in-vehicle ultra-clear video, and augmented reality are constantly being applied. The emergence of the IoV also presented higher requirements for network resources in the IoV. Since cloud computing and edge computing technology can provide many computing and storage resources, combining IoV and edge computing makes up for the lack of vehicle data processing capabilities w.r.t. the development of the IoV [2]. Since vehicle users have specific computing capabilities, in the IoV architecture that supports cloud computing, vehicle users' computing tasks may be calculated locally or migrated to the cloud for processing. [3]. Moreover, mobile edge computing (MEC) technology deploys MEC servers with computing and storage resources around users, reducing the unloading delay of onboard tasks. Due to the close relationship between computing resources, communication, and storage resources, when resources are allocated, how to reasonably utilize limited computing, communication, and storage resources to provide diversified services to more users has become one of the critical issues in the current IoV.

Currently, the research on MEC in the IoV at home and abroad mainly focuses on the strategic analysis of resource allocation, computing offloading, and content caching. Reference [4] selected the appropriate vehicle user equipment for the vehicle user equipment according to the classification results of the Bayesian classifier. The offload mode is adopted, and a solution based on Q-learning is proposed, which realizes the balance between the delay requirement and energy consumption during the task execution process. Reference [5] offered an online learning method to minimize task delay. To some extent, alleviate the problem of limited resources in mobile edge computing. In Reference [6], based on the simulated annealing algorithm, a task is offloaded, and a resource allocation scheme is proposed, which performs cooperative offload calculation processing for jobs with higher priority. Reference [7] studied vehicle movement information and offers an iterative resource allocation scheme considering different modes of transport and channel distribution information that improves the utilization rate of wireless resources in the IoV. Reference [8] analyzed the cost and profit of replicas on the MEC server and proposes an adaptive replication scheme of the MEC. The algorithm dynamically adjusts the image in each time interval. It forwards the user request to the adjacent MEC server through this algorithm to ensure the load balancing of the MEC server. This scheme shortens the average response time and improves the service quality of the data packets. The strict delay requirements for safety-related applications are a significant challenge for intelligent transportation systems.

Reference [9] introduced the measurement of perception-reaction time (PRT). It used information by combining the central network technology and the fog virtualization method, a novel fog resource scheduling mechanism is proposed to minimize PRT. Aiming at the problem that computing resources are easily limited in edge computing, the Reference [10] suggested the data centre collaborative sharing scheme that has been dramatically improved regarding task request blocking tasks and service delays. Reference [11] comprehensively considered the order of scheduling of subtasks and the order of processing of different service nodes in task unloading. A task scheduling and offloading technique is proposed to cope with the offloading multiple independent subtasks problem of a single vehicle. Reference [12] suggested a combined auction mechanism that matches the MEC server and the user terminal. The task is offloaded to the optimal MEC server for the calculation to achieve the shortest delay in task completion. To reduce the load pressure of the edge server, Reference [13] fully utilized the computing resources of the vehicle itself and proposes a joint task offloading & resource allocation. For processing, the vehicle can transfer the generated tasks to edge servers or other vehicles. To provide efficient and stable data transmission for users of high-speed moving vehicles, Reference [14] proposed a precache and task allocation method based on deep reinforcement learning.

Reference [15] suggested a resource block allocation scheme (resource block, RB), where road side units (RSUs) are dynamically based on the count of available RBs and the arrival rate of service requests. Users allocate resources to reduce service request rejection rate and improve RB utilization. The high-speed mobility of vehicles brings severe challenges to the continuity of communication, a dynamic unloading scheduling scheme. The scheme comprehensively considers the vehicle speed, cell coverage, data transmission rate, and other factors to obtain the optimal unloading ratio and transmission unit. At the same time, the scheme has good adaptability to changes in vehicle speed and wireless environment. Therefore, it has a significant impact on delay and energy consumption. Reference [17] jointly optimized the unloading decision and resource allocation problems. The optimal communication resource and the computing resource allocation scheme are obtained by solving the optimization problem to reduce the delay of the whole system.

To optimize the overall long-term projected return of the in-vehicle cloud computing system, Reference [18] presented an optimum allocation strategy for computing resources. The revenue and cost of the in-vehicle cloud computing system and the variability of available resources were comprehensively considered to obtain the procedure reward and model the optimization problem as an infinite-time semi-Markov decision process with state space, action space, reward model, and transition probability distribution defined by the onboard cloud computing system. Use an iterative algorithm to solve what should be taken in a specific state. In Reference [19], considering the heterogeneity of vehicle computing resources and the independence of computing tasks, an efficient task scheduling scheme is proposed. This scheme divides the computing tasks generated by the target vehicle into multiple independent subtasks which are unloaded to different vehicles for processing, and an optimization model is established with the task completion delay as the optimization objective. Reference [20] proposed a new task scheduling scheme based on a genetic algorithm to improve vehicle performance. The time complexity of solving the problem is reduced while computing resource utilization.

The literature above studies communication and computing collaboration technology, and the research mainly focuses on one MEC server serving multiple vehicle users simultaneously. On the contrary, collaborative computing between MEC servers is rarely considered. Therefore, to optimize communication and processing resources together, a joint optimization scheme is proposed. The system adopts a partial offloading strategy to offload vehicle tasks to local and multiple MEC vehicles to minimize the average delay in task completion of all users [21].

2. SYSTEM MODEL

2.1. Network Model

The expressway system model is shown in Figure 1. 5G millimeter-wave microbase stations (BS) are installed on the side of the expressway at certain intervals and the BS are connected to the core network through optical fibers. The technology communicates with 5G millimeter wave micro-BS, and multiple 5G micro-BS form a cluster and share a MEC server. The MEC server can provide computing resources for vehicle tasks, shorten the task processing delay, and reduce vehicle energy consumption. The vehicle will generate tasks that require a lot of computing resources during the driving process. These tasks can be processed locally in the vehicle or offloaded to the MEC server for processing via the V2I link. Assuming limited computing resources of a single MEC server where the MEC computing resources are insufficient at the time, the computing task can be offloaded to the neighboring MEC servers within one hop to two hops away through the optical fiber for collaborative computing. The set of adjacent MEC servers is M = 1,2,.... 1, 2,, n represents the set of vehicles in the group, the total number of vehicles ψ=|N| (N= {1,2, {1, 2, ⋯, n}). Consider that the amount of computing task data generated by vehicle n is s_n, since the task can be processed locally, it can also be offloaded to multiple. Therefore, this paper divides the computing task generated by the vehicle into numerous subtasks and further defines λ_n=[λ_1,λ_2,⋯,λ_m] as the vector of the task, λ_0 represents the local processing ratio in the vehicle, λi represents the proportion of offloading to the MEC server i for processing.

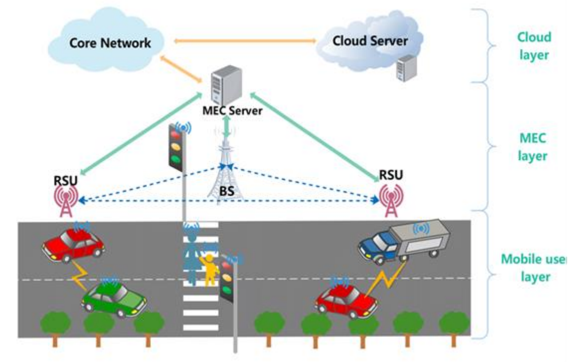

Figure 1: Highway system model

2.2. Communication Model

In the above scenario, S = 1, 2, s, s represents the set of 5G millimeter wave base stations in the cluster, the available resource blocks in the group are denoted as kK, and the bandwidth of each resource block is ω_0. Assume that all the resources in the collection are the 5G millimeter wave base station of 5G that shares these K resource blocks to serve the vehicles within its coverage. According to the literature [17], in the 60GHz millimeter wave environment, the channel gain of vehicle n and base station i in the resource block k is:

$$h_{i,n}^{k} = 10\delta_{i,n}\lg(d_{i,n}) + \frac{15d_{i,n}}{1000} \tag{1}$$

Among them, $\delta_{i,n}$ represents the loss index between vehicle n and base station i, and $d_{i,n}$ is the Euclidean distance between vehicle n and base station i. In addition, 15 $d_{i,n}$ /1000 represents the millimeter wave signal of 60 GHz in the atmosphere—attenuation loss.

Therefore, when vehicle n uses the resource block k to communicate with base station i, the signal-to-interference and noise ratio of base station is:

$$SINR_{i,n}^{k} = \frac{p_n h_{i,n}^{k}}{\sigma_{n,k}^{2} + I_{m,j}^{k} + I_0} \quad (2)$$

$$I_{m,j}^{k} = \sum_{\substack{m \in \phi, m \neq n \\ j \in \phi, j \neq i}} p_m h_{j,m}^{k} \quad (3)$$

Among them, p_n represents the maximum transmit power of vehicle n, $\sigma_{n,k}^{2}$ represents the Gaussian white noise power of resource block k, $I_{m,j}^{k}$ represents the interference from other base stations and vehicles, φ represents the difference between the vehicle in the current cluster and the vehicle in the adjacent cluster Set, φ represents the set of base stations in the existing collection and base stations in contiguous groups, I_0 represents the interference from satellite communication. As per Shannon's formula, the rate of data transmission w.r.t. vehicle n served by base station i using subchannel k is:

$$r_{i,n}^{k} = \omega_0 \sigma_{i,n}^{k} log_2(1 + SINR_{i,n}^{k}) \quad (4)$$

Among them, $\sigma_{i,n}^{k}$ represents the resource block allocation indicator factor. When $\sigma_{i,n}^{k}$=1, it means that 5G millimeter-wave base station i allocates the resource block k to its serving vehicle; when $\sigma_{i,n}^{k}$ =0, it means that 5G millimetre-wave base station i does not allocate the resource block k to its service vehicle. Therefore, the transmission rate of vehicle n served by base station i is:

$$r_{i,n} = k \in K r_{i,n}^{k} \quad (5)$$

Since the volume of data of tasks generated by vehicle n is s_n , and the proportion of subtasks unloaded into the vehicle for processing is λ_0 , the data volume of subtasks unloaded to MEC for processing is (1-λ_0)s_n, regardless of MEC the subtask of MEC. Since the data volume of the calculation result is small, the receiving hold of the calculation result can be ignored. Therefore, the transmission delay of vehicle n served by base station i unloading the subtask to the MEC server is as follows:

$$D_{i,n}^{t} = \frac{(1-\lambda_0)s_n}{r_{i,n}} \quad (6)$$

2.3 Computational model

Since the vehicle has specific computing resources, vehicle computing tasks can be processed locally or offloaded to the MEC server. The MEC server performs cooperative processing to shorten the task processing delay.

2.3.1 Local computing model

The computational delay required by the vehicle to process a subtask with a data volume of $\lambda_0 s_n$ locally is:

$$D_n^c = \frac{\lambda_0 s_n C_n}{f_n} \quad (7)$$

Among them, C_n represents the number of CPU cycles that the vehicle needs to process 1 bit of data, and f_n represents the computing resources of the vehicle.

2.3.2 MEC server computing model

The computational delay required by the MEC_i server to process a subtask with a data volume of $\lambda_i s_n$ is:

$$D_{MEC_i}^{c} = \frac{\lambda_i s_n C_{MEC_i}}{f_{MEC_{i,n}}} \quad (8)$$

Among them, C_{MEC_i} represents the count of CPU cycles that the MEC server i takes to process 1 bit of data, and $f_{MEC_{i,n}}$ represents the computing resources allocated by the MEC server i to vehicle n.

2.4 Queuing model

Although the CPU power of the MEC server is greater than that of the vehicle terminal, at the same time the MEC server has a high load. When a large number of tasks are offloaded to MEC server for processing, a high queueing delay may occur, resulting in the processing of the entire task, and the delay increases. Assuming that the MEC server task processing can be regarded as a queueing system, the average task arrival rate is γ, the average service rate is μ, and the service density is ρ. When subtasks are offloaded to the MEC server for processing, their queueing is. The waiting time is:

$$D_{MEC}^{w} = E(D_{MEC}^{queue} + D_{MEC}^{service}) \tag{9}$$

Among them, D_{MEC}^{queue} represents the average queuing delay w.r.t. MEC server, and $D_{MEC}^{service}$ represents the average service delay of the MEC server.

$$E\left(D_{MEC}^{queue}\right) = \frac{\rho}{1-\rho}\frac{1}{\mu} \tag{10}$$

$$E\left(D_{MEC}^{service}\right) = \frac{1}{\mu} \tag{11}$$

From Equation (10) and Equation (11), we can get:

$$D_{MEC}^{w} = \frac{\rho}{1-\rho}\frac{1}{\mu} \tag{12}$$

Therefore, the total delay of vehicle n processing tasks is represented by the transmission delay of offloading the subtask to the MEC server, the calculation delay of the subtask locally and at the MEC server, and the queuing delay, and its expression is

$$D_n = max\left\{\frac{\lambda_0 s_n C_n}{f_n}; \frac{(1-\lambda_0)s_n}{r_{i,n}} + \beta\, max\left\{\frac{\lambda_i s_n C_{MEC_i}}{f_{MEC_{i,n}}} + \frac{\rho}{1-\rho}\frac{1}{\mu}\right\}\right\} \tag{13}$$

Among them, β=$\begin{cases}1, \lambda_0 \neq 1 \\ 0, \lambda_0 = 1\end{cases}$,indicates that when the tasks are all processed in the vehicle, the MEC server does not generate calculations and queueing delays.

2.5 Optimization Goals

In the above scenario, the vehicle offloads the generated computing tasks to the MEC server for processing through the 5G millimeter wave base station. The communication resources, computing resources, and the unloading ratio obtained by the vehicle jointly determine the completion time of the business, so this paper uses all vehicle tasks to complete. Meanwhile, the optimization objective is considered, and the communication resources of the base station are considered, and the following optimization model is established:

$$\mathrm{p}_0\text{: } \min_{\lambda_0, \alpha_{i,n}^k} \frac{1}{N}\sum_{n\in N} D_n \tag{14}$$

s.t.

$$C_1\text{: } \alpha_{i,n}^k \in \{0,1\}, \forall i \in S, n \in N, k \in K$$

$$C_2\text{: } \lambda_i \in [0,1], \forall i \in MU\{0\}$$

$$C_3\text{: } \sum_{i=0}^{m} \lambda_i = 1, \forall i \in MU\{0\}$$

$$C_4\text{: } \sum_{n\in N_i}\sum_{k\in K} \alpha_{i,n}^k \leq |K|, \forall i \in S$$

$$C_5\text{: } \sum_{n\in N_i} \alpha_{i,n}^k \leq 1, \forall i \in S, k \in K$$

$$C_6\text{: } D_n \leq \frac{d}{v_n}, \forall n \in N$$

$$D_n = max\left\{\frac{\lambda_0 s_n C_n}{f_n}; \frac{(1-\lambda_0)s_n}{r_{i,n}} + \beta max\left\{\frac{\lambda_i s_n C_{MEC_i}}{f_{MEC_{i,n}}} + \frac{\rho}{1-\rho}\frac{1}{\mu}\right\}\right\}$$

Among them, constraint condition C_1 restricts the resource block allocation indicator factor to be only a binary variable, constraint condition C_2 represents the value range of the unloading scale factor, and the constraint condition C_3 means that the sum of the unloading scale factor for vehicle n is 1. Constraint condition C_4 means that the sum of the unloading scale factor is 1. Furthermore, the sum of the number of resource blocks allocated to the vehicle cannot exceed the number of resource blocks available at the base station. Constraint C_5 means that base station i cannot assign the same resource blocks to the users it serves. Finally, constraint C_6 restricts the vehicle from completing the calculation within the specified distance. Task d represents the specified distance, and v_n means the vehicle's average speed of the vehicle.

3. SOLVING THE OPTIMIZATION PROBLEM

Since the optimization variable resource block allocation indicator factor is a binary variable and the unloading scale factor is a continuous variable, the optimization problem denoted by P_0 is regarded as 0-1 mixed-integer nonlinear programming problem 0-1. The optimization variables are coupled in constraint C_5, and the conventional convex optimization algorithm is difficult to use directly to solve the problem. In this study, the optimization problem P_0 is considered decomposed into the resource block allocation subproblem and the unloading decision subproblem to be solved separately.

3.1 Resource Block Allocation Sub-Problem

The aim of optimization problem P_0 is to lower downs the completion time of vehicle computing tasks. For vehicle users, the allocation of resource blocks is to obtain the optimal data transmission rate, thus reducing the time for data upload to the MEC server. First, the task unloading ratio vector of each vehicle user is initialized. It is assumed that the computing tasks are evenly distributed among the vehicles and each MEC, that is, λ_i=1/(m+1). After obtaining the vector of the vehicle's Minimum data transfer rate $r_{i,n}^{min}$.

$$r_{i,n}^{min} \geq \frac{(1-\lambda_0)s_n}{\frac{d}{v_n} - max\left\{\frac{\lambda_i s_n c_{MEC_i}}{f_{MEC_{i,n}}} + \frac{\rho \quad 1}{1-\rho\mu}\right\}} \quad (15)$$

In this paper, it is assumed that the access of vehicle users is known, so the base station in the cluster uses the water injection algorithm to allocate resource blocks in the first stage for the users it serves. First, calculate the SINR values of all users when they occupy different resource blocks, and the base station will prioritize the allocation of resource blocks. The resource block with the highest SINR value is allocated to the user it serves. If the resource block has been assigned to other users, the base station selects the resource block with the highest SINR value from the remaining resource blocks and gives it to the user block allocation.

After completion, calculate the data transmission rate of all users, and perform subchannel allocation in the second stage for users whose data transmission rate does not meet the requirements until the data transmission rate of each vehicle user meets the requirements. If there are remaining resource blocks, vehicle users are arranged in ascending order of the data transmission rate, and the resource blocks with the enormous SINR value are sequentially allocated to vehicle users with the lower data transmission rate until all resource blocks are given, to ensure fairness among users. The algorithm for resource block allocation based on the water injection algorithm is mentioned in Algorithm 1.

Algorithm 1: Resource block allocation algorithm based on water injection algorithm.

1. Initialization stage. Available resource blocks k K in the cluster; Set table of 5G millimeter wave base stations in the clustershown as S={1, 2,⋯ , s}; N_i denotes the number of vehicles accessing base station i; U_i denotes base station I the serviced vehicle does not meet the data transfer rate requirement.
2. Calculation phase. Calculate the SINR value of each base station user on K resource blocks.
3. Allocation of resource blocks for the first stage
4. Fori=1:s
5. Forn=1:N_k
6. $\alpha_{i,n}^{k}$=argmax$SINR_{i,n}^{k}$
7. Assign $\alpha_{i,n}^{k}$ to vehicle n, and update the resource block set of resources at the same time; let K=K/k
8. EndFor
9. EndFor
10. Calculate the rate of data transmission rate of the vehicle served by base station i, and add users who do not meet the requirements and add to Collection UI
11. Forn=1: N_i
12. If$r_{i,n} \leq r_{i,n}^{min}$
13. $U_i = U_i U n$
14. End For
15. Second-stage allocation of resource blocks for users who do not meet the transmission rate
16. forn ∈U_i
17. $\alpha_{i,n}^{k}$=argmax$SINR_{i,n}^{k}$
18. Assign $\alpha_{i,n}^{k}$ to vehicle n, update the resource block set of resources simultaneously, and let K=K/k.
19. End For
20. Repeat steps 10-19 until all users meet the data transferRetirement.
21. Calculate the data transmission rate of the users served by base station i and sort them in ascending order.The remaining resources after allocation are allocated to vehicle users with lower transmission rates to ensure fairness of users.

3.2 Unloading the Decision Sub-problem

The optimal data transmission rate w.r.t. each vehicle user can be obtained by solving the sub-problem of resource block allocation so that the original optimization problem P_0 can be converted into an optimization problem about the unloading proportional vector.

$$P_1: \min_{\lambda} D_n \tag{16}$$

s.t.

$$C_1: \lambda_i \in [0,1], \forall i \in MU\{0\}$$

$$C_2: \sum_{i=0}^{m} \lambda_i = 1, \forall i \in MU\{0\}$$

$$C_3: D_n \leq \frac{d}{v_n}, \forall n \in N$$

This paper uses a particle swarm algorithm to solve the above sub-problems. Assuming that the number of MEC servers that vehicle users can uninstall is M (i.e. the search space dimension is M+1), and L particle representations are generated by initialization, and the expression is as follows:

$$W^{(l)} = (\lambda_0, \lambda_1, \cdots, \lambda_m) \tag{17}$$

Among them, l = 1, 2,…, L, each particle looks for the optimal solution according to the flight speed, and the individual optimal position passed by the particle l is recorded as $(p_{l0}, p_{l1}, \cdots, p_{lm})$, all the optimal work that the particle passes through is recorded as $(p_{g0}, p_{g1}, \cdots, p_{gm})$, and the particle's speed and position of the particle are updated according to equation (18):

$$v_d^{(l)}(t+1) = wv_d^{(l)}(t) + c_1 r_1 \left(p_{ld}(t) - W_d^{(l)}(t)\right) + c_2 r_2 \left(p_{gd}(t) - W_d^{(l)}(t)\right) \quad (18)$$

$$W_d^{(l)}(t+1) = W_d^{(l)}(t) + v_d^{(l)}(t+1) \quad (19)$$

Because the original particle swarm algorithm has a fast convergence speed in the early stage, and with iteration, the number of times increases, and the diversity of the population decreases, resulting in a stagnation phenomenon, there is the local optimal solution problem. The weight coefficient c_1 is used to represent the cognitive ability of the particle. The child will move to the optimal position in its history. Therefore, this paper solves the problem by iterating. In the solution process, c_1 is dynamically and non-linearly increased to make the learning ability of the particles continue. The improvement solves the problem of premature convergence, and the learning factor c_1 is carried out according to equation (20).

$$c_1 = c_{1min} + (c_{1max} - c_{1min})\frac{t}{T} \qquad (20)$$

Among them, c_{1min} is the minimum value ofc_1, c_{1max} is the maximum value ofc_1, t refers to the current count of iterations, and T refers to the set of maximum count of iterations. During the solution process, the particle's position is continuously adjusted, and its speed is updated so that the user can use the shortest possible number of iterations to complete the business. In this paper, the user's business completion time D_n is used as the fitness function U_n to update the optimal position of the particle. The value of the utility function value of the optimal particle position, which is currently global is U_g^{opt}. The value of the utility function of the optimal position w.r.t. particle l is U_l^{opt}, the optimal global position of the particle is denoted as W_g^{opt}, and the optimal experienced position by particle l is denoted as W_l^{opt}. The specific steps of the algorithm are shown in Algorithm 2.

Algorithm 2: Unloading proportional vector algorithm based on particle swarm optimization

1. Initialization stage: randomly initialize to generate l particles W(l)(t), set U=Un, U_g^{opt}=min$U^{(1)}$(t)
2. When tT
3. For l=1:L
4. Calculate the fitness value $U^{(1)}$(t) of the current particle
5. If $U^{(l)}$(t) ≤ U_1^{opt}
6. Update individual optimal: U_l^{opt} = $U^{(1)}$(t), W_1^{opt}= $W^{(l)}$(t)
7. End
8. End
9. Compare the fitness value of each particle with the global optimal utility function value to update the global the best position in the game:
10. U_g^{opt} =min U_1^{opt} , W_g^{opt} =min W_1^{opt}
11. For l=1:L
12. Update weight $c_1 = c_{1min} + (c_{1max} - c_{1min})\frac{t}{T}$
13. Update particle velocity $v^{(1)}(t+1) = wv^{(1)}(t) + c_1 r_1 \left(W_1^{opt}(t) - W^{(1)}(t)\right) + c_2 r_2 \left(W_g^{opt}(t) - W^{(1)}(t)\right)$
14. Update particle position $W^{(1)}(t+1)$=$W^{(1)}(t)$+$v^{(1)}(t+1)$
15. End
16. t = t+1
17. End

3.3 Joint Optimization Algorithm for the Resource Block and Offload Vector

Algorithm 1 obtains the optimal allocation strategy of resource blocks based on the water injection algorithm after fixing the unloading proportion vector. Algorithm 2 uses the particle swarm algorithm to obtain the optimal unloading proportion vector based on Algorithm 1. Based on the above algorithm, this paper proposes a joint optimization algorithm of resource block and offload proportional vector; the specific flow of the algorithm is shown in Algorithm 3.

Algorithm 3: Resource block and offload vector joint optimization algorithm.

1. Initialize the unloading scale vector λ= $\frac{1}{m+1}, \frac{1}{m+1}, \cdots, \frac{1}{m+1}$, and randomly initialize to generate l particles $W^{(1)}(t)$.
2. The optimal allocation strategy of resources blocks is obtained by solving Algorithm 1.
3. The optimal unloading ratio vector is obtained by solving Algorithm 2.
4. If the objective function of the optimization problem P1 does not converge, go to Step 2.

The time complexity of Algorithm 3 depends mainly on the time complexity of steps 2 and 3 in each round of iteration. Algorithm 1 traverses the set of resources for each user to find the optimal resource block. The user performs the allocation of the resource block in the second stage, so the time complexity of step 2 is O(|S|| N_k |+| U_i |). The particle swarm algorithm proposed in Algorithm 2 is a cyclic iterative process. In the iteration, each particle calculates its utility value and updates its optimal utility value, and at the same time, the optimal global position selects the highest utility value, the position of a suitable particle. Each particle also needs to upgrade its velocity & place, so the time complexity of step 3 is O(2LT). To sum up, assuming that Algorithm 3 reaches convergence after max_ite times, therefore, its time complexity is O(|S| |N_k|+|U_i|+2LT)max_ite).

4. SIMULATION ANALYSIS AND PERFORMANCE EVALUATION

In this paper, the performance of the joint optimization algorithm of the resource block and the load vector is simulated and verified in Python, and the specific simulation parameters are listed in Table 1.

4.1 Simulation Environment and Parameter Settings

The system bandwidth is set to 800MHz, the number of resource blocks allocated to each 5G millimeter-wave micro base station is 50, and the bandwidth of each resource block is 16MHz. The coverage radius of the micro base station is set to 30m, assuming that the arrival process of the vehicle obeys the Poisson distribution, the speed of the vehicle is 28 m/s, the noise power spectral density is fixed to -174 (dBm/Hz), the computing power of the vehicle is set to 5 GHz, the processing power of the MEC server is 30 GHz, and the average arrival rate of the tasks of the MEC server is 0.2~ 0.3, the average service rate is 0.35-0.45, and the uplink transmit power of the vehicle terminal is 50mW.

Table 1: Simulation parameters

Parameter	Parameter settings
Task data/Gbit	0.5~1

MEC CPU processing power/GHz	35
Local terminal CPU processing capacity/GHz	6
CPU cycles required to process 1 bit	1440
task arrival rate	0.25~0.35
Channel bandwidth/MHz	800
Available resource blocks	50
Vehicle terminal transmit power/mW	50
Noise Power Spectral Density/(dBm/Hz)	-175

4.2 Fast Unloading Algorithm Based on Q-learning

Taking into account the two factors of delay and energy consumption, the optimization model is established with the benefit of the vehicle terminal as the optimization objective. The Q-learning-based fast unloading algorithm proposed in the literature [20] uses the instant reward and experience reward obtained after the vehicle terminal makes the unloading decision. The cumulative discount reward is updated in combination and finally the optimal resource allocation scheme and the task offloading strategy are obtained to maximize the benefits of the vehicle terminal.

4.3 Simulation Results and Analysis

Figure 2 shows the effect of the size of the population of particle swarms on the convergence of the algorithm. It can be observed from the figure that when the population size is small, the convergence speed of the algorithm is slow, and there is a risk of falling into a locally optimal solution. When the population size increases, the algorithm in this paper has a better optimization ability and a fast convergence speed. Still, when the population number increases to a certain level, it will no longer have a significant effect.

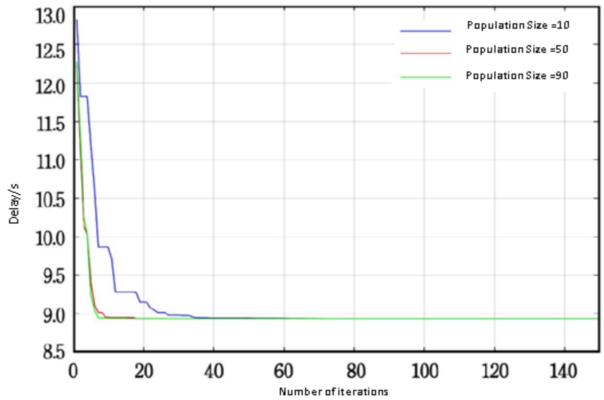

Figure 2: Comparison of different population sizes

Table 2 and Figure 3 show the difference in the task processing delay of several algorithms. Each algorithm's processing time for a job rises as the task's data volume grows. When all tasks are offloaded to the MEC server due to underutilization, the local computing resources of the vehicle cause the task load of the MEC server to be too large, thereby increasing the task transmission delay and queuing delay. For example, the Q-learning-based fast unloading algorithm is performed every time the vehicle terminal passes through a MEC server service area. Therefore, multiple tasks are unloaded, increasing the task processing delay. In addition, when the amount of input task data is consistent, the proposed algorithm in this paper fully utilized the vehicle's local computing resources and the MEC server's help to achieve the minimum processing delay.

Table 2: Comparison of Task Processing Delay

Delay/s	Task Volume/MB	Algorithm	Uninstall all to MEC	Quick learning based on Q off-loading algorithm
6	500	7.7	6.92	9.6
7	550	8.25	7.42	10.2
8	600	8.36	7.52	10.32
9	650	9.24	8.32	11.28
10	700	9.9	8.92	12
11	750	10.12	9.12	12.24
12	800	10.78	9.72	12.96
13	850	11.55	10.42	13.8
14	900	12.1	10.92	14.4
15	950	12.43	11.22	14.76
16	1000	13.2	11.92	15.6

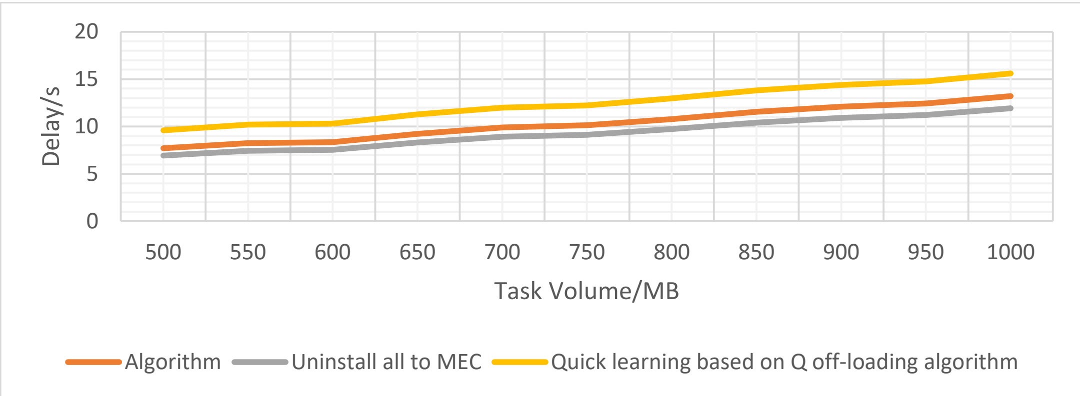

Figure 3: Comparison of Task Processing Delay of Different Algorithms

Table 3 and Figure 4 show the difference in the vehicle task processing delay when the average task arrival rate (i.e. AR_{avg}) is different. As can be observed from the simulation results, with the increase of the average task arrival rate, the task processing delay also increases. This is because, with increasing the average arrival rate of tasks, the task load of the MEC server also increases, resulting in the accumulation of functions, thus increasing the queueing delay.

Table 3: Comparison of the average arrival rate of different tasks

Delay/s	Task Volume/MB	AR_{avg}= 0.20	AR_{avg}= 0.25	AR_{avg}= 0.30
6	500	7	7.5	8
7	550	7.5	8	8.5

8	600	7.6	8.1	8.6
9	650	8.4	8.9	9.4
10	700	9	9.5	10
11	750	9.2	9.7	10.2
12	800	9.8	10.3	10.8
13	850	10.5	11	11.5
14	900	11	11.5	12
15	950	11.3	11.8	12.3
16	1000	12	12.5	13

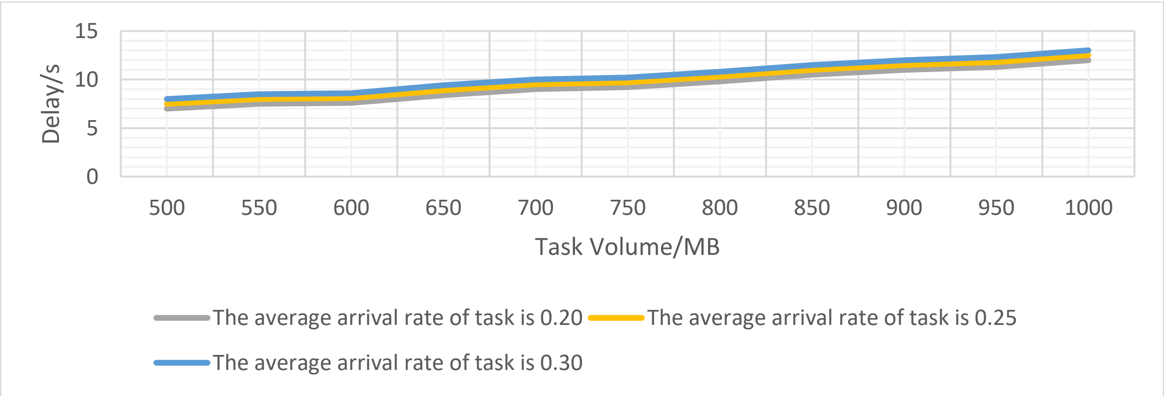

Figure 4: Comparison of the average arrival rate of different tasks

Table 4 and Figure 5 show the task processing of each algorithm at different vehicle speeds. The simulation results show that the task processing delay of each algorithm does not increase or decrease with the acceleration of the vehicle speed. Still, in certain conditions, it fluctuates around a value, which shows that the algorithm in this paper has good stability.

Table 4: Comparison of different speeds

Delay/s	Task Volume/MB	Algorithm	Uninstall all to MEC	Quick learning based on Q off-loading algorithm
9	18	10	11.2	12.4
9.5	20	9.6	10.8	12
10	22	9.8	11	12.2
10.5	24	9.5	10.7	11.9
11	26	9.7	10.9	12.1
11.5	28	10	11.2	12.4
12	30	9.6	10.8	12
12.5	32	9.4	10.6	11.8
13	34	9.7	10.9	12.1
13.5	36	9.5	10.7	11.9
14	38	9.7	10.9	12.1

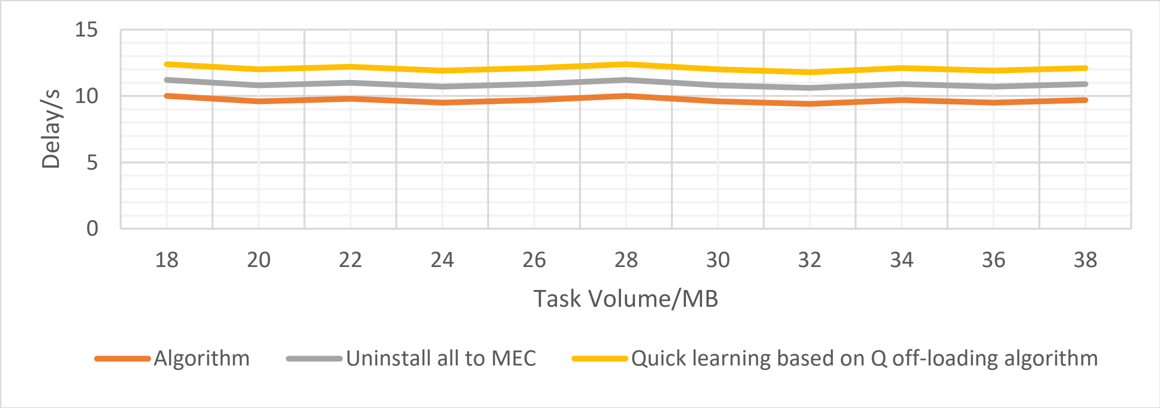

Figure 5: Comparison of different speeds

Table 5 and Figure 6 show the difference in vehicle task processing delay when the average service rate (SR_{avg}) of the MEC server is different. The average service rate represents the number of users the MEC server can serve per unit of time. With the increase in the average service rate, the number of accumulated tasks is reduced, so the delay in waiting for lessons is reduced. For example, from the figure it can be observed that when the task volume is 800M and the average service rate is 0.35, the task processing delay is reduced by 8.3% and 22.3% respectively.

Table 5: Processing delay w.r.t. different average service rate of the MEC server

Delay/s	Task Volume/MB	SR_{avg}= 0.35	SR_{avg}= 0.40	SR_{avg}= 0.45
6	500	8.45	7.67	10.35
7	550	9	8.17	10.95
8	600	9.11	8.27	11.07
9	650	9.99	9.07	12.03
10	700	10.65	9.67	12.75
11	750	10.87	9.87	12.99
12	800	11.53	10.47	13.71
13	850	12.3	11.17	14.55
14	900	12.85	11.67	15.15
15	950	13.18	11.97	15.51
16	1000	13.95	12.67	16.35

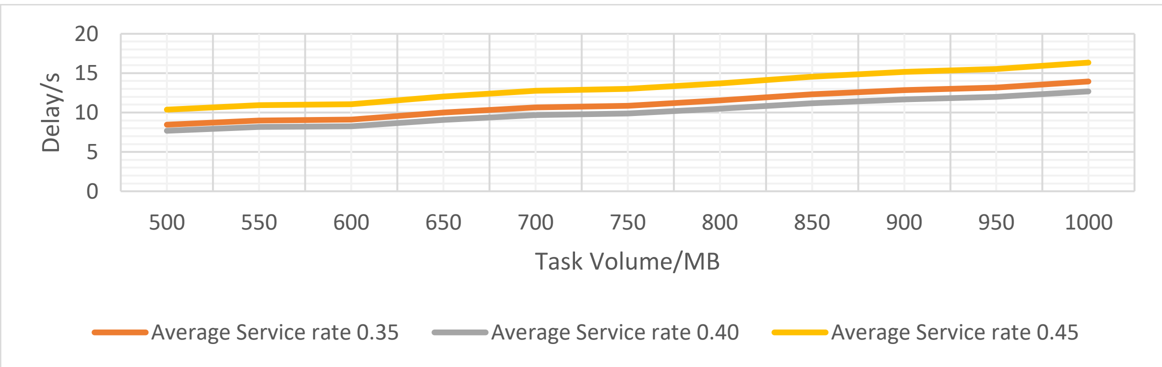

Figure 6: Processing delay w.r.t. different average service rate of the MEC server

5. Conclusions

This paper mainly studies vehicle task unloading decisions and resource block allocation problems in the millimeter-wave high-speed mobile scenario. It proposes a joint optimization algorithm of resource blocks and unloading vectors to solve the optimal resource block allocation strategy and to unload to minimize all users' average task processing delay. Since the original optimization problem is a 0-1 mixed-integer nonlinear programming problem, this paper decomposes it into a resource block allocation sub-problem and an offloading decision sub-problem and uses water injection. Algorithms and particle swarm optimization are solved separately to obtain the optimal strategy. Compared to prominent algorithms, the proposed joint optimization algorithm of resource block and unload vector fully utilizes the computing resources of the vehicle and the MEC server to achieve a minor task processing delay.

References

[1]. P. Roy, R. K. Vishwakarma, A. Jain and R. Singh, "Multiband millimeter wave antenna array for 5G communication," 2016 International Conference on Emerging Trends in Electrical Electronics & Sustainable Energy Systems (ICETEESES), 2016, pp. 102-105, DOI: 10.1109/ICETEESES.2016.7581361.

[2]. S. S. Max Chung and S. Tuan, "Preliminary Design of a Waveguide-Fed Milimeter Wave Metasurface Antenna with LCD Controlled Array Factor for 5G User Equipment," 2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), 2019, pp. 1-2, DOI: 10.1109/ISPACS48206.2019.8986311.

[3]. Cheng, H., Yuan, S., Li, W., Yu, X., Liu, F., Liu, X., & Bezabih, T. T. (2024). De-accumulated error collaborative learning framework for predicting Alzheimer's disease progression. Biomedical Signal Processing and Control, 89, 105767.

[4]. Luvembe, A. M., Li, W., Li, S., Liu, F., & Wu, X. (2024). CAF-ODNN: Complementary attention fusion with optimized deep neural network for multimodal fake news detection. Information Processing & Management, 61(3), 103653.

[5]. S. S. M. Chung, C. -T. Wu, Y. -C. Chuang and H. -C. Hsieh, "Preliminary design of 94 GHz E-band phase array antenna for future mobile communication," 2016 Asia-Pacific International Symposium on Electromagnetic Compatibility (APEMC), 2016, pp. 899-902, DOI: 10.1109/APEMC.2016.7522903.

[6]. A. Turkmen et al., "Coverage Analysis for Indoor-Outdoor Coexistence for Millimetre-Wave Communication," 2019 UK/ China Emerging Technologies (UCET), 2019, pp. 1-4, DOI: 10.1109/UCET.2019.8881890.

[7]. N. Nguyen-Trong and M. Ikram, "Multiple-Open-Ended-Slot Antenna for Integrated 4G/5G Mobile Application," 2021 15th European Conference on Antennas and Propagation (EuCAP), 2021, pp. 1-3, DOI: 10.23919/EuCAP51087.2021.9411044.

[8]. Rida, I., Al Maadeed, S., Jiang, X., Lunke, F., & Bensrhair, A. (2018, April). An ensemble learning method based on random subspace sampling for palmprint identification. In 2018 IEEE International conference on acoustics, speech and signal processing (ICASSP) (pp. 2047-2051). IEEE.
[9]. Gera, T., Singh, J., Mehbodniya, A., Webber, J. L., Shabaz, M., & Thakur, D. (2021). Dominant feature selection and machine learning-based hybrid approach to analyze android ransomware. Security and Communication Networks, 2021, 1-22.
[10]. M. A. Mustafar et al., "Milimeter-Wave Beamforming MIMO Antenna Design for 5G Wireless Applications," 2018 IEEE International RF and Microwave Conference (RFM), 2018, pp. 9-12, DOI: 10.1109/RFM.2018.8846545.
[11]. Z. Ying, O. Zander, M. Chung, L. Liu and F. Tufvesson, "Antenna Designs for a Milimeter Wave Massive MIMO Testbed with Hybrid Beamforming," 2021 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (APS/URSI), 2021, pp. 1241-1242, DOI: 10.1109/APS/URSI47566.2021.9703870.
[12]. N. Saba, L. Mela, M. U. Sheikh, J. Salo, K. Ruttik and R. Jäntti, "Rural Macrocell Path Loss Measurements for 5G Fixed Wireless Access at 26 GHz," 2021 IEEE 4th 5G World Forum (5GWF), 2021, pp. 328-333, DOI: 10.1109/5GWF52925.2021.00064.
[13]. E. V. P. Anjos, D. M. M. . -P. Schreurs, G. A. E. Vandenbosch and M. Geurts, "A 24 - 30 GHz Ultra-Compact Phase Shifter Using All-Pass Networks for 5G User Equipment," 2020 IEEE/MTT-S International Microwave Symposium (IMS), 2020, pp. 217-220, DOI: 10.1109/IMS30576.2020.9223788.
[14]. Wawale, S. G., Shabaz, M., Mehbodniya, A., et.al. (2022), Biomedical Waste Management Using IoT Tracked and Fuzzy Classified Integrated Technique, Human-centric Computing and Information Sciences, 12, 32.
[15]. Zheng, W., Mehbodniya, A., Neware, R., Wawale, S. G., Ganthia, B. P., & Shabaz, M. (2022). Modular unmanned aerial vehicle platform design: Multi-objective evolutionary system method. Computers and Electrical Engineering, 99, 107838.
[16]. Parashar, A., Parashar, A., Ding, W., Shekhawat, R. S., & Rida, I. (2023). Deep learning pipelines for recognition of gait biometrics with covariates: A comprehensive review. Artificial Intelligence Review, 1-65.
[17]. M. Kelner, C. Ziółkowski and B. Uljasz, "Evaluation of angular dispersion for various propagation environments in emerging 5G systems," 2018 22nd International Microwave and Radar Conference (MIKON), 2018, pp. 637-641, DOI: 10.23919/MIKON.2018.8405311.
[18]. Rida, I. (2018). Feature extraction for temporal signal recognition: An overview. arXiv preprint arXiv:1812.01780.
[19]. E. Onggosanusi et al., "Modular and High-Resolution Channel State Information and Beam Management for 5G New

Radio," in IEEE Communications Magazine, vol. 56, no. 3, pp. 48-55, March 2018, DOI: 10.1109/MCOM.2018.1700761.

[20]. H. Mei, Q. Zhao and L. Peng, "Energy-Efficiency in Cache-Enabled mmWave Cellular Networks," 2019 International Conference on Networking and Network Applications (NaNA), 2019, pp. 107-112, DOI: 10.1109/NaNA.2019.00028.

[21]. Panwar, P., Shabaz, M., Nazir, S., Keshta, I., Rizwan, A., & Sugumar, R. (2023), Generic Edge Computing System for Optimization and Computation Offloading of Unmanned Aerial Vehicle, Computers and Electrical Engineering, 109, 108779.