Received October 5, 2018, accepted November 6, 2018, date of publication February 14, 2019, date of current version February 27, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2882000

Resource Allocation Algorithm With Multi-Platform Intelligent Offloading in D2D-Enabled Vehicular Networks

YAPING CUI, YINGJIE LIANG, AND RUYAN WANG
School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
Key Laboratory of Optical Communication and Networks in Chongqing, Chongqing 400065, China
Key Laboratory of Ubiquitous Sensing and Networking in Chongqing, Chongqing 400065, China

Corresponding author: Yaping Cui (cuiyp@cqupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61771082, Grant 61871062, and 61801065, and in part by the Program for Innovation Team Building at the Institutions of Higher Education in Chongqing under Grant CXTDX201601020.

ABSTRACT The latest research of network and computing contributes greatly to the development of vehicular networks. However, in existing works, these two important enabling technologies are studied separately. To reduce the delay, in this paper, we propose a multi-platform intelligent offloading and resource allocation algorithm which can dynamically organize the computing resources to improve the performance of the next-generation vehicular networks. Considering the task calculation problem, the K-nearest neighbor algorithm is used to select the task offloading platform (i.e., cloud computing, mobile edge computing, or local computing). For the computational resource allocation problem and system complexity in non-local computing, reinforcement learning is used to solve the optimization problem of resource allocation. The simulation results show that compared with the baseline algorithm that all tasks are offloaded to the local or mobile edge computing server, the resource allocation scheme achieves a significant reduction in latency cost, and the average system cost can be saved by 80%.

INDEX TERMS Vehicular networks, mobile edge computing, reinforcement learning, resource allocation.

I. INTRODUCTION

In recent years, the 5G cellular network is being deployed, and the future network will become an ‘‘intelligent system’’ that can ‘‘self-learn’’ and make predictions based on the results of learning, and then make decisions. The whole communication system is self-aware, self-learning, self-planning, self-growth, and self-evolving. Whether it is 5G or B5G (Beyond 5G), massive data and intelligent algorithms will become the important foundations for the further development of wireless networks [1]. With advanced technologies, smart cities are expected to provide their citizens with improved quality of life and a variety of innovative service applications, including transportation, education, healthcare, energy, waste management, tourism, and so on [2]. In terms of transportation, vehicular networks has attracted great interests in both academia and industry. D2D (Device-to-Device) is one of the key technologies of 5G. Under the control of cellular system, D2D communication allows end users to communicate directly within a certain range by sharing cell resources. In addition to the human-centric D2D communication network, D2D is also used in vehicular networks [3]. Vehicular networks uses advanced technologies to connect vehicles with various infrastructures, devices, users, services, etc. The developments of connected vehicles are heavily influenced by information and communication technologies [4]. Vehicular networks’ five advantages: low latency, calculation requirements, high reliability, security certification, and privacy protection. They also become different research aspects of the vehicular networks, and have been closely watched by the academic community [5].

In the network field, vehicular networks develops rapidly. The vehicle can communicate with the vehicle or with the base station or the roadside units(BS / RSU), and can also be connected to the Internet for communication. The vehicular networks will realize the V2X (vehicle to everything) comprehensive network architecture of vehicles, vehicles and vehicles, vehicles and people, vehicles and roads, vehicles and cloud service platforms, which will greatly enhance the intelligentization of automobiles and lay out new traffic management networks. The new traffic management networks can provide data support and reliability guarantee for technologies such as assisted driving and automatic driving [6].

Recent advances in computing will have profound impacts on vehicular networks as well. Considering task processing calculations, cloud computing has become very popular to enable access to a shared pool of computing resources [7], [8]. Nevertheless, as the distance between the cloud and the end device is usually large, cloud computing services may not provide guarantees to low latency applications in vehicular networks while only use cloud computing are not feasible or economical. Recently, Mobile Edge Computing has been specifically proposed for wireless access networks where computing resources are deployed closer to the location of terminal vehicles, which effectively satisfies computational requests that require large computations and low latency [9], [10]. Finally, considering that each vehicle is equipped with an in-vehicle device (VE), its function is similar to a small computer with a networked interface [11]. In this way, the vehicle itself also has certain computing capacity, which can effectively meet the calculation request that requires a small amount of calculation and low latency [12].

In [13], mobile edge computing and local computing are considered simultaneously, by using reinforcement learning and deep reinforcement learning approach to solve the computational unloading problem. However, the computing resources of mobile edge computing servers is limited and the type of vehicle task varies, taking cloud computing into consideration is a way to save computing resources of MEC servers. In [14], the joint work of mobile edge computing and cloud computing is considered, by using the deep reinforcement learning method to determine whether the computing task should be offloaded to the MEC server. Based on the work of [14], in [15], the content of the deep reinforcement learning algorithm is introduced in detail. However, both of these works neglect the computing capacity of the vehicle itself, which can reduce the delay for some computing tasks. In [16], the cooperation between cloud computing and local computing is considered. In the paper, the concept of vehicle cloud computing is proposed, which appropriately coordinates the vehicle cloud (composed of the computing resources of the vehicle) and the remote cloud to provide timely services to users. The scheme proposed in this paper takes into account the addition of computing resources of the BS / RSU to the vehicle cloud, and uses the semi-Markov decision process for resource allocation. However, this work does not explicitly utilize and divide the computing resources of BS / RSU and vehicles. At present, there is no work to consider the collaboration of cloud computing, mobile edge computing, and local computing. Unlike smart devices such as mobile phones, vehicles have more computing resources and should not be ignored. Considering the vehicle's local computing resources into the overall resource allocation problem is a viable and efficient solution.

Inspired by the aforementioned works in [13]–[16], this paper proposes a multi-platform offload intelligent resource allocation algorithm combining cloud computing, mobile edge computing and local computing, aiming at minimizing delay. To improve the performance of next-generation vehicular network, the allocation algorithm proposed in this paper first uses the KNN algorithm to determine whether the task should be offloaded to the cloud computing server, mobile edge computing server or local. For the case of being offloaded to the mobile edge computing server and the cloud computing server, and considering the complexity of the system, we use the reinforcement learning algorithm to select the computing platform again and perform the resource allocation problem of the computing resources. Finally, we have verified that the proposed algorithm can effectively reduce the latency of the system and save the average cost of 80%.

The remainder of this paper is organized as follows. Section II discusses the system model. Section III and IV describe the optimization problem and the proposed joint optimization algorithm. The simulation results are presented in Section V. Finally, the conclusion and future work are explained in the Section VI.

II. SYSTEM MODEL

In this section, the network model, task model, and calculation model are introduced to present the system model.

A. NETWORK MODEL

We consider a vehicular networks model that is divided into three layers for calculation. The vehicle's computing tasks can be performed locally, in a mobile edge computing server or in the cloud server. There are multiple vehicles with computing tasks, as shown in Figure 1 [2]. The characteristics of the cloud computing layer are high computing capacity, high latency and far from the vehicle. In this layer, the tasks require high computational complexity and do not sensitive to latency (e.g.,path planning). The mobile edge computing layer is closer to the vehicle, therefore the computing capacity of the mobile edge computing layer is not as high as the cloud computing layer. Moreover, this layer has a lower delay than the cloud computing layer. It is design to calculate tasks which has general computing demands and sensitive to delays (e.g., real-time location navigation). The local computing layer has relatively weak computing capacity. However, the latency can be very low due to local device advantage when small tasks is calculated. It is design to calculate the tasks that has small size and very sensitive to delays (e.g., road hazard warnings). Here, the set of all BS / RSU and vehicles is defined as $K = \{1, \ldots, k\}$ and $N = \{1, \ldots, n\}$. The set of MEC servers is denoted as $M = \{1, \ldots, m\}$. Considering the heterogeneity of vehicles, the computing resources of different vehicles are different. Considering the heterogeneity of vehicles, the computing resources of different vehicles are different.

We consider that the wireless channel between the vehicle and BS / RSU is a real time-varying channel, so it is modeled

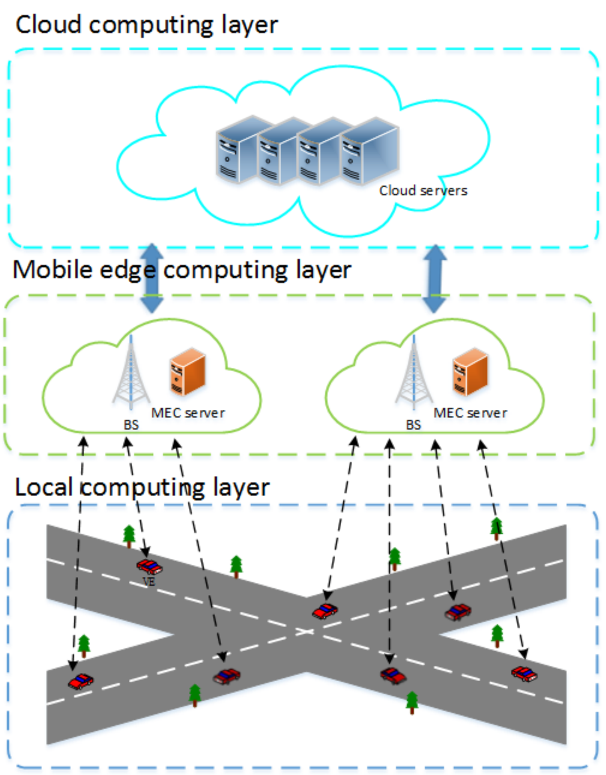

FIGURE 1. Network model.

as Rayleigh channels. For simplicity, this paper does not consider the vertical handover. We assume that each vehicle n has a computing task to complete. Each vehicle n can offload tasks to the cloud or MEC server or execute locally. The capacity of the MEC server is limited and may not be sufficient for all vehicles to offload tasks. We define W as the spectral bandwidth of the wireless channel. Assuming that multiple vehicles choose to offload tasks simultaneously, the wireless bandwidth is allocated evenly to the vehicle to upload data. According to [17], the achievable upload data rate for vehicle n is

$$r_n = \frac{W}{N} \log\left(1 + \frac{P_n h_n}{\frac{W}{N} N_0}\right) \tag{1}$$

where N is the number of tasks, P_n is the transmission power of the vehicle n for uploading data, h_n is the channel gain of the vehicle n in the wireless channel, and N_0 is the variance of complex white Gaussian channel noise [11].

B. TASK MODEL

We assume that each vehicle n has a computing task R_n, defined as $R_n = (B_n, D_n, \tau_n)$, The R_n can be offloaded to the vehicle's local CPU or MEC server or cloud computing server for calculation. Here B_n denotes the size of the input data required to calculate R_n, including the program code and input parameters. D_n denotes the total number of CPU cycles required to complete the calculation task R_n. D_n denotes the amount of computing resources required to complete task R_n. We assume that the size of D_n is a fixed value whether it is executed locally or executed on the MEC server or in the cloud server. τ_n denotes the maximum tolerable delay of task R_n, that means the delay of completing each vehicle should not exceed τ_n, which will be an important constraint of our optimization problem. Define the offloading decision vector as $A = [\alpha_1, \alpha_2, \ldots, \alpha_N]$. We have $\alpha_n = \{\alpha_n^l, \alpha_n^o, \alpha_n^c\}$, $\alpha_n^l, \alpha_n^o, \alpha_n^c \in (0, 1)$, and $\alpha_n^l + \alpha_n^o + \alpha_n^c = 1$. If the task is offloaded to the local, MEC server or cloud server, the corresponding parameter is 1, otherwise it is 0. A computing task R_n can only be offloaded to a platform for calculation.

C. COMPUTATION MODEL

1) LOCAL COMPUTING MODEL

If vehicle n chooses to perform task R_n locally, T_n^l is defined as the local execution delay of vehicle n, which only includes the processing delay of the local CPUs. Then f_n^l is defined as the computation capacity (i.e., number of CPU cycles per second) of vehicle n. Due to the heterogeneity of vehicles, the computation capacity between vehicles may be different. The local execution delay T_n^l of task R_n is

$$T_n^l = \frac{D_n}{f_n^l} \tag{2}$$

Taking into account the above time consumption, the total cost of local computing can be given as

$$C_n^l = T_n^l \tag{3}$$

2) MOBILE EDGE COMPUTING MODEL

If the vehicle n chooses to perform the task R_n at the MEC, the entire offloading will be divided into three steps. First, the vehicle n needs to upload the input data (i.e., program code and parameters) of the task to the BS / RSU through the wireless access networks, and the BS / RSU forwards the data to the MEC server. Then, the MEC server allocates part of the computing resources to perform the computing task, and finally the MEC server returns the execution result to the vehicle.f_n^o is defined as the computation capacity of the MEC server (i.e., number of CPU cycles per second). According to the above steps, $T_{n,t}^o$ is defined as the transmission delay of vehicle n

$$T_{n,t}^o = \frac{B_n}{r_n} \tag{4}$$

where r_n is the upload data rate defined in the wireless channel mentioned in network model. $T_{n,p}^o$ is defined as the processing delay of the MEC server to perform R_n

$$T_{n,p}^o = \frac{D_n}{f_n^o} \tag{5}$$

$T_{n,b}^o$ is defined as the download delay for processing results

$$T_{n,b}^o = \frac{B_n}{r_b} \tag{6}$$

r_b is the download data rate of vehicle n. According to the literature [18] and [19], the download data rate is generally very high, and the resulting data size is much smaller than the data size of the input data, so the delay of this step is ignored in the rest of the paper. Define the total cost of the MEC calculation as C_n^o

$$C_n^o \approx T_{n,t}^o + T_{n,p}^o \tag{7}$$

3) CLOUD COMPUTING MODEL

If the vehicle n chooses to perform the task R_n on the cloud server, the entire offloading will be divided into three steps. First, the vehicle n needs to upload R_n input data (i.e., program code and parameters) to the BS / RSU through the radio access networks, and the BS / RSU forwards the data to the cloud server. Then, the cloud server allocates part of the computing resources to perform the computing task, and finally the cloud server returns the execution result to the vehicle n. f_n^c is defined as the computation capacity of the cloud server (i.e., number of CPU cycles per second). According to the above steps, $T_{n,t}^c$ is defined as the transmission delay of vehicle n

$$T_{n,t}^c = T_{n,t}^o = \frac{B_n}{r_n} \tag{8}$$

here, the transmission delay is the same as the transmission delay at the MEC model, because in this step, the input data of R_n is uploaded to the BS / RSU through the wireless access networks. $T_{n,p}^c$ is defined as the processing delay for the cloud server to perform R_n

$$T_{n,p}^c = \frac{D_n}{f_n^c} \tag{9}$$

and $T_{n,b}^c$ is defined as the download delay for processing results

$$T_{n,b}^c = \frac{B_n}{r_b} \tag{10}$$

The download data rate r_b is the partly same as that in the MEC server. Generally, it is very high. Since the data of task processed by the cloud computing server is large, the data of result is also large, so in the cloud computing model part, the delay of this step cannot be ignored. Considering the above time energy consumption, define the total cost of cloud computing as C_n^c

$$C_n^c = T_{n,t}^c + T_{n,p}^c + T_{n,b}^c \tag{11}$$

and the total cost of the entire system is defined as

$$C_{all} = \sum_{n=1}^{N} a_n^l C_n^l + a_n^o C_n^o + a_n^c C_n^c \tag{12}$$

III. PROBLEM FORMULATION

In this section, we first develop the system's task offload and resource allocation as optimization problems. In this paper, we aim to minimize the total cost of delay of all vehicles in the entire system. Under the limitations of maximum tolerant delay and computing capacity, the problem is expressed as follows:

$$\begin{aligned}
&\min_{A,f} \sum_{n=1}^{N} a_n^l C_n^l + a_n^o C_n^o + a_n^c C_n^c \\
&s.t.\ C1: \alpha_n^l, \alpha_n^o, \alpha_n^c \in (01)\, \alpha_n^l + \alpha_n^o + \alpha_n^c = 1 \\
&\quad C2: a_n^l T_n^l + a_n^o T_n^o + a_n^c T_n^c \le \tau_n, \quad \forall n \in N \\
&\quad C3: 0 \le f_n^o \le a_n^o F, \quad \forall n \in N \\
&\quad C4: \sum_{n=1}^{N} a_n^o f_n^o \le F, \quad \forall n \in N
\end{aligned} \tag{13}$$

In the above formula, A is the offloading decision vector defined as $A = [\alpha_1, \alpha_2, \ldots, \alpha_N]$. f is the computing resource allocation vector defined as $f = [f_1, f_2, \ldots, f_N]$. The goal of the optimization problem is to minimize the total cost of the entire system.

C1 means that each vehicle n chooses to perform its calculation tasks through local computing or MEC or cloud computing;

C2 indicates that the time cost should not exceed the maximum allowable delay;

C3 indicates that the computing resource allocated for the vehicle n cannot exceed the total resource F of the MEC server;

C4 indicates that the sum of the computing resources allocated to the vehicle cannot exceed the total resources F of the MEC server.

Problem (13) can be solved by finding the optimal offload decision vector A and the computational resource allocation vector f. But since A is a binary vector, the feasible set of the problem (13) and the objective function are not convex. In addition, if the number of users increases, the size of the problem (13) will increase rapidly, and the general approach is difficult to solve this non-convex problem [10]. Therefore, we propose to use the reinforcement learning method to find the optimal A and f, and find the optimal solution of the problem (13) by determining the optimal A and f.

IV. THE PROPOSED JOINT OPTIMIZATION ALGORITHM

In this section, we present an intelligent algorithm for task offloading and resource allocation. The proposed algorithm considers which platform (cloud computing server, MEC server, and local) to offload tasks for computing, and then use the resource allocation problem of offloading to MEC server and cloud computing server as a reinforcement learning process [20].

First we need to determine which platform to offload the task to. This is a classification problem that requires all tasks to be divided into three categories. KNN algorithm is used here because this algorithm is simple, efficient, easy to calculate, and can quickly classify tasks. Considering the delay requirement and task size of each task, the algorithm is used to find the platform that is most suitable for offloading. The KNN algorithm is used to calculate the Euclidean distance between each task with cloud computing server, MEC server, and local computing, respectively. The parametersdelay τ and task size B are considered. The distance can be expressed as follows

$$D = \sqrt{(\tau_{R_j} - \tau_i)^2 + (B_{R_j} - B_i)^2} \tag{14}$$

Here we assume that the task latency and computational requirements will be met, and our goal is to find an offloading platform that can provide minimal delay [21]. In this paper, we only consider single-vehicle local computing, without considering the collaborative computing of the vehicle. The KNN algorithm compares each distance, selects the shortest distance, and selects the offloading position of the task. If the task is selected to offload in the local computing, the task will be calculated directly.

When multiple tasks are selected to be offloaded to the MEC server or the cloud computing server, considering the limited computing resources of the MEC server, it is necessary to consider the allocation of computing resources. Vehicular networks is highly dynamic. Whenever the system changes slightly, the optimization problem needs to be recalculated, which leads to huge network overhead. Reinforcement learning is a type of algorithm that refers to the learning from environmental state to behavioral mapping, so as to maximize the cumulative reward value of system behavior from environment. Using the reinforcement learning method can be well adapted to the environment of the vehicular networks and find the best resource allocation solution.

BS / RSU is responsible for collecting the status of the MEC server and then assembling the entire information into a system state. The BS / RSU then sends the constructed system state to the agent and obtains feedback on the optimal strategy for allocating resources for a particular vehicle. The entire process is represented in Algorithm 1.

In the non-local resource allocation process using reinforcement learning, after obtaining the action, the BS/RSU will notify the vehicle. In order to get the optimal strategy, it is necessary to determine the system state, behavior and reward functions. Next, we'll introduce the system state, action, and reward functions, respectively.

A. STATE

Available BS / RSU state $K = \{1, \ldots, k\}$, Available MEC server $M = \{1, \ldots, m\}$, For vehicle us at time interval $t \in \{0, 1, \ldots, T-1\}$, The state vector can be described as $s = (tc, ac)$. Here, $tc = C_{all}$, ac is the available computing capacity of the MEC server, can be defined as $ac = F - \sum_{n=0}^{N} f_n$.

B. ACTION

In our system, actions are defined as offloading decision vector $A = [\alpha_1, \alpha_2, \ldots, \alpha_N]$ and computational resource allocation vector$f = [f_1, f_2, \ldots, f_N]$. The offloading decision vector indicates whether the vehicle task n is offloaded to the MEC server. The computing resource allocation vector represents the amount of computing resources allocated by the MEC server to the task.

Algorithm 1 Intelligent Joint Optimization Algorithm

```
**Phase 1:**Initialization
a) Task $R_i$ Maximum delay $\tau_{R_i}$, Task size $B_{R_i}$
b) Current location of the task $P = 1$
c) $Q$ matrix, parameter $r$, reward matrix $R$
d) local($L$), MEC($M$), cloud($C$), general delay $\tau_1$, $\tau_2$, $\tau_3$, allow maximum task size $B_1$, $B_2$, $B_3$
**Phase 2:**Select task offload location
Compute the Euclidean distance $D$ of $(\tau_{R_i}, B_{R_i})$ and $(\tau_1, B_1)$
$D = \sqrt{(\tau_{R_i} - \tau_1)^2 + (B_{R_i} - B_1)^2}, P = 1$
for $j = 2, 3$ do
    Compute the Euclidean distance $d_j$ of $(\tau_{R_i}, B_{R_i})$ and $(\tau_j, B_j)$
    if $d_j < D$ then
        Then $D = d_j, P = j$
    end if
end for
if $P = 1$ then
    offloading to the local
end if
if $P = 2$ or 3 then
    into the Phase 3
end if
**Phase 3:**Resource Allocation
if $P = 3$ then
    Computing task
end if
if $P = 2$ then
    for each episode do
        Choose a random state $S_t$
        for each step: do
            Choose an action $a$ from all possible actions of state $s$
            Execute chosen $a$, observe reward and $s'$
            Compute
            $Q(s, a) \leftarrow R(s, a) r \times \max[Q(S', allaction)]$
            $s \leftarrow s'$
            until the $Q$ invariance
        end for
    end for
end if
```

C. REWARD

The optimization problem in this paper is to obtain the minimum total cost. The goal of reinforcement learning is to get the maximum reward, and the reward is negatively related to the total cost, therefore defined as $\frac{tc_{local} - tc(s,a)}{tc_{local}}$. Here tc_{local} is the total cost for the local calculation, and $tc(s, a)$ is the total cost of the system.

In the reinforcement learning process, we use the Q-learning algorithm, which is a learning method for recording Q values. Each state action pair has a corresponding value. For each step, the agent is calculated and stored the $Q(s, a)$

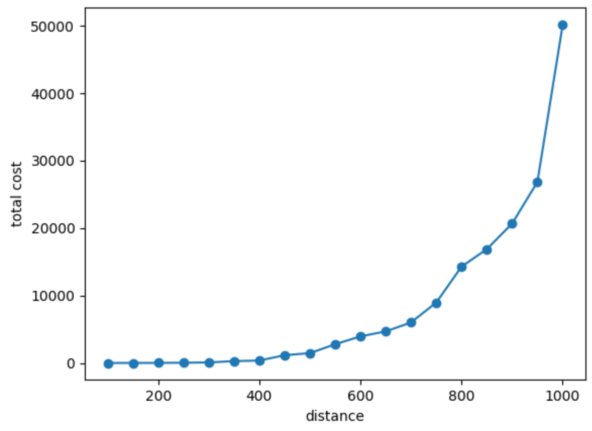

FIGURE 2. Sum cost verus the distance.

in the Q table, and this value can be considered a long-term reward. It can be expressed as:

$$Q(s,a) = R(s,a) + \gamma * maxQ(s', a') \tag{15}$$

Where s, a are the current state and action, s',a' are the next state and action. We define γ as a learning parameter and the value of γ is defined as $0 \leq \gamma \leq 1$. If it approaches 0, which means that the agent mainly considers direct rewards, and if it approaches 1, it means that the agent is very concerned about future rewards. For each step, the $Q\,(s,a)$ is iterated. We can get the optimal value f from Q table.

V. SIMULATION RESULTS AND DISCUSSIONS

A. SIMULATION RESULTS

In the non-local resource allocation process using reinforcement learning, after obtaining the action, the BS/RSU will notify the vehicle. In order to get the optimal strategy, it is necessary to determine the system state, behavior and reward functions. Next, we'll introduce the system state, action, and reward functions, respectively.

In this section, we will present simulation results to evaluate the performance of the proposed scheme. In the simulation, we assume the following scenario. We consider a single small cell with a bandwidth $W = 10MHz$ and the BS / RSU deployed with the MEC server which is centrally located. The vehicles are randomly distributed within a distance of 100 m to 1000 m from the BS / RSU. The CPU's frequency of each vehicle is $f_n^l = 1GHz/sec$, the computing power of the MEC server is $f_n^o = 200GHz/sec$, and the computing power of the cloud computing server is $f_n^c = 1000GHz/sec$. And we assume that the data size of the offloading Bn (in kbits) is subject to a uniform distribution between (1, 50), and the number of CPU cycles D_n (in Megacycles) obeys a uniform distribution between (1, 100). We compare the proposed algorithm with the other two methods, as shown below. "Full Local" means that all vehicle perform their tasks through local calculations. "Full MEC" means that all vehicles offload their tasks to the MEC server.

We first analyze the relationship between the distance from the vehicle to the BS / RSU and the total cost of the system.

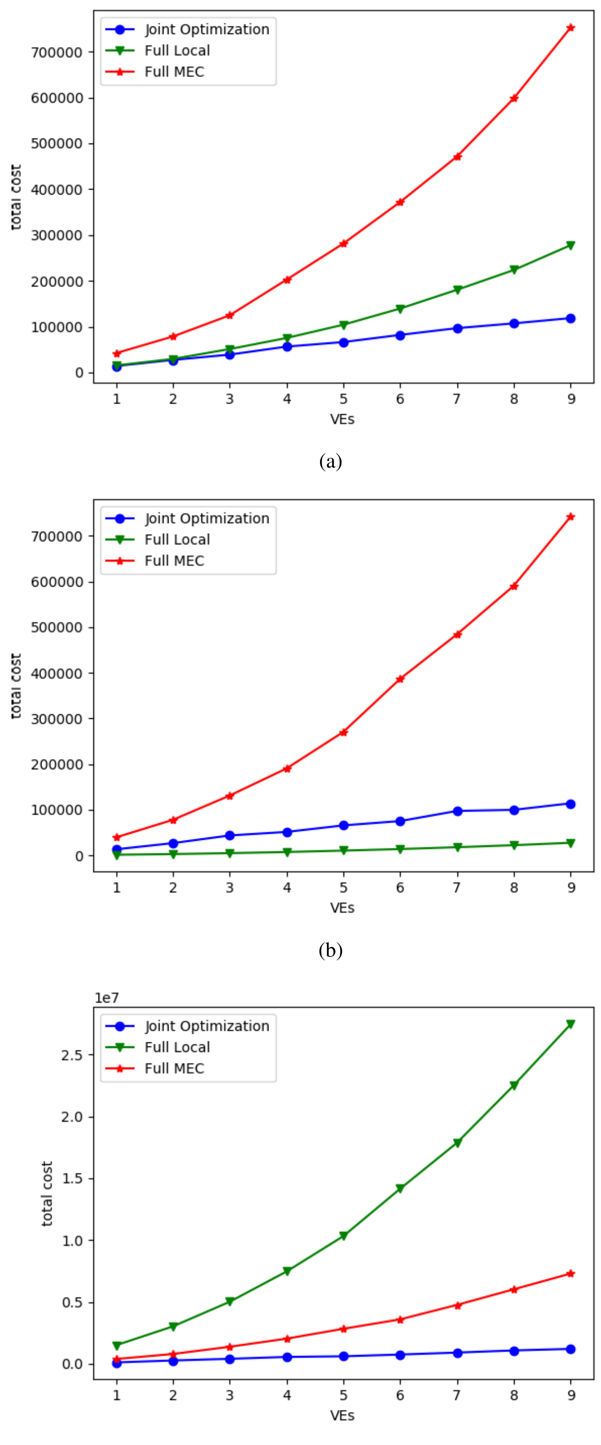

FIGURE 3. Sum cost verus the number of VE. (a) VEs = 5, the local capacity = 1, the MEC server capacity = 200, the cloud server capacity = 1000; (b) VEs = 5, the local capacity = 10, the MEC server capacity = 200, the cloud server capacity = 1000; (c) VEs = 5, the local capacity = 0.1, the MEC server capacity = 200, the cloud server capacity = 1000.

For the problem considered in this paper is resource allocation under the Internet of Vehicles, a major feature of the Internet of Vehicles is the mobility of the vehicle, so it is meaningful to take the impact of the distance from the vehicle to the BS /

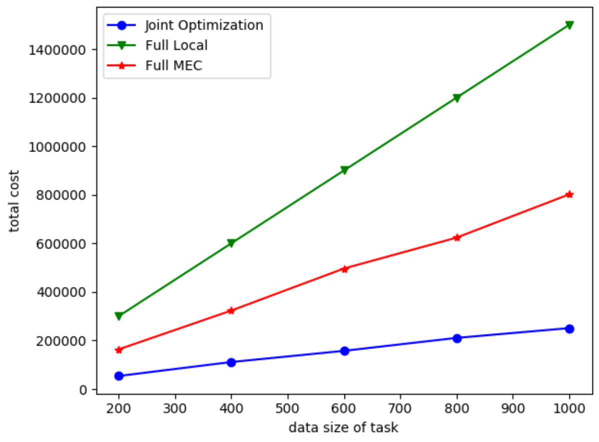

FIGURE 4. Sum cost verus the size of offloading data.

RSU on the system into account. In Figure 2, as the distance from the vehicle to the BS / RSU in the x-axis increases, the curve shows an upward trend, indicating that the total system cost is also increasing. This is because the farther the vehicle from the BS / RSU is, the smaller the uploading data rate of the vehicle task, and the delay of the processing task will increase, the total cost of the system will inevitably increase.

Figure 3 shows the relationship between the number of vehicles and the total cost. In Figure 3, as the total number of vehicles in the x-axis increases, the curves of the three algorithms all show an upward trend. In general, as the number of vehicles continues to increase, the total cost of the three algorithms naturally increases. In Figure 3a, at VEs = 5, compared to the "full MEC" algorithm, the total cost is saved by 80%. Compared with the "Full Local" algorithm, the total cost is saved by 50%. The proposed algorithm obtains the optimal results and proves the effectiveness of the proposed algorithm. In Figure 3b, we can see that when the local computing power increases, and the cost of joint optimization increases faster than the cost of all local processing. When the local computing capacity reduces. In Figure 3c, the local processing cost mostly, and the optimization algorithm proposed in this paper cost leastly. This shows that the local computing capacity will have an impact on the algorithm proposed in this paper. The results show that the algorithm has different adaptability to different computing capacity, and the algorithm is suitable for the case where the local computing capacity is weak.

Figure 4 describes the relationship between the data size B_n of the task and the total cost, where the number of VEs is 5. As shown in Figure 4, as the size of the task in the x-axis increases, the curves of the three algorithms all show an upward trend, indicating that the total cost of all algorithms increases as the size of the task increases. The proposed algorithm achieves the best results because the slope of its growth curve is the smallest, indicating the slowest growth trend. As the size of task increases, The slope of the Full Local curve is larger than the slope of the curves of the other two algorithms, indicating that the total cost of the system is the fastest. This indicates that the larger the size of data for the computing task, the longer the local processing time, resulting in a larger total system cost. The computing capacity of the MEC server is stronger than that of the local, resulting in a total system cost that is less than the total local cost.

B. DISCUSSIONS

Through the analysis of the above simulation results, the proposed algorithm can effectively save the total system cost, but at the same time, it is found that the proposed algorithm has certain adaptability and is suitable for the case where the local computing power is weak. When the local computing power is strong, the total cost to offload the computing task to the local is less. In the future work, we will optimize the performance of the algorithm for this problem, so that the algorithm has a certain universality. At the same time, more optimization indicators will be considered to optimize the total energy consumption of the system. At the same time, we will consider multiple MEC servers and multi-local resource allocation issues [22].

VI. CONCLUTION

In this paper, we propose a multi-platform intelligent offloading and resource allocation algorithm which can dynamically organize the computing resources to improve the performance of vehicular networks. To select the most suitable task offload platform, we considered cloud computing, mobile edge computing, and local computing. In addition, when the task is offloaded to the mobile edge computing server or cloud computing server, we develop the resource allocation strategy as a joint optimization problem. In this paper, we use the new big data reinforcement learning method to solve this problem. The simulation results show that the proposed algorithm based on the reinforcement learning method can effectively save the total system cost and optimize the overall system performance. In the future work, we will consider optimization indicators such as energy consumption to optimize the total system cost and consider the resource allocation problem under multi-MEC server and multi-local computing.

REFERENCES

[1] D. Wu, J. Yan, H. Wang, D. Wu, and R. Wang, "Social attribute aware incentive mechanism for device-to-device video distribution," *IEEE Trans. Multimedia*, vol. 19, no. 8, pp. 1908–1920, Aug. 2017.
[2] Y. He, F. R. Yu, N. Zhao, V. C. M. Leung, and H. Yin, "Software-defined networks with mobile edge computing and caching for smart cities: A big data deep reinforcement learning approach," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 31–37, Dec. 2017.
[3] L. Liang, G. Y. Li, and W. Xu, "Meeting different QoS requirements of vehicular networks: A D2D-based approach," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2017, pp. 3734–3738.
[4] C. Luo, J. Ji, X. Chen, P. Li, and Q. Wang, "Channel state information prediction for 5G wireless communications: A deep learning approach," *IEEE Trans. Netw. Sci. Eng.*, to be published.
[5] D. Wu, S. Si, S. Wu, and R. Wang, "Dynamic trust relationships aware data privacy protection in mobile crowd-sensing," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2958–2970, Aug. 2018.
[6] D. Wu, F. Zhang, H. Wang, and R. Wang, "Security-oriented opportunistic data forwarding in mobile social networks," *Future Gener. Comput. Syst.*, vol. 87, pp. 803–815, Oct. 2018.

[7] H. Zhang, Q. Zhang, and X. Du, ''Toward vehicle-assisted cloud computing for smartphones,'' *IEEE Trans. Veh. Technol.*, vol. 64, no. 12, pp. 5610–5618, Dec. 2015.

[8] Q. Ding, B. Sun, and X. Zhang, ''A traffic-light-aware routing protocol based on street connectivity for urban vehicular ad hoc networks,'' *IEEE Commun. Lett.*, vol. 20, no. 8, pp. 1635–1638, Aug. 2016.

[9] M. Sookhak, F. R. Yu, Y. He, H. Talebian, N. S. Safa, and N. Zhao, ''Fog vehicular computing: Augmentation of fog computing using vehicular cloud computing,'' *IEEE Veh. Tech. Mag.*, vol. 12, no. 3, pp. 55–64, Sep. 2017.

[10] R. Wang, J. Yan, D. Wu, H. Wang, and Q. Yang, ''Knowledge-centric edge computing based on virtualized D2D communication systems,'' *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 32–38, May 2018.

[11] R. Hussain, J. Son, H. Eun, S. Kim, and H. Oh, ''Rethinking vehicular communications: Merging VANET with cloud computing,'' presented at the IEEE Int. Conf. Cloud Comput. Technol. Sci., Taipei, Taiwan, Dec. 2013.

[12] C. Li, S. Wang, X. Huang, X. Li, R. Yu, and F. Zhao, ''Parked vehicular computing for energy-efficient Internet of vehicles: A contract theoretic approach,'' *IEEE Internet Things J.*, to be published.

[13] J. Li, H. Gao, T. Lv, and Y. Lu, ''Deep reinforcement learning based computation offloading and resource allocation for MEC,'' presented at the IEEE Wireless Commun. Netw. Conf. (WCNC), Barcelona, Spain, Apr. 2018.

[14] Y. He, C. Liang, C. Zhang, F. R. Yu, N. Zhao, and H. Yin, ''Resource allocation in software-defined and information-centric vehicular networks with mobile edge computing,'' in *Proc. IEEE 86th Veh. Technol. Conf.*, Toronto, ON, Canada, Sep. 2017, pp. 1–5.

[15] T. He, N. Zhao, and H. Yin, ''Integrated networking, caching, and computing for connected vehicles: A deep reinforcement learning approach,'' *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 44–55, Jan. 2018.

[16] C.-C. Lin, D.-J. Deng, and C.-C. Yao, ''Resource allocation in vehicular cloud computing systems with heterogeneous vehicles and roadside units,'' *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3692–3700, Oct. 2018.

[17] P. Zhao, H. Tian, C. Qin, and G. Nie, ''Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing,'' *IEEE Access*, vol. 5, pp. 11255–11268, 2017.

[18] X. Chen, L. Jiao, W. Li, and X. Fu, ''Efficient multi-user computation offloading for mobile-edge cloud computing,'' *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.

[19] K. Zhang *et al.*, ''Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks,'' *IEEE Access*, vol. 4, pp. 5896–5907, 2016.

[20] V. Mnih *et al.*, ''Human-level control through deep reinforcement learning,'' *Nature*, vol. 518, pp. 529–533, 2015.

[21] H. Ye, L. Liang, G. Y. Li, J. Kim, L. Lu, and M. Wu, ''Machine learning for vehicular networks: Recent advances and application examples,'' *IEEE Veh. Technol. Mag.*, vol. 13, no. 2, pp. 94–101, Jun. 2018.

[22] Z. Zhang, P. Zhang, D. Liu, and S. Sun, ''SRSM-based adaptive relay selection for D2D communications,'' *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2323–2332, Aug. 2018.

YAPING CUI received the B.E. degree in communication engineering from PLA Information Engineering University, Zhengzhou, China, in 2008, and the M.S. degree in communication and information system and the Ph.D. degree in traffic information engineering and control from Southwest Jiaotong University, Chengdu, China, in 2011 and 2017, respectively. From 2011 to 2012, he was with ZTE Corporation as a Baseband Algorithm Engineer. He joined the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China, in 2017, where he is currently a Lecturer. His research interests include millimetre-wave communications, multiple antenna technologies, and smart antennas for vehicular networks.

YINGJIE LIANG received the B.E. degree in software engineering from Sun Yat-sen University, Guangzhou, China, in 2017. She is currently pursuing the master's degree in information and communication engineering with the Chongqing University of Posts and Telecommunications, Chongqing, China. Her research interests include vehicular networks, mobile edge computing, and resource allocation.

RUYAN WANG received the M.S. degree from the Chongqing University of Posts and Telecommunications (CQUPT), Chongqing, China, in 1997, and the Ph.D. degree from the University of Electronic and Science Technology of China, Chengdu, China, in 2007. Since 2002, he has been a Professor with the Special Research Center for Optical Internet and Wireless Information Networks, CQUPT. His research interests include network performance analysis and multimedia information processing.