Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH Open Access

A collaborative scheduling strategy for IoV computing resources considering location privacy protection in mobile edge computing environment

Meiyu Pang[1,2*], Li Wang[1,2] and Ningsheng Fang[3]

Abstract

This paper proposes a collaborative scheduling strategy for computing resources of the Internet of vehicles considering location privacy protection in the mobile edge computing environment. Firstly, a multi area multi-user multi MEC server system is designed, in which a MEC server is deployed in each area, and multiple vehicle user equipment in an area can offload computing tasks to MEC servers in different areas by a wireless channel. Then, considering the mobility of users in Internet of vehicles, a vehicle distance prediction based on Kalman filter is proposed to improve the accuracy of vehicle-to-vehicle distance. However, when the vehicle performs the task, it needs to submit the real location, which causes the problem of the location privacy disclosure of vehicle users. Finally, the total cost of communication delay, location privacy of vehicles and energy consumption of all users is formulated as the optimization goal, which take into account the system state, action strategy, reward and punishment function and other factors. Moreover, Double DQN algorithm is used to solve the optimal scheduling strategy for minimizing the total consumption cost of system. Simulation results show that proposed algorithm has the highest computing task completion rate and converges to about 80% after 8000 iterations, and its performance is more ideal compared with other algorithms in terms of system energy cost and task completion rate, which demonstrates the effectiveness of our proposed scheduling strategy.

Keywords: Mobile edge computing, Privacy of vehicle location, Computational collaborative scheduling strategy, Kalman filter, Double DQN algorithm

Introduction

According to relevant statistics, the number of connected vehicles on the road will reach 2.5 billion. This enables many new in-vehicle services, such as autonomous driving capabilities, to be realized. Connected cars have become a reality, and the function of connected cars is rapidly expanding from luxury cars and high-end brands to large-scale mid-size cars [1]. The increase and creation of digital content in automobiles will drive the demand for more complex infotainment systems, which creates opportunities for the development of application processors, graphics accelerators, displays and human-machine interface technologies [2]. As a typical application of Internet of Things in the automotive industry, Internet of Vehicles (IoV) is regarded as a next-generation intelligent transportation system with great potential by equipping vehicles with various sensors and communication modules [3, 4]. In recent years, the automotive industry is undergoing critical and tremendous changes, many new in-vehicle applications, services and concepts have been proposed .

* Correspondence: pangmeiyu@126.com
[1]School of Internet of Things Engineering, Wuxi Taihu University, Wuxi 214064, Jiangsu, China
[2]Jiangsu Key Construction Laboratory of IoT Application Technology, Wuxi 214064, Jiangsu, China
Full list of author information is available at the end of the article

In addition to data processing requirements, future in-vehicle applications have strict requirements on network bandwidth and task delay. Although the configuration of mobile devices such as computing power and running memory is becoming more and more powerful, it is still insufficient for computationally intensive tasks [5]. This inspired the development of Mobile Cloud Computing (MCC) concept. In MCC, mobile devices upload tasks to the Internet cloud by mobile operators' core networks and use the Internet cloud's powerful computing and storage resources to perform these tasks [6]. In the IoV application scenario, the future demand for delays and other service qualities of in-vehicle applications makes MCC technology not the best choice for IoV scenarios. According to the characteristics of IoV, Mobile Edge Computing (MEC) currently faces many challenges and difficulties in the implementation of IoV technologies [7], which is manifested in the following three aspects:

(1) Complexity of architecture: Due to different communication standards adopted in different regions, multiple communication modes such as DSRC and LTE-V coexist under the existing research architecture. Besides, the complexity of application-driven network architecture construction is increasing with the innovation of various applications. In VANET, Road Side Unit (RSU) serves as a wireless access point in IoV, upload information such as vehicles and traffic conditions to the Internet and publish relevant traffic information. This cooperative communication model of vehicles and infrastructures requires the participation of a large number of roadside nodes, which increases construction cost and energy consumption [1].
(2) Uncertainty of the communication environment: The alarm communication under IoV is extremely susceptible to the impact of surrounding environments, such as the surrounding buildings, the interference of surrounding channels, and the poor network coverage of roadside units [8, 9].
(3) Strict Quality of Service (QoS) requirements: Due to the suddenness of road traffic accidents, information transmission between vehicles needs to have strong timeliness and reliability requirements.

With the continuous improvement of relevant standards and continuous increase of intelligent vehicles, it is foreseeable that more and more vehicles will realize network interconnection by relevant protocols in the future. With the increasing number of vehicles, road hazards have become a problem that must be faced in the development of IoV [10]. Besides, the communication transmission of vehicle safety business has higher timeliness and reliability requirements. In some application scenarios of IoV, such as automatic driving, the delay requirement even needs to be lower than 10 ms. This makes the research on the transmission strategy of IoV security services more and more important [11, 12]. In the vehicle communication process based on IEEE 802.11P and LTE-V protocols, channel congestion, channel interference, shadow fading and intelligent computing processing are main factors that affect the communication performance of vehicles. How to schedule computing resources and communication resources in IoV to improve the communication performance of vehicle safety business has important research value. Besides, the proposed scheduling strategy is based on an IoV system of multi-area multi-user multi-MEC server. A vehicle distance prediction method based on Kalman filtering is proposed combined with the mobility of IoV users in this paper. Furthermore, the total cost of communication delay and energy consumption of all users is formulated as the optimization goal, the Double DQN algorithm is used to solve the optimal scheduling strategy for minimizing the total consumption cost of system.

Related work

In IoV, the low-latency and highly reliable broadcast transmission of alarm communication is fundamental guarantee for traffic safety, and communication protocols are the basis for alarm transmission. Among them, IEEE 802.11p is a communication protocol expanded by IEEE 802.11 standard. It is mainly used for information transmission between vehicles and vehicles, and between vehicles and roadside nodes in a vehicle-mounted self-organizing network [13]. LTE-V is based on LTE-D2D technology, it is designed and modified according to the characteristics of IoV application scenarios to realize wireless communication technology that supports IoV communication.

MEC evolved from MCC to provide IT service environment and cloud computing capability on the wireless access network side close to mobile users. In the MEC environment, users are closer to edge servers, and the transmission delay caused by task offloading is greatly reduced. Moreover, service requesting can be responded at the edge, which can effectively relieve the burden on core networks [14]. In the past 2 years, due to MEC's close range, ultra-low latency, high bandwidth and other characteristics, the research on MEC has become increasingly fierce. In terms of task offloading decision and resource allocation, people have proposed different solutions according to different needs and application scenarios [15]. Since there are too many factors to be considered for task offloading and resource allocation, it is difficult to take all factors into account during modeling. Therefore, existing work simplified the modeling of task offloading. Part of them only studied task offloading to edge servers, and two types of task offloading models were obtained, namely, two-state task offloading and partial task offloading models. Reference [16] aimed to save energy while also considering the allocation of wireless resources, assuming that the computing power of servers is a fixed constant. They performed offloading by classifying different tasks, gave priority to them based on task delay and wireless resource requirements and the weighted sum of energy consumption. The purpose of reference [17] was to minimize the weighted sum of energy consumption and delay. Each user had multiple tasks and they were considered more comprehensively. Reference [18] used game theory to solve the optimization problem and proved the existence of Nash equilibrium. Reference [19] calculated the theoretical upper limit of server-side task processing, and proved that their algorithm can be close to the theoretical value. It transformed non-convex quadratic functions into separable semi-definite programming problems by relaxation techniques under quadratic constraints. Reference [20] proposed a compromise solution where tasks can be processed locally and then offloaded to cloud to execute the remaining part.

However, task delay is only used as a reference condition, it cannot guarantee that the delay of each task can be guaranteed in schemes proposed in the above paper. Reference [21] also considered task offloading and allocation of computing resources. They assumed that wireless bandwidth is a fixed constant, task execution cost is minimized when the strict time constraints of tasks are satisfied. Reference [22] used game theory techniques to allocate the computing power of MEC servers under the premise for the best decision of each user (users' own maximum revenue), which maximizes the operator's revenue. Reference [23] proposed the allocation scheme of wireless channels and computing resources under the condition of satisfying time delay, so as to minimize the energy consumption of users. Reference [2] used Markov decision model to allocate resources, which can ultimately reduce the delay, but cannot guarantee it. Reference [24] minimized energy consumption under the constraints of latency and limited cloud computing resources. However, the guarantee of collaborative reliability for computing resources and the efficiency of task execution is not considered in the heterogeneous wireless network environment. Thus, energy consumption becomes a secondary factor, improving system reliability and task execution efficiency are the most important issues in the IoV scenario.

In order to integrate with actual LTE network, there are also a few work to study the MEC system of heterogeneous wireless networks. Reference [25] proposed a wireless resource allocation scheme under the environment of heterogeneous infinite network, which made the successful execution probability of tasks with strict delay requirements increased by 40%. In reference [26], the interference between macro base stations and micro base stations was reduced, and the wireless rate of multiple users was maximized by periodically suspending macro base stations to transmit signals. Reference [27] proposed a random self-organizing algorithm to solve the problem of wireless resource allocation based on Markov chain and game theory methods. Its purpose was to minimize operating costs. However, when user requesting peak, wireless communication and computing resources cannot avoid the shortage. Reference [28] allocated time slots or sub-channels of wireless channels by time division multiple access and orthogonal frequency division multiple access techniques. Under the condition of satisfying task delay, the energy consumption of mobile users was minimized.

Based on the existing LTE network architecture, a layered MEC network architecture needs to be considered in order to be closer to the actual situation. Taking advantage of the short distance between edge servers and vehicles, a reasonable task offloading decision is made to improve the efficiency of system's task execution according to the amount of data uploaded by computing tasks and computing resources required to perform the task [29, 30]. In this system, vehicles can choose to access either micro base stations or macro base stations. In addition, data centers with different computing capabilities are deployed near two base stations and the Interne, and they consist of servers that provide various functions.

Therefore, from a global perspective, under the premise of strictly meeting application requirements (high reliability), a collaborative scheduling strategy for IoV computing resources based on MEC is proposed to minimize the average task completion time. The main innovations are summarized as follows:

(1) Aiming at the problem of large amount of data and limited local computing capacity of vehicles, a multi region, multi-user and multi MEC server system is designed in this paper, in which one MEC server is deployed in each area, and multiple vehicle user devices in the region can unload the computing tasks to MEC servers in different regions through wireless channel.
(2) The existing scheduling strategies have the problems of high energy consumption and low task completion rate. The proposed scheduling strategy takes the communication delay of all users, the location privacy of vehicles and the total cost of energy consumption as the optimization objectives, which takes into account the system state, action strategy, reward and punishment function, and uses Double DQN algorithm solves the optimal scheduling strategy to minimize the total

consumption cost of the system and complete more computing tasks in the shortest time.

System model and problem modeling

System model

In the system, macro base stations are connected to Internet by the core network in cellular communication system. MEC servers are deployed at macro base stations and micro base stations [30]. It is assumed that micro base stations are connected to macro base stations in a wired manner in this system. Since the interference between macro base stations is small, it is assumed that there is a network architecture of n micro base stations within the coverage of one macro base station, and $n = \{1, 2, \cdots, N\}$ represents a collection of micro base stations. There are i vehicles under the micro base station n, $i = \{1, 2, \cdots, I\}$ represents a collection of vehicles. Only single-antenna vehicles and micro base stations are considered in this system. The system model for multiple base stations and multiple MEC servers is shown in Fig. 1.

It is assumed that each vehicle has a computationally intensive and demanding task that needs to be completed in unit time. Each vehicle can offload the calculation to MEC servers by the micro base station or macro base station connected to it. Each vehicle will upload a task, the tasks uploaded by vehicle i are:

$$T_i = \left\{D_i, C_i, T_i^{\max}\right\} \tag{1}$$

where D_i is the amount of data uploaded by tasks, C_i is the number of CPU cycles required by the server to process tasks, and $T_i^{\max}$ is the maximum time allowed for the task to complete.

During the task offloading process, the vehicle is constantly moving, and the access base station may be switched. This system mainly considers task-intensive and ultra-low-latency task offloading, $T_i^{\max}$ is less than tens of milliseconds. Therefore, it is assumed that no base station handover occurs during task offloading.

Vehicle distance prediction based on Kalman filtering

There are three key random vectors in the whole process of Kalman filtering: the predicted value X_t^p of system state, the measured value X_t^m and the estimated value X_t^c. X_t^p represents the final estimation of t cycle system state by Kalman filtering, which is obtained by data fusion between X_t^m and X_t^c [31]. The prediction process is:

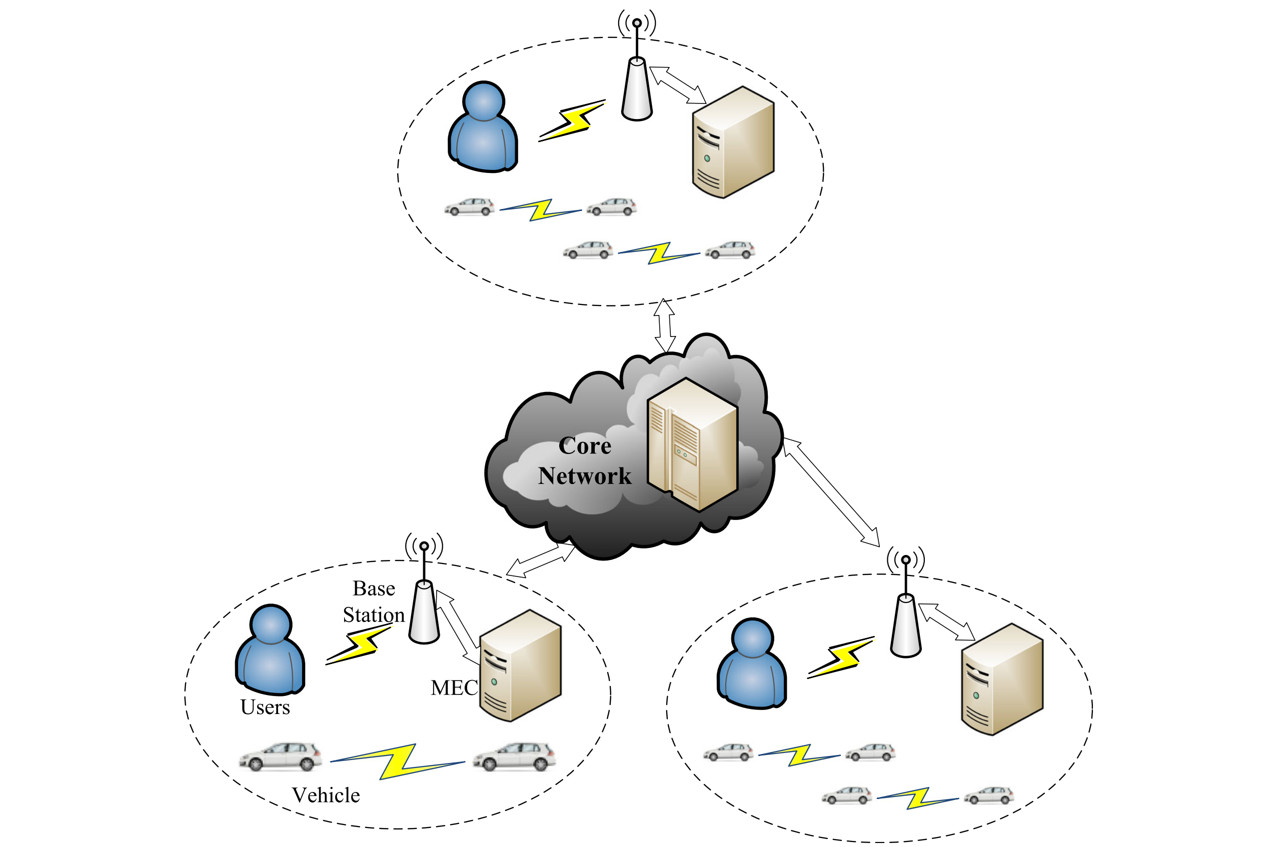

Fig. 1 System model of multi-area multi MEC server

$$\begin{aligned} x_t^p &= F_t x_{t-1}^c + B_t u_{t-1} \\ P_t^p &= F_t P_{t-1}^c F_t^T + Q_{t-1} \end{aligned} \tag{2}$$

where x_t^p is the mean of X_t^p and P_t^p is the covariance matrix of X_t^p. x_t^c is the mean of X_t^c and P_t^c is the covariance matrix of X_t^c. F_t represents the transition matrix of the impact for $t-1$ cycle system state on t cycle system state, and u_{t-1} is the control input matrix. B_t represents the matrix that transforms the influence of the control input to system state, and Q_{t-1} represents the covariance matrix of predicted noise. Here, the prediction noise is assumed to be a Gaussian distribution with zero mean, so it only affects the covariance matrix of this predicted value. Moreover, the prediction noise indicates the accuracy of the prediction model. If the prediction model is more accurate, the prediction noise is smaller.

In an actual system, the object of measurement may not be system states, but some measurement parameters related to it. The measured value of system states can be obtained indirectly by these measurement parameters. Let these measurement parameters be Z_t, and their relationship with the measured values is:

$$Z_t = H_t x_t^m + s_t \tag{3}$$

where Z_t represents the matrix that maps system states to the measurement parameters. s_t represents measurement noise, subjects to a Gaussian distribution with mean zero and covariance matrix R_t.

The process of Kalman filtering is shown in Fig. 2. The left half of this figure indicates that when the system is in period t, the system state of period t + 1 is predicted. The right half of this figure shows that after the t + 1 period, the measured value of t + 1 period is obtained. Thus, the estimated value of t + 1 period is calculated as the input for the next round of prediction. It is applied to vehicle distance prediction.

The system state is the location information of vehicle i (vehicle i, v_i). Since the width of road is negligible relative to the length, the vehicle position is modeled as a one-dimensional coordinate. In order to make the prediction model more accurate, speed is also added to the system state. Thus, the mean value $X_{i,t}^c$ of the estimated value $x_{i,t}^c$ of v_i in t period is shown in Eq. 3–7, and the predicted and measured values are the same.

$$x_{i,t}^c = \begin{bmatrix} loc_{i,t}^c \\ velocity_{i,t}^c \end{bmatrix} \tag{4}$$

Use uniformly accelerated linear motion to predict this system, and set the period interval to T. The acceleration of v_i is $a_{i,\ t}$, then:

$$F_t \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, B_t = \begin{bmatrix} \frac{(\Delta t)^2}{2} \\ \Delta t \end{bmatrix}, u_t = a_{i,t} \tag{5}$$

When directly measuring the position and speed, $X_{i,t}^m = Z_{i,t}$, that is:

$$H_t = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, Z_{i,t} = X_{i,t}^m, R_{i,t} = P_{i,t}^m \tag{6}$$

where $Z_{i,\ t}$ is the measurement parameter, H_t represents the matrix that maps system states to measurement parameters, and $R_{i,\ t}$ is the covariance matrix of measurement noises.

Substituting eqs. (4)–(6) into eqs. (2) (3), Kalman filtering can be applied to vehicle position prediction. Since system states are a two-dimensional Gaussian distribution composed of position and velocity, it is easy to obtain a one-dimensional Gaussian distribution in various dimensions. Let $LOC_{i,t}^c$ be the estimated value of position for v_i in t period. Similarly, $LOC_{i,t}^p$ is the

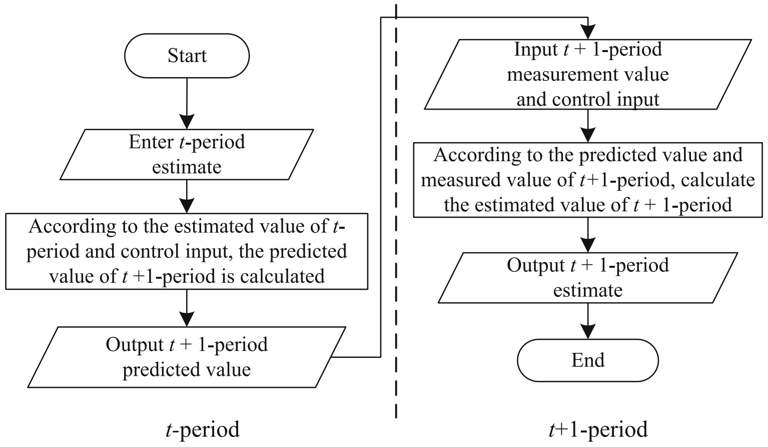
Start
Enter *t*-period estimate
According to the estimated value of *t*-period and control input, the predicted value of *t* +1-period is calculated
Output *t* + 1-period predicted value
t-period
Input *t* + 1-period measurement value and control input
According to the predicted value and measured value of *t*+1-period, calculate the estimated value of *t* + 1-period
Output *t* + 1-period estimate
End
t+1-period

Fig. 2 Kalman filtering process

predicted value and $LOC_{i,t}^{m}$ is the measured value. They all obey one-dimensional Gaussian distribution, namely:

$$LOC_{i,t}^{e} \sim N\left(\mu_{i,t}^{e}, \left(\mu_{i,t}^{e}\right)^{2}\right), LOC_{i,t}^{p} \sim N\left(\mu_{i,t}^{p}, \left(\mu_{i,t}^{p}\right)^{2}\right),$$
$$LOC_{i,t}^{c} \sim N\left(\mu_{i,t}^{c}, \left(\mu_{i,t}^{c}\right)^{2}\right) \tag{7}$$

For two vehicles v_i and v_j, at the t cycle, random variable $D_{i,\ j,\ t}$ between them can be obtained by subtracting the position random variables $LOC_{i,\ t}$ and $LOC_{j,\ t}$:

$$D_{i,j,t} = LOC_{i,t} - LOC_{j,t} \tag{8}$$

A random variable representing the distance between two vehicles can be obtained by the above formula. At the same time, $D_{i,\ j,\ t}$ follows a one-dimensional Gaussian distribution, such as:

$$D_{i,j,t} = N\left(\mu_{i,t} - \mu_{j,t}, \left(\sigma_{i,t}\right)^{2} + \left(\sigma_{j,t}\right)^{2}\right) \tag{9}$$

Compared to random variables, Vehicle to Vehicle (V2V) computing offloading and V2V communication resource algorithms hope to obtain an exact value directly representing the distance between two vehicles. In this way, V2V computing offloading and V2V communication resource allocation algorithms can completely ignore the mobility and focus on the problem itself to achieve decoupling of complex problems [32].

Participation in vehicle location privacy protection mechanism

Note that the probability of disturbance from real position l_i^r to position l_j^o of the participant is $p(l_j^o|l_i^r)$, so for all positions of the participant, the probability matrix of disturbance can be obtained as P and $P = \{p_{i,\ j}\}_{L \times m}$, which is expressed as follows

$$\mathbf{P} = \begin{bmatrix} p\left(l_1^o|l_1^r\right) & p\left(l_1^o|l_2^r\right) & \cdots & p\left(l_1^o|l_m^r\right) \\ p\left(l_2^o|l_1^r\right) & p\left(l_2^o|l_2^r\right) & \cdots & p\left(l_2^o|l_m^r\right) \\ \vdots & \vdots & \vdots & \vdots \\ p\left(l_L^o|l_1^r\right) & p\left(l_L^o|l_2^r\right) & \cdots & p\left(l_L^o|l_m^r\right) \end{bmatrix}_{L \times m} \tag{10}$$

Therefore, $p_{i,j} = p(l_j^o|l_i^r)$ can also be understood as the conditional probability of l_i^r disturbance to l_j^o in the real position. Next, based on the differential privacy, the location indistinguishability disturbance mechanism is proposed.

The probability perturbation mechanism P satisfies the position indistinguishability if and only if it satisfies the following inequality

$$p\left(l_j^o|l_{i_1}^r\right) \le e^{ed\left(l_{i_1}^r, l_{i_2}^r\right)} p\left(l_j^o|l_{i_2}^r\right) \tag{11}$$

Where $l_{i_1}^r$ and $l_{i_1}^r$ belong to set l^R. As the differential privacy budget e represents the degree of privacy protection, generally speaking, the smaller e is, the higher the degree of privacy protection is, the more difficult it is for $l_{i_1}^r$ and $l_{i_1}^r$ to distinguish; on the contrary, it means the degree of privacy protection is low, and the distinction between the two real locations is high. The function $d(l_{i_1}^r, l_{i_2}^r)$ represents the distance between position $l_{i_1}^r$ and position $l_{i_2}^r$, which can be Euclidean distance or Hamming distance. The distance function adopted in this chapter is Euclidean distance. In fact, it can be seen from formula (11) that when the appropriate differential privacy budget e is selected, if two positions are selected. The smaller the distance between $l_{i_1}^r$ and $l_{i_2}^r$, that is, the closer the two positions are, the smaller the probability of generating disturbance position l_j^o from these two positions is. In other words, in this case, the attacker can't exactly distinguish the real location of the participant or the location near the participant.

Because the participant only publishes the disturbed location, the attacker can observe the disturbed location of the participant, but can't get its real location directly. In this chapter, we consider that the attacker has background knowledge, that is, the attacker can obtain disturbance mechanism P and probability $p(l_i^r)$, then the attacker can use Bayesian theorem to deduce the observed disturbance location to get its real location. Probability $p(l_i^r|l_j^o)$ represents the probability that the real location of the participant is in l_i^r under the premise of disturbance location. From Bayes theorem and total probability formula, we can get:

$$p\left(l_i^r|l_j^o\right) = \frac{p(l_i^r)p\left(l_j^o|l_i^r\right)}{p\left(l_j^o\right)} = \frac{p(l_i^r)p\left(l_j^o|l_i^r\right)}{\sum_{i=1}^{m} p\left(l_j^o|l_i^r\right)p(l_i^r)} \tag{12}$$

From the above formula, it can be seen that since the disturbance mechanism P (i.e. the probability from the real location p_i^r to the disturbed location p_j^o) can be obtained by the attacker, and the probability $p(l_i^r)$ of the real location can also be obtained (the attacker can get the posterior probability $p(l_i^r|l_j^o)$ by using the Markov model through the public data set). And $p(l_i^r|l_j^o)$ is bounded. Therefore, the disturbance probability matrix satisfying formula (11) can realize the indistinguishability of participants' location, overcome the attackers with prior knowledge, and protect the participants' location privacy.

System model analysis

In the analysis of vehicle edge computing, it is assumed that the edge network node base station serves as the dispatch control center. The vehicle user equipment is the computing task generator, and vehicles and base stations are the computing task offloading processors, as shown in Fig. 3. When a computing task is generated on the vehicle equipment side, scheduling requesting will first reach the edge network node base station. The task will be scheduled by base stations, and the scheduling algorithm decides to schedule computing tasks to a service queue on the base station node side or a service queue for vehicles [33]. Once the computing task enters a queue, it will queue up at the end of this queue. At the same time, it is assumed that vehicle users have a total of M different computing tasks. For each computing task m, there is a fixed communication workload f_m, a fixed computing workload d_m and a fixed task time constraint T_m. The computing task volume can be expressed by the number of revolutions of the CPU.

Vehicles perform periodic state interactions, and information such as location, driving direction, speed and idle computing power of neighboring vehicles can be obtained by the communication network. When the vehicle equipment generates a computing task, it initiates the computing offloading request information to edge nodes. The request information includes explanation information about computing tasks. The explanation information includes: the communication task size f_m of computing tasks, the computing task size d_m, the delay requirement T_m, and the idle computing capacity of the neighboring vehicle.

It is also assumed that vehicles on the road are traveling at a constant speed at a fixed speed. In the analysis of vehicle communication mechanism in the previous two chapters, it can be seen that there is a communication link between edge node base stations and vehicles. Information such as the vehicle's computing power, location, driving direction and speed can be periodically interacted with base stations by CAM messages. The system scheduling decision is $b_k^t \in (b_1, b_2, \cdots, b_m, b_{m+1})$, where b_k^t indicates that the computing task that arrives at time t is placed in the corresponding computing processing queue k [9]. Therefore, when the computing request of vehicle users arrives, how to allocate computing tasks to the corresponding calculation service queue, and thus ensure the delay requirement of the long message security service, which allows the system to have the greatest alarm revenue.

In the analysis of our designed computing task scheduling model, the scheduling process is regarded as a Markov decision process [34]. When the base station receives computing offloading requests sent by the user equipment of vehicles, base stations calculate queue status according to the calculation. The state of the available computing processing queue of vehicles and the information of the computing task combined with Markov decision model determine a certain computing processing queue as the offloading queue of computing tasks. The definition of system states at time t is as follows:

$$S^t = \left(q_1^t, q_2^t, \cdots, q_m^t, q_{m+1}^t, v_{m+1}^t, d^t, f^t\right) \tag{13}$$

where $q_1^t, q_2^t, \cdots, q_m^t$ is the queue length (computing task size) of m computing processing queues at edge nodes at time t. q_{m+1}^t is the length of vehicles' calculation processing queue, and d^t is the amount of computing task generated by users at time t. f^t is the size of communication task generated by users at time t. The value of v_{m+1}^t is the idle computing capacity of vehicles generating the emerging alarm service and its neighboring auxiliary vehicles.

The system state at time t is $(q_1^t, q_2^t, \cdots, q_m^t, q_{m+1}^t, v_{m+1}^t, d^t, f^t)$, and the scheduling decision is $b_k^t \in (b_1, b_2, \cdots, b_m, b_{m+1})$. The actual processing capacity of each computing

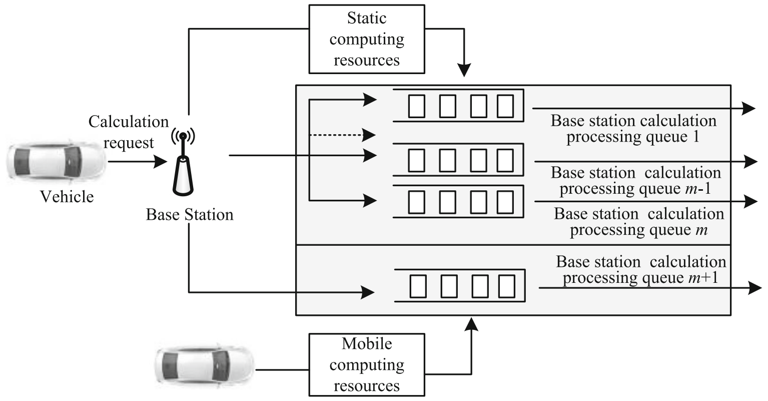
Static computing resources
Calculation request
Vehicle
Base Station
Base station calculation processing queue 1
Base station calculation processing queue m-1
Base station calculation processing queue m
Base station calculation processing queue m+1
Mobile computing resources

Fig. 3 Analysis of vehicle edge computing

processing queue within time interval τ is shown in the following formula:

$$\hat{S}_k^t = \min\left(q_k^t + P_k^t \times d^t, v_k \times \tau\right) \tag{14}$$

In the formula, when the scheduling probability P_k^t is 1, it means that computing task d^t that arrives at time t is scheduled to the computing task processing queue k. When P_k^t is 0, it means that the computing task that arrives at time t is not scheduled to the computing task processing queue k. Therefore, the system state at $t + 1$ can be derived as shown in the following formula:

$$\begin{aligned} S^{t+1} = & \left(q_1^t + P_1^t \cdot d^t - \hat{S}_1^t, \cdots, q_m^t + P_m^t \cdot d^t - \hat{S}_m^t,\right. \\ & \left. q_{m+1}^t + P_{m+1}^t \cdot d^t - \hat{S}_{m+1}^t, v_{m+1}^{t+1}, d^{t+1}, f^{t+1}\right) \end{aligned} \tag{15}$$

In addition, the impact of communication resource allocation on computing resource scheduling needs to be considered. If the scheduling behavior b_k^t schedules the computing task of the vehicle safety application to vehicle nodes, then tasks will be coordinated by neighboring vehicles to participate in the calculation process, and the processing delay is as follows:

$$T_b^t = \frac{d_m^t + q_{m+1}^t}{v_{m+1}} \tag{16}$$

If the scheduling behavior b_k^t schedules computing tasks of the vehicle safety application to base stations, then the completion delay T_b^t of task m due to scheduling is:

$$T_b^t = \frac{d_m^t + q_k^t}{v_k} + \frac{f_m^t}{C} \tag{17}$$

where the uplink communication rate between user equipment of vehicle C and the edge node base station.

At this point, the return r_t from the state transition from S_t to S_{t+1} caused by behavior decision b_k^t can be analyzed as:

$$r_t = r\left(s^t, b^t, s^{t+1}\right) = \sum_{k=0}^{m+1} \left(\frac{\hat{S}_k^t}{V_k} \cdot \zeta_k\right) - \alpha\left(q_k^{t+1}\right)^2 - \beta F_2\left(T_b^t - T_m\right) \tag{18}$$

The first item about r_t is the total alarm revenue from computing resources provided by each service queue within a time interval. The second term is to punish the square of queue length in order to avoid a serious imbalance in the length of service queue. The last item is the punishment of whether tasks are completed within time delay requirement to improve the alarm performance. In order to obtain better performance in the long term, computing resource providers must consider not only the return at the current moment, but also the future return to be obtained. The ultimate goal is to learn an optimal scheduling strategy to maximize the cumulative discount reward, as shown in the following formula:

$$\pi^* = \arg \max_{\pi} E\left[\sum_{t=0}^{\infty} \left(\eta^t \cdot r_t\right)\right] \tag{19}$$

where $\eta (0 \leq \eta \leq 1)$ is the discount factor. When t is large enough, η^t tends to 0, which means that r_t has a small effect on the total return. The ultimate goal is to learn an optimal scheduling strategy π^* to maximize system revenue.

Resource allocation based on deep reinforcement learning

Deep reinforcement learning theory

Reinforcement learning is a major branch of machine learning, and its essence is the problem of choosing the optimal decision to obtain decision rewards. Reinforcement learning is mainly composed of four units: Agent, Environment, action and reward. The goal of reinforcement learning is how to act based on the environment to maximize the expected return. As shown in Fig. 4, Agent is an intelligent learning unit. It interacts with the environment, obtains state from this environment, trains by neural network and decides the behavior strategy it wants to make. Every behavior decision will bring a certain logical return according to the corresponding logic. Similarly, each action may update system state at the previous moment.

Markov decision is an important mathematical application model in reinforcement learning. The Markov property is satisfied between state transitions, that is, each state transition depends only on the previous finite state. And give a certain reward for the state transition brought by each step of action. It is mainly used in the mathematical analysis of learning strategies, emphasizing the intelligent learning unit according to a specific state s. Choose the corresponding behavior strategy action to get the desired return r. At the same time, an optimal strategy problem in the Markov decision process is a reinforcement learning problem [35]. In the mathematical analysis, first introduce R_t to represent the overall discounted return from a certain moment t to the future and:

$$R_t = r_{t+1} + \eta r_{t+2} + \cdots = \sum_{k=0}^{\infty} \eta^k r_{t+k+1} \tag{20}$$

The value function is defined as the expected return, and the mathematical formula can be expressed by Bellman equation:

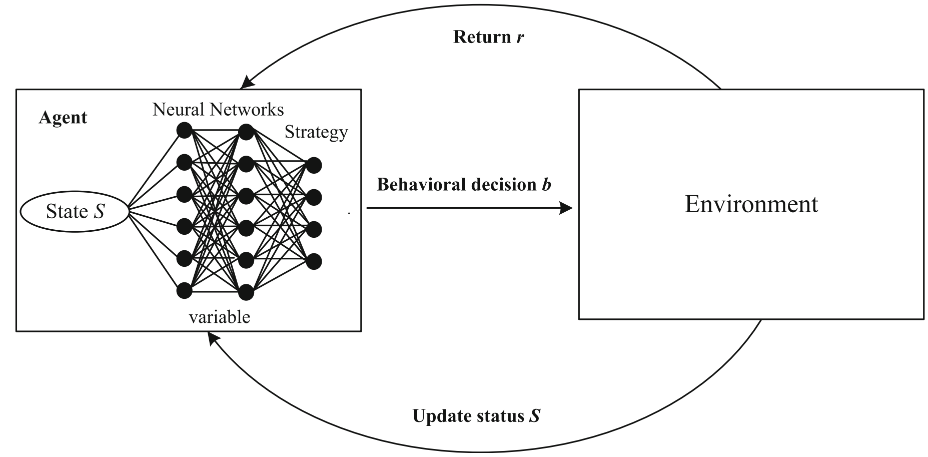

Fig. 4 Interaction block diagram of reinforcement learning

$$v(s) = E[R_t|S_t = s] = E[r_{t+1} + \eta v(s_{t+1})|S_t = s] \quad (21)$$

The Bellman equation shows that the value function can be calculated by iteration. The traditional iterative algorithm of Markov decision process is divided into two types: value iteration method and strategy iteration method. Both of these methods are algorithms updated using Bellman equation. The convergence of value iteration is because Bellman optimal equation has the nature of contraction mapping. We can know iterative convergence of values by Banach Fixed Point Theory. The reason for policy iteration convergence is Monotone Boundary Convergence Theory. After updating this strategy, the cumulative return becomes larger and the monotonic sequence formed will always converge.

In order to better characterize the maximum value of current reward including the future, we use the action value function $Q^{\pi}(s, a)$ to describe this iterative process:

$$\begin{aligned} Q^{\pi}(s, b) &= E\left[r_{t+1} + \eta r_{t+2} + \eta^2 r_{t+3} + \cdots |s, b\right] \\ &= E_s\left[r + \eta Q^{\pi}\left(s^{'}, b^{'}\right)|s, b\right] \end{aligned} \quad (22)$$

where s and b represent state and action behavior respectively, r_t represents the return value at time t.

DQN uses ε-greedy strategy, that is, with a certain probability ε, it randomly selects behaviors to explore changes in the environment to avoid local optimization. The behavior selected at the other moments is based on the following formula.

$$b^t = \arg\max_b Q(s^t, b; \theta) \quad (23)$$

Since Q-table is applicable to the space with limited state, DQN uses neural network to predict Q value by continuous learning and training to update the parameters of neural network. In DQN, the approximation of neural network, which is a nonlinear value function, is expressed as follows:

$$Q(s, b; \omega) \approx Q^*(s, b) \quad (24)$$

where ω is the weight of neural network, and parameter ω needs to be updated to make Q function approximate the optimal Q value.

Resource allocation based on double DQN

To solve the problem of limited capacity and high latency of a single MEC server, we consider the IoV scenario of multiple MEC servers in multiple cells. Compared with traditional static user scenario, this dynamic scenario makes the problem more complicated. Thus, the problem is mixed integer nonlinear programming. Traditional optimization methods or heuristic algorithms can only obtain suboptimal solutions to problems [36]. Considering users' mobility, dynamic scenarios and models with more complex problems, this paper proposes deep reinforcement learning to solve this problem. This method adopts a centralized control distribution method, using the controller of multiple MEC servers located in core network as an agent, and coordinating the MEC servers of all cells by controller. Since reinforcement learning is model-free, we first need to model this problem based on three factors other than state transition probabilities.

(1) State: The state of each time slot is set to the computing power that each MEC server has at the beginning of this time slot, that is, the remaining computing power of MEC servers. Since the sizes of tasks performed by MEC servers are different, the computing power allocated to each task is also different. This results in the remaining computing power of MEC servers being different after a time

slot ends. In addition, the computing power of each time slot for MEC servers is only related to the remaining computing power of the previous time slot and the computing power for the completion calculation of the last time slot. This state change satisfies Markov property, and state space S is defined as follows:

$$S(t) = \{s_1(t), \cdots s_i(t) \cdots s_M(t)\} \tag{25}$$

where $S(t)$ represents state space of the t time slot. $s_i(t)$ represents computing power of the i MEC server at the beginning of the t time slot.

(2) Action: Since the core of Double DQN is still Q learning algorithm, the action is discretized in order to avoid the continuous action causing action space to be too large. According to modeling problem, the variables to be optimized mainly include the offloading decision of user task in IoV, user's transmission power and the computing power allocated by MEC to users [37]. It should be noted that the multi-cell scenario. For IoV user tasks, there are three calculation modes: local calculation, offload calculation to MEC server of cells, and offload calculation to MEC server of other cells in the vicinity. In order to show the distribution scheme of agent more intuitively, define the motion vector as Ψ:

$$\Psi = \{X, f_1, \cdots, f_i, \cdots, f_N, p_1, \cdots, p_N\} \tag{26}$$

where X is offloading decision vector of user tasks, which describes the offloading decision of user tasks. f_i represents the computing power allocated by MEC server for the i user. p_i represents the transmission power of the i IoV user.

(3) Reward: The agent expresses its satisfaction with the action by expected value of reward in a period of time. Combining objective function C_{sum} of this problem, the objective of original problem is to minimize the cost function. The goal of reinforcement learning is to maximize immediate rewards. Considering that the immediate reward and cost function appear to be negatively correlated, the immediate reward function is defined as follows:

$$R(s,b) = \frac{C_{local} - C_{sum}(s,b)}{C_{local}} \tag{27}$$

where $R(s,b)$ represents the immediate reward of selecting action b in state s. C_{local} represents the cost of all tasks calculated locally, which can be understood as the upper limit of cost function. $C_{sum}(s,b)$ represents the cost consumption of performing action b when the current time slot is in state s.

By the immediate reward function, the long-term cumulative discount reward value $Q_\pi(s,b)$ of original problem can be expressed as follows:

$$Q_\pi(s,b) = E_\pi\left[\sum_{t=1}^{T} \beta^{t-1} R(t)\right] = E_\pi\left[\sum_{t=1}^{T} \beta^{t-1} \frac{C_{local} - C_{sum}(s,b)}{C_{local}}\right] \tag{28}$$

For MEC controllers, the purpose of learning is to find strategies to maximize long-term cumulative rewards:

$$\pi^* \text{ arg } \max_{b \in B} Q_\pi(s,b) \tag{29}$$

Although the action has been discretized, the action strategy assigned by centralized controller covers all action possibilities. This may include a variety of non-existent situations, for example: the computing power allocated by a certain MEC server in this time slot is greater than its remaining computing power in this time slot. This paper screens the action space after the action space is constructed, screen out the impossible situations and eliminate them, further reducing action space, speeding up the training speed, and reducing training delay. In addition, due to the mobility of IoV users, it may cause changes in cells where the user is located. And with the increase of users, the action space will increase exponentially. Therefore, a certain pre-processing is performed on this situation, that is, if the delay is calculated locally, then the local computing task will be preferred for IoV users. Otherwise, it will choose to offload to MEC servers for calculation. The increase of action space can be well controlled by a series of preprocessing of data. The process of task offloading and resource allocation algorithm based on Double DQN is shown in Fig. 5.

Experimental results and analysis

Generally speaking, the density of vehicles in cities is 1000–3000 vehicles/km^2; the density of suburban vehicles is 500–1000 vehicles/km^2; the density of highway vehicles is 100–500 vehicles/km^2. In different scenarios, the coverage of base stations will be different. Taking the urban environment as an example for simulation and selecting a square area of 500 m × 500 m, the number of vehicles in this area is probably between 250 and 750.

The centralized controller can schedule all base stations and MEC servers, and the proposed IoV computing cooperative scheduling strategy is deployed in the centralized controller. The actual scenario uses offline

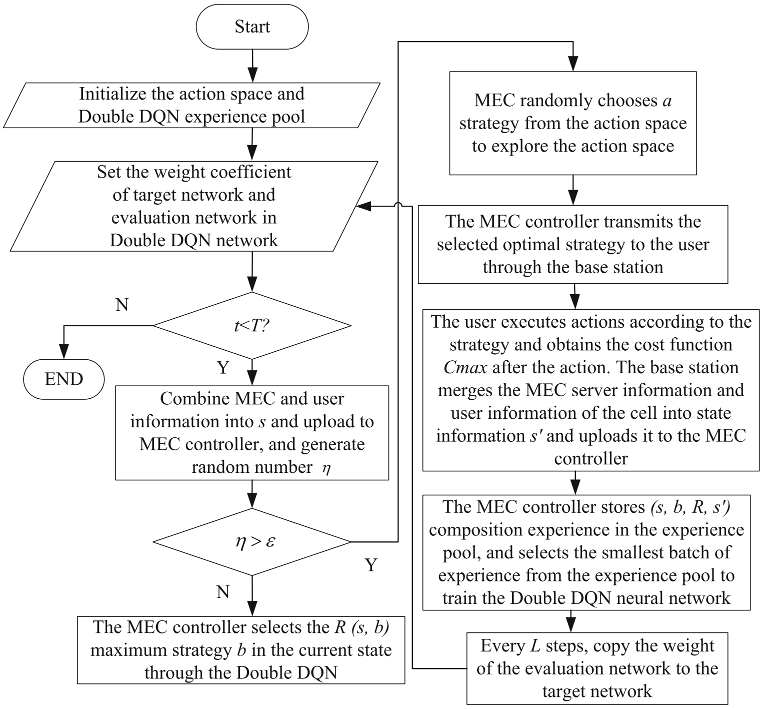
Start
Initialize the action space and Double DQN experience pool
Set the weight coefficient of target network and evaluation network in Double DQN network
N
t<T?
END
Y
Combine MEC and user information into s and upload to MEC controller, and generate random number η
η > ε
Y
N
The MEC controller selects the R (s, b) maximum strategy b in the current state through the Double DQN
MEC randomly chooses a strategy from the action space to explore the action space
The MEC controller transmits the selected optimal strategy to the user through the base station
The user executes actions according to the strategy and obtains the cost function Cmax after the action. The base station merges the MEC server information and user information of the cell into state information s' and uploads it to the MEC controller
The MEC controller stores (s, b, R, s') composition experience in the experience pool, and selects the smallest batch of experience from the experience pool to train the Double DQN neural network
Every L steps, copy the weight of the evaluation network to the target network

Fig. 5 Collaborative scheduling strategy flow of computing resources based on Double DQN

training and online resource scheduling. The specific simulation parameters are shown in Table 1.

Iterative analysis

In order to demonstrate the convergence of proposed method, it is compared with the algorithms in reference [20], reference [25] and reference [27]. The results are shown in Fig. 6.

It can be seen from this figure above that after 500 iterations, the cost function of proposed algorithm gradually converges to about 1.5. And there is a certain fluctuation in image convergence, the main reason is that the amount of task data for each user is different. The remaining computing power of MEC servers is different for each time slot, so there will be some fluctuations in the calculation cost function. In addition, the cost functions of reference [20], reference [25] and reference [27] converge to approximately 3.8. It can be seen from this that the cost function of proposed algorithm is superior to other algorithms. This is mainly due to Double DQN algorithm solves the problem of overestimation by improving the loss function.

Parameter discussion

Impact of greedy factors on the performance of proposed strategy

Under different vehicle densities, the influence of greedy factors on the performance of this algorithm is evaluated in terms of average calculation queue length and computing task completion.

As can be seen from the above figure, as the queue length greedy factor value increases, the average queue length value will decrease under different vehicle densities. As the vehicle density value increases, the average calculation queue length will tend to stabilize. And when ε value becomes larger, the computing task completion rate will increase slightly under different vehicle densities. Especially when vehicle density is below 70, the computing task completion rate increases with vehicle density is not obvious. The reason is that vehicle density

Table 1 Simulation parameters

System simulation parameter	Setting value	System simulation parameter	Setting value
local computing power	1GHz	reward discount	0.09
Maximum capacity of MEC server	5GHz	task packet size	[500,800] kbits
ε greedy factor	0.99	time interval	10 ms
Optimizer	Adam	activation function	ReLu

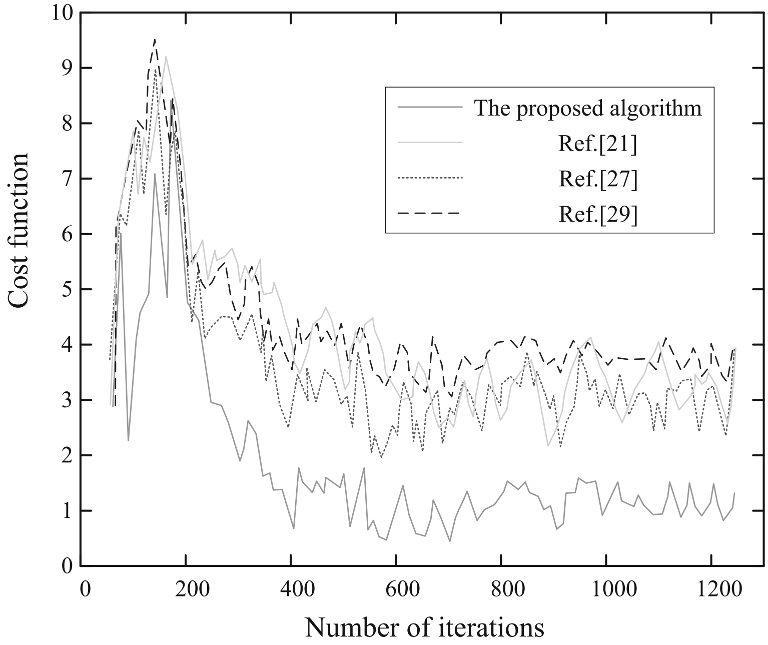

Fig. 6 Convergence of different algorithms

only affects the calculation processing capacity in vehicle calculation queue. When the size of vehicle computing power grows to a certain extent, the upper limit of system's revenue and the completion rate of computing tasks is limited by the processing rate of base station's computing processing queue (Fig. 7).

Impact of MEC capacity on the performance of proposed strategy

For comparison and description, three offloading schemes are introduced, namely, user tasks are only calculated locally, user tasks are only offloaded within the base station (local and local base station MEC server), and user tasks are randomly offloaded (local, local base station MEC server and nearby base station MEC server). In order to simplify description, all local, local base station offloading and random offloading are used. Among them, the impact of the maximum computing capacity of MEC servers on the cost function is shown in Fig. 8.

It can be seen from the figure that this solution where all tasks are calculated locally does not involve MEC servers. Therefore, the cost function remains unchanged. Random offloading schemes have large fluctuations in costs due to random resource allocation. The cost of proposed resource allocation scheme and tasks based on Double DQN is only offloaded within base stations, and the cost of scheme gradually decreases as MEC server capacity increases. When the computing power of MEC servers is 4GHz/s, the resource allocation scheme based on Double DQN is about 15% lower than the cost of offloading scheme at this base station only. However, as the capacity of MEC serves increases, the gap between proposed solutions and tasks only offloaded within base stations gradually decreases. The main reason is that the resources of MEC servers are sufficient to meet the task requirements of current base station.

Comparison of algorithm revenue results

In order to demonstrate the results of our proposed algorithm for computing resource collaborative scheduling, this paper compares it with the algorithms in reference [20], reference [25] and reference [27] for revenue. The results are shown in Fig. 9.

It can be drawn from the figure above that the performance of proposed algorithm at the beginning of initial iteration is relatively close to that of reference [27]. But the performance of these two decisions is better than the scheduling strategy in reference [25] and reference [20]. The proposed algorithm and reference [25], reference [27] system revenue are gradually increased. But the increase rate of proposed algorithm is greater than other algorithms. As proposed scheduling strategy uses Double DQN algorithm, it can find the optimal scheduling scheme in a short time, which can reduce system energy overhead and increase system revenue. Reference [20] can be drawn from this figure, the system revenue may decrease with the increase for number of iterations. This is because the actions taken by random decision may prevent computing tasks from being completed in a timely manner, so that the length of waiting for computing tasks in each calculation queue is long. And it can be seen that after 1000 iterations, the

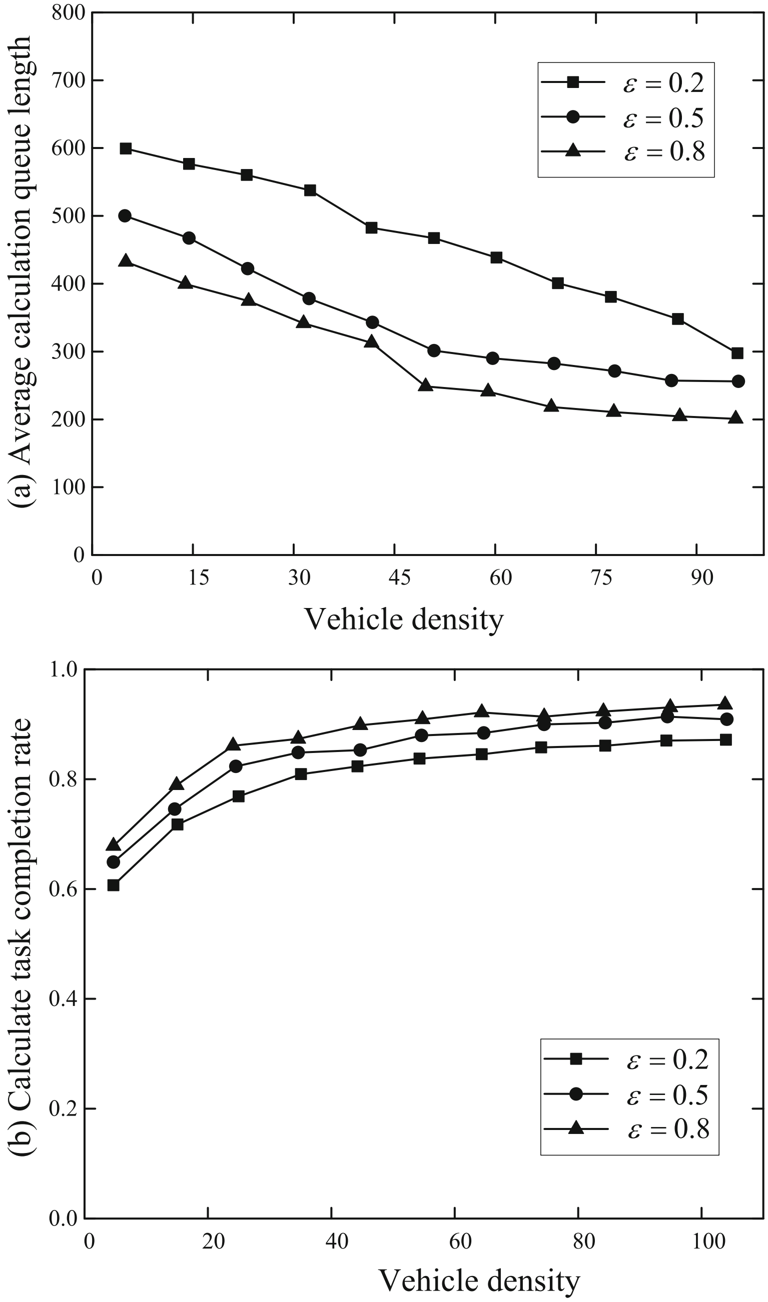

Fig. 7 Comparison of the results for the proposed scheduling strategy under different greedy factors

overall performance of proposed algorithm is better than other algorithms. After 8000 iterations of training, the performance of proposed algorithm tends to converge and stabilize.

As the number of iterations of different algorithms increases, the comparison of their computing task completion rates is shown in Fig. 10.

In the previous analysis, it was specified that IoV user equipment randomly generates computing tasks, and each computing task has a delay requirement. If the computing task is completed within prescribed time, the task is successfully completed. Otherwise, the task is not completed in time and system revenue will be punished accordingly. It can be seen from the figure that proposed algorithm has the highest computing task completion rate and converges to about 80% after 8000 iterations. The performance of reference [27] is second, converging to about 60% after 8000 iterations. Reference [20] has the worst performance using a random strategy and eventually converges to around 50%.

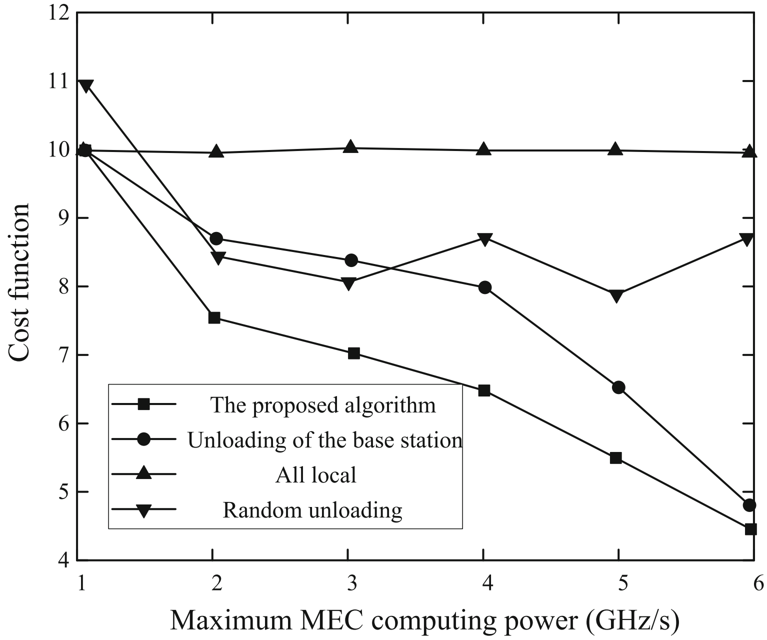

Fig. 8 Impact of MEC capacity on the performance of proposed strategy

Under different vehicle densities, the system benefits of different algorithms are shown in Fig. 11.

It can be seen from the figure that as vehicle density increases, the system revenue under different algorithms will increase. This is because the calculation processing capacity in vehicle calculation queue will increase as vehicle density increases, thereby improving system revenue. But system revenue will not increase with the increase of vehicle density. Since when the computing power of vehicle calculation processing queue increases to a certain degree, it will always complete the assigned computing tasks in time. There will be no more obvious improvement in system revenue. The proposed algorithm will not have a more obvious increase in revenue after vehicle density is around 75. In reference [27], the vehicle density will not increase significantly around 62.

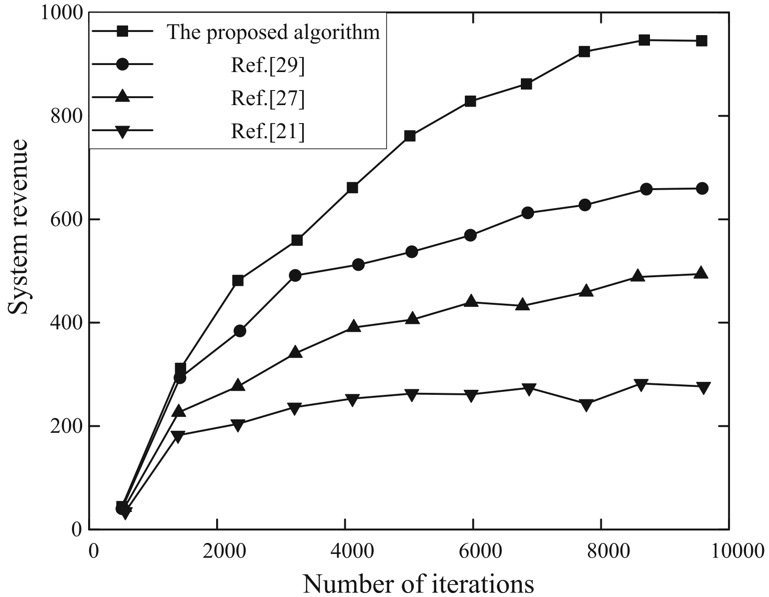

Fig. 9 Comparison for the revenue results of different algorithms

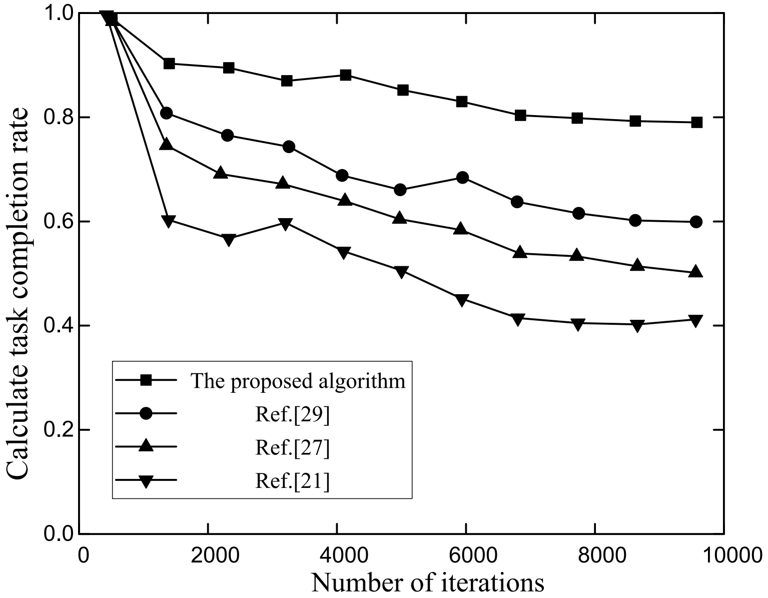

Fig. 10 Comparison of task completion rate of different algorithms

From the performance of system revenue as vehicle density changes, the performance of our proposed algorithm is superior to other comparison algorithms.

Conclusion

The application scenarios of IoV in the future are high-bandwidth, low-latency and high-reliability, MEC technologies can well meet the needs of such scenarios. Therefore, the proposed collaborative scheduling strategy of computing resources for IoV based on MEC can solve the problems of user task offloading decision and wireless and computing resource allocation, so that all tasks are completed as soon as possible, which improves the task execution efficiency of system. Besides, the proposed scheduling strategy is based on an IoV system of multi-area multi-user multi-MEC server. Simulation experiments show the convergence of our proposed algorithm, our proposed algorithm has the smallest system energy overhead compared with other algorithms. It completes tasks with the least number of iterations, which

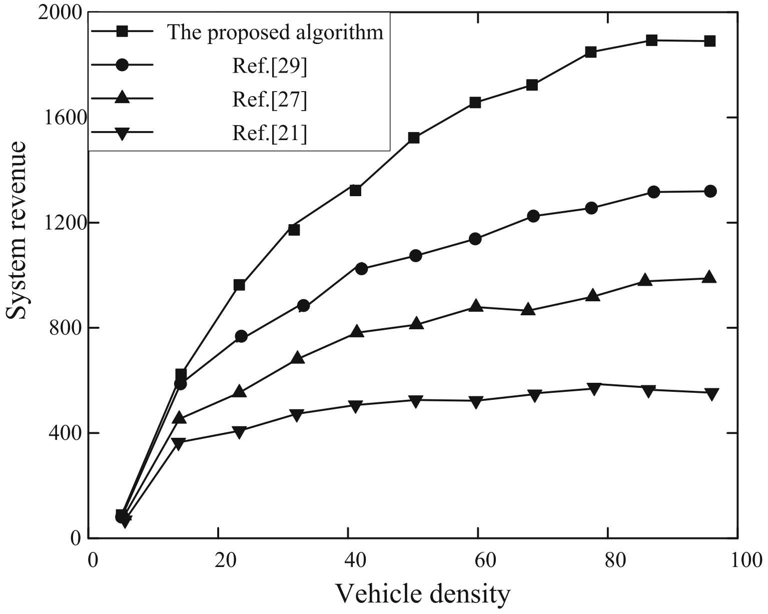

Fig. 11 Comparison for the revenue results of different algorithms under different vehicle densities

demonstrates the effectiveness of our proposed scheduling strategy.

When task offloading is performed in the proposed scheduling strategy, only wireless resources related to task transmission and computing resource allocation related to task calculation are considered. More resources are involved in offloading actual tasks, such as the backhaul network cable channel, data center memory and cache resources.

Acknowledgements
Not applicable

Authors' contributions
The main idea of this paper is proposed by Meiyu Pang. The algorithm design and experimental environment construction are jointly completed by Meiyu Pang and Li Wang. The experimental verification was completed by all the three authors. The writing of the article is jointly completed by Meiyu Pang and Li Wang. And the writing guidance, English polish and funding project are completed by Ningsheng Fang. The author(s) read and approved the final manuscript.

Authors' information
Meiyu Pang, Master of Computer Science, Lecturer. Graduated from Taiyuan University of Technology in 2006. Worked in Wuxi Taihu University. His research interests include computer application and artificial intelligence.
Li Wang, Master of Computer Science, Lecturer. Graduated from Jiangsu University of Science and Technology in 2015. Worked in Wuxi Taihu University. His research interests include image processing.
Ningsheng Fang, Master of Computer Science, Associate Professor. Graduated from Dongnan University in 1999. Worked in Dongnan University. His research interests include computer application and software engineering.

Funding
This work was supported by the Natural Science Foundation of China (No. 61873112).

Availability of data and materials
All the data listed in this article can be shared.

Competing interests
The authors of this article declare that they have no competing interests.

Author details
[1]School of Internet of Things Engineering, Wuxi Taihu University, Wuxi 214064, Jiangsu, China. [2]Jiangsu Key Construction Laboratory of IoT Application Technology, Wuxi 214064, Jiangsu, China. [3]School of Computer Science and Engineering, Southeast University, Nanjing 211189, Jiangsu, China.

Received: 13 July 2020 Accepted: 10 September 2020
Published online: 22 September 2020

References

1. Xu W, Zhou H, Cheng N et al (2018) Internet of Vehicles in Big Data Era. IEEE/CAA Journal of Automatica Sinica 5(1):19–35.
2. Wang J, Jiang C, Han Z et al (2018) Internet of Vehicles: Sensing-Aided Transportation Information Collection and Diffusion. IEEE Transactions on Vehicular Technology 67(5):3813–3825.
3. Li P, Wu X, Shen W et al (2019) Collaboration of heterogeneous unmanned vehicles for smart cities. IEEE Netw 33(4):133–137
4. Qi L, Zhang X, Dou W, Ni Q (2017) A distributed locality-sensitive hashing based approach for cloud service recommendation from multi-source data. IEEE J Selected Areas Commun 35(11):2616–2624
5. Philip BV, Alpcan T, Jin J et al (2019) Distributed real-time IoT for autonomous vehicles. IEEE Trans Ind Informatics 15(2):1131–1140
6. Nkenyereye L, Liu CH, Song JS (2019) Towards secure and privacy preserving collision avoidance system in 5G fog based internet of vehicles. Futur Gener Comput Syst 95(06):488–499
7. Guo J, Kim S, Wymeersch H et al (2019) Guest editorial: introduction to the special section on machine learning-based internet of vehicles: theory, methodology, and applications. IEEE Trans Veh Technol 68(5):4105–4109
8. Sharma V (2019) An energy-efficient transaction model for the Blockchain-enabled internet of vehicles (IoV). IEEE Commun Lett 23(2):246–249
9. Zhou YF, Yu HX, Li Z, Su JF, Liu CS (2020) Robust optimization of a distribution network location-routing problem under carbon trading policies. IEEE Access 8(1):46288–46306
10. Xie P S, Han X M, Feng T et al (2020) A Method of Constructing Arc Edge Anonymous Area Based on LBS Privacy Protection in the Internet of Vehicles. International Journal of Network Security 22(2):275–282.
11. Jabri I, Mekki T, Rachedi A et al (2019) Vehicular fog gateways selection on the internet of vehicles: a fuzzy logic with ant colony optimization based approach. Ad Hoc Netw 91(08):1–16
12. Lianyong Qi, Wanchun Dou, Chunhua Hu, Yuming Zhou and Jiguo Yu. A Context-aware Service Evaluation Approach over Big Data for Cloud Applications, IEEE Transactions on Cloud Computing, 2015. DOI: https://doi.org/10.1109/TCC.2015.2511764
13. Sanchez-Iborra R, Santa J, Skarmeta A (2019) Empowering the Internet of Vehicles with Multi-RAT 5G Network Slicing. Sensors 19(14):1–16
14. Qian Y, Jiang Y, Hu L et al (2020) Blockchain-based privacy-aware content caching in cognitive internet of vehicles. IEEE Netw 34(2):46–51
15. Mohammed B, Naouel D (2019) An efficient greedy traffic aware routing scheme for internet of vehicles. Comput Mater Continua 58(2):959–972
16. Yao Z, Jiang Y, Wang Y et al (2019) Discrete model of dynamic heterogeneous traffic flow platoon in internet of vehicles. Beijing Jiaotong Daxue Xuebao 42(2):106–117
17. Tolba A (2019) Content accessibility preference approach for improving service optimality in internet of vehicles. Comput Netw 152(04):78–86
18. Zhang L, Luo M, Li J et al (2019) Blockchain based secure data sharing system for internet of vehicles: a position paper. Vehicular Commun 16(04):85–93
19. Dai Y, Xu D, Maharjan S et al (2019) Artificial intelligence empowered edge computing and caching for internet of vehicles. IEEE Wirel Commun 26(3):12–18
20. Hao R, Yang H, Zhou Z (2019) Driving behavior evaluation Model Base on big data from internet of vehicles. Int J Ambient Comput Intell 10(4):78–95
21. Priyan MK, Devi GU (2019) A survey on internet of vehicles: applications, technologies, challenges and opportunities. Int J Advanc Intell Paradigms 12(1–2):98–119
22. Chen LW, Ho YF (2019) Centimeter-grade metropolitan positioning for lane-level intelligent transportation systems based on the internet of vehicles. IEEE Trans Ind Informatics 15(3):1474–1485
23. Dow CR, Nguyen DB, Cheng S et al (2019) VIPER: an adaptive guidance and notification service system in internet of vehicles. World Wide Web 22(4): 1669–1697
24. Li Y, Wang M, Zhu R et al (2019) Intelligent augmented keyword search on spatial entities in real-life internet of vehicles. Futur Gener Comput Syst 94(05):697–711
25. Ghafoor KZ, Kong L, Rawat DB et al (2019) Quality of service aware routing protocol in software-defined internet of vehicles. IEEE Internet Things J 6(2): 2817–2828
26. Liu K, Xu X, Chen M et al (2019) A hierarchical architecture for the future internet of vehicles. IEEE Commun Mag 57(7):41–47
27. Silva R, Iqbal R (2019) Ethical implications of social internet of vehicles systems. Internet Things J IEEE 6(1):517–531
28. Kaur K, Garg S, Kaddoum G et al (2019) SDN-based internet of autonomous vehicles: an energy-efficient approach for controller placement. IEEE Wirel Commun 26(6):72–79
29. Tang X, Bi S, Zhang YJA (2019) Distributed routing and charging scheduling optimization for internet of electric vehicles. Internet Things J IEEE 6(1):136–148
30. Qi L, Dou W, Wang W, Li G, Yu H, Wan S (2018) Dynamic Mobile crowdsourcing selection for electricity load forecasting. IEEE Access 6: 46926–46937
31. Guan K, He D, Ai B et al (2019) 5-GHz obstructed vehicle-to-Vehicle Channel characterization for internet of intelligent vehicles. Internet Things J IEEE 6(1):100–110

32. Li Y, Zhang W, Zhu R et al (2019) Fog-based pub/sub index with Boolean expressions in the internet of industrial vehicles. IEEE Trans Ind Informatics 15(3):1629–1642
33. Zhou YF, Chen N (2019) The LAP under facility disruptions during early post-earthquake rescue using PSO-GA hybrid algorithm. Fresenius Environ Bull 28(12A):9906–9914
34. Xu X, Xue Y, Qi L et al (2019) An edge computing-enabled computation offloading method with privacy preservation for internet of connected vehicles. Futur Gener Comput Syst 96(07):89–100
35. Qi Q, Wang J, Ma Z et al (2019) Knowledge-driven service offloading decision for vehicular edge computing: a deep reinforcement learning approach. IEEE Trans Veh Technol 68(5):4192–4203
36. Wang Z, Li L, Xu Y et al (2019) Handover control in wireless systems via asynchronous multiuser deep reinforcement learning. IEEE Internet Things J 5(6):4296–4307
37. Wang C, Wang J, Shen Y et al (2019) Autonomous navigation of UAVs in large-scale complex environments: a deep reinforcement learning approach. IEEE Trans Veh Technol 68(3):2124–2136

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.