Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8255733, 13 pages
https://doi.org/10.1155/2022/8255733

Research Article

Adaptive Resource Allocation of Vehicles under Dynamic Environment

Lei Du,[1] **Ru Huo**,[1,2] **Chuang Sun**,[3] **Shuo Wang**,[4] **Jie Guo**,[5] **and Tao Huang**[4]

[1]*Information Department, Beijing University of Technology, Beijing 100124, China*
[2]*Purple Mountain Laboratories, Nanjing 211111, China*
[3]*Department of Automation, Tsinghua University, Beijing 100084, China*
[4]*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[5]*State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China*

Correspondence should be addressed to Ru Huo; huoru@bjut.edu.cn

Received 30 December 2021; Accepted 13 April 2022; Published 4 May 2022

Academic Editor: Abdul Basit

With the rapid development of the Internet of Vehicles (IoV), the capabilities and intelligence of vehicles are rapidly increasing, which will have the potential to support a large number of services for the users of vehicles. However, taking into account the mobility of the vehicle, how to combine the computing resources from nearby vehicles with high mobility and RSUs to provide services collaboratively is a crucial problem. In this paper, we propose an adaptive allocation scheme of computing resources to allocate the resources more reasonably. First, the vehicular mobility model is built to predict the trajectory of vehicles. Second, the sub-networks to execute the V2V communication are constructed according to the prediction of trajectory. Third, the allocation of computing resources from nearby vehicles and RSUs is formulated as an optimization problem and the intelligent optimization algorithm is employed to deal with it. The relative experiments are conducted to verify the effectiveness of the proposed method.

1. Introduction

With the rapid development of urban transportation, the quantity of vehicles has increased dramatically and the road conditions are becoming increasingly complicated. To address with these challenges, the Internet of Vehicles (IoV) and related technologies emerge. The IoV is a network that interconnects pedestrians, cars, and parts of urban infrastructure with the support of advanced information technologies [1] [2]. It uses various sensors, software, in-built hardware, and types of connection to enable reliable and continuous communication. As a part of a smart city, IoV strives to make transportation more autonomous, safe, fast, and efficient, reducing resource waste and detrimental impacts on the environment [3] [4]. Simultaneously, the IoV can provide users with many mobile computing services, such as driving assistance applications, and AR/VR applications. Providing computing services need a large number of computing resources, and some applications have high real-time requirements. However, the computing resources of the vehicle itself are limited, and it is difficult to meet the computing requirements.

In the IoV, roadside units (RSU) are usually deployed with edge servers, which possess rich computing and storage resources [5]. Some scholars have proposed many related task offloading strategies and methods on the IoV [6–9]. The main principle is to offload tasks to the RSUs equipped with edge servers. However, due to the heterogeneity of vehicles and onboard tasks, there is usually an unbalanced load among vehicles. Therefore, some vehicles may have surplus computing and storage resources. Making use of communication technologies such as wireless networks, the onboard tasks which waiting to be executed could be offloaded to the edge side RSU and neighboring vehicles with idle computing

resources. In this way, the computing resources in RSU and neighboring vehicles are allocated, and the utilization of edge system resources and the quality of computing service are improved. The architecture of vehicular edge computing (VEC) is proposed to address with communication problem in IoV [10]. The vehicle-to-vehicle (V2V) is utilized for communications among vehicles. The vehicle-to-infrastructure (V2I) is utilized for communications between vehicles and RSUs. The wired link is utilized for communications among RSUs. The computing architecture can provide users with low-latency, low-energy, and high-reliability computing services by integrating and utilizing computing resources on neighboring vehicles and RSUs.

However, there are still some technical problems in the IoV. First, the high-speed mobility of vehicles will cause real-time changes in the location of the vehicle, which results in a complex and changeable communication environment. For example, the high mobility of vehicles will cause frequent switching of V2I or V2V communications. In addition, when data is transmitted via V2I or V2V, the high mobility of the vehicle will cause the link to be unstable and cause data loss. In addition, the unbalanced geographical distribution of vehicles and their high-speed mobility also bring great challenges for resource allocation of edge systems. Second, users will generate a lot of data or application requests. Due to different request types and requirements, the computing and storage resources of edge devices are different, and communication and network resources are different. Therefore, to improve resource utilization and quality of service (QoS), effective edge resource allocation is required.

Combining the above two considerations, in this paper, we propose a cooperative adaptive scheme of resource allocation, in which the task vehicles and multiple service providers (including service vehicles and RSUs) jointly execute the onboard tasks. The vehicles which require services in this paper are called task vehicle, and the vehicles with idle computing resources are called service vehicle. Firstly, the mobility of the vehicles on the road is modeled and the trajectory of the vehicles is predicted. Secondly, constructing the resource subnetworks consist of vehicles based on the predicted trajectory. The resource subnetworks include resource-rich service vehicles near the task vehicles. Finally, the task vehicles connected to the RSU comprehensively consider the computing power and maximum service time of each service vehicle and decide the tasks assigned to each service vehicle or RSU to minimize the task execution time. Here, the maximum service time of each service vehicle and RSU depends on the relative movement of the task vehicles and the service providers. We must ensure that the task vehicles and each service provider are always within communication range of each other.

The key contributions of this paper are summarized as follows:

(i) We consider modeling the mobility of vehicles with LSTM and social pooling. The vehicular mobility model (VMM) is constructed by analyzing the historical data of target vehicle driving and the interdependence of all vehicle motions in the road section, and it can predict the behavior and trajectory of the vehicles more dynamically and reasonably

(ii) According to the mobility of vehicles and division of road sections, we propose the construction of subnetworks which consists of computing resources from vehicles. The constructed subnetworks can effectively employ the idle resources in the vehicles

(iii) Taking into account the mobility of vehicles and joint allocation of computing resources in the vehicles and the RSUs, we propose an adaptive allocation scheme of computing resources (AASCR) from multiple service providers. The average response time of the tasks and the resource utilization rate of vehicles are regarded as two indicators, and the multiobjective optimization algorithm is employed to address with the allocation of computing resources. The experimental results show that the proposed scheme has reduced the average response time of the task and improved the resource utilization rate comprehensively compared with other approaches

The rest of this paper is organized as follows. In Section 2, some related works on the tasks offloading and computing resources allocation in the IoV are presented. In Section 3, the proposed scheme of resources allocation is introduced in detail. In Section 4, the relative experiments are conducted to demonstrate the performance of the proposed approach. Section 5 presents the conclusion of this paper.

2. Related Work

Nowadays, the VEC can significantly reduce task latency, enabling latency-sensitive applications to run in real time [11]. Lots of scholars had proposed meaningful works on the tasks offloading and allocation of computing resources. VEC models widely employed to provide computing services are mainly divided into two types, one is the VEC model for offloading tasks via V2I, and the other is the VEC model for offloading tasks composed of V2I and V2V.

Regarding the VEC model of offloading tasks via V2I, Zhou et al. [6] studied the problem of multiuser computing offloading on a single edge server, intending to shorten the completion time of virtual reality (VR) applications on the IoV. They divided the VR task into two subtasks so that the vehicle and the mobile edge computing (MEC) server can perform task calculations together. On this basis, they proposed an algorithm to jointly optimize the offload ratio, communication resources, and computing resource allocation. Du et al. [7] considered the cost of vehicles and edge servers. They formulated a bilateral optimization problem to minimize the cost of offloading both parties. On the vehicle side, the unloading decision and the local central processing unit (CPU) frequency are jointly optimized. On the edge server side, wireless resource allocation and service provision are considered at the same time. Different from the abovementioned research on a single edge server, Liu et al. [8] considered a scenario with multiple edge servers. They studied the problem of

multivehicle computing and unloading in this scenario, to reduce the computing overhead of vehicles. Taking into account the coupling of vehicle unloading decisions, they employed game theory methods to make the best unloading decisions for the vehicle and select the appropriate channel. Zhang et al. [9] proposed a hierarchical cloud-based offloading framework in which multiple MEC servers can share a backup server to make up for their lack of computing resources. Then, the Stackelberg game method is used to obtain the optimal offloading strategy to maximize the revenue of the MEC server under the task delay constraint. Hou et al. [12] proposed the tasks offloading model based on the software defined network (SDN) and edge assistance, by integrating the computing resources of mobile edge computing nodes and fixed edge computing nodes to support computing-intensive and delay-sensitive applications.

Regarding the VEC model for offloading tasks composed of V2I and V2V, Hung et al. [13] studied how to ensure low-latency communication in a V2V network with a V2I network as the underlying network. To solve this problem, based on the observation of vehicle data queues, this paper proposed an algorithm that focuses on the effective operation and resource redistribution of network associations, thereby avoiding the exchange of a large amount of control information and signaling, and reducing its delay. Truong et al. [14] combined fog computing and SDN and proposed a new vehicle self-organizing network. The network combines the excellent properties of SDN in terms of flexibility, expansion, and programming, the ability of SDN to learn global information, and the low-latency service of fog computing, thereby enhancing V2V and V2I communications and optimizing resource utilization rate and reduce service delay. Tang et al. [15] proposed a three-tier architecture based on mobile devices, vehicles, and RSUs for scheduling tasks from mobile devices. The goal is to use the computing resources of vehicles and RSUs to reduce task delay as much as possible. At the same time, this paper proposed a task scheduling algorithm based on greedy thinking, to optimize the total response time of tasks under the constraints of connection duration, computing resources, and budget. Raza et al. [16] proposed a mobile-aware partial task migration algorithm to reduce the total response time of the task. Ye et al. [17] proposed a hybrid fog computing architecture based on fog computing radio access network and vehicle fog computing to improve flexibility and data processing capabilities. This paper proposed an optimization algorithm based on deep learning to reduce task delay. Ma et al. [18] integrated the parked vehicles on and off the street into a parking cluster and proposed an algorithm for joint edge server selection and resource allocation to improve task migration performance. Huang et al. [19] proposed a task migration and resource allocation strategy that combines task types and vehicle speed perception to reduce vehicle energy consumption and task processing delay.

Summarizing the above references, although many scholars have done a lot of work on vehicular task offloading and allocation of computing resources, there are still many disadvantages. First, during the process of tasks offloading and executing, the vehicles cannot stay still. Due to the movement of the vehicles, the task vehicles will be out of the communication ranges of RSUs and services vehicles. To maintain the stability of the service, the task vehicles are supposed to be within the communication range during the movement. Therefore, we consider modeling the mobility of vehicles to forecast traffic flow briefly. The mobility of vehicles is regarded as a crucial basis to allocate computing resources. Second, driving and moving vehicles will trigger the temporal and spatial migration of traffic flow and IoV services, thereby changing the adaptation relationship between traffic conditions and computing resources. Due to the high dynamic characteristics of the vehicle itself, the adaptation relationship between the IoV service and the allocation of computing resources becomes extremely complicated. Therefore, we proposed an adaptive allocation scheme to flexible and efficient schedule computing resources according to traffic flow and services requires. The limited computing resources can be dynamically matched with various services to support the corresponding service requirements.

2.1. Adaptive Allocation Scheme of Computing Resources. In this section, we will propose an adaptive allocation scheme of computing resources (AASCR) from multiple service providers. The scheme is introduced in six subsections, which are the structure of AASCR, vehicular mobility model, vehicular task model, communication system model, computation system model, and adaptive allocation scheme of computing resources.

2.2. Architecture Design of AASCR. In this paper, the scenario we consider is a straight section of urban road. As depicted in Figure 1, the road in the scenario can be divided into multiple sections $S = \{s_1, s_2, \cdots, s_m\}$. S represents a set of sections, where $s_i (i = 1, 2, \cdots, m)$ represents one of the road sections, and m represents the quantity of road sections. The $U = \{u_1, u_2, \cdots, u_m\}$ represents the set of RSUs, where $u_i (i = 1, 2, \cdots, m)$ represents one of the RSUs. Every section s_i is within the coverage of RSU u_i. Each RSU u_i is equipped with an edge server and provides services for vehicles. The RSUs are interconnected through the Internet with wired links. Therefore, RSUs can communicate with each other through the infrastructure-to-infrastructure (I2I). When the vehicle is within the coverage of the RSU, the vehicle communicates with the RSU through the V2I via long-term evolution (LTE) protocol. Meanwhile, vehicles communicate with each other via V2V, which is implemented through the 802.11p protocol. We employ $V = \{V^{1-1}, V^{1_2}, \cdots, V^{m_n}\}$ to represent the set of vehicles on the road, where $V^i (i = 1, 2, \cdots, m)$ represents the set of vehicles on the section s_i, and $V^{i_j} (j = 1, 2, \cdots, n)$ represents the set of vehicles in the subnetworks of vehicular resources. Each subnetwork is a resources pool that executes V2V tasks, which is constructed based on the mobility of vehicles on the road section. Here, n is the quantity of subareas. We set $v_k (k = 1, 2, \cdots, o)$ to represent each vehicle in the V, where o is the quantity of vehicles. In this scenario, V2V communication links are established among the vehicles in the subnetwork. At the same time, each vehicle establishes a V2I

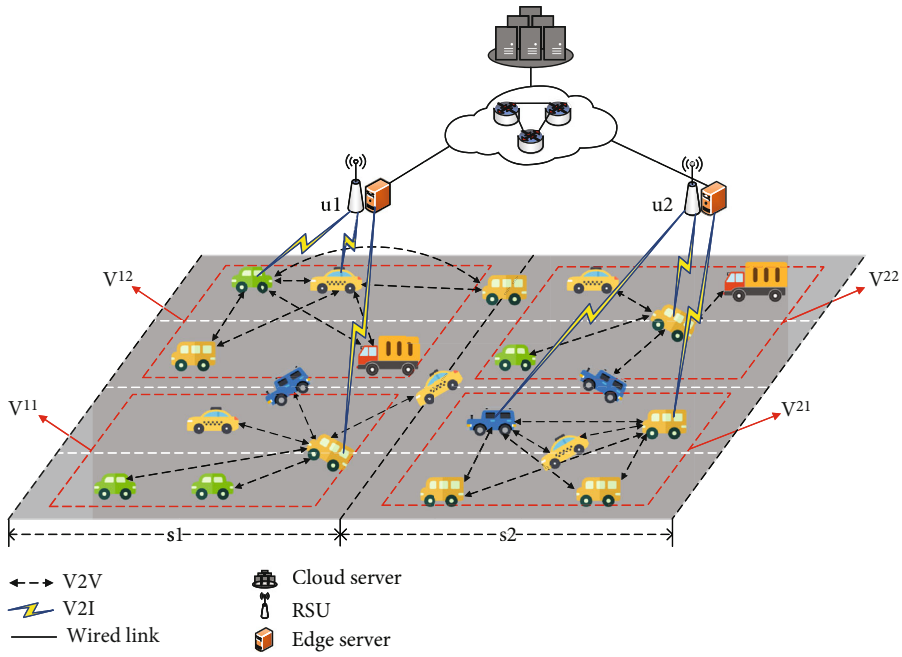


Figure 1: The architecture design of AASCR.

communication link with the corresponding RSUs on the road section to realize the communication between the vehicle and the RSU. It should be noted that vehicles between different subnetworks do not establish a V2V communication link. Therefore, vehicles between different subnetworks cannot directly communicate with each other.

There are main parameters and corresponding values in the AASCR listed in Table 1.

The vehicles in the same subnetwork will maintain the communication link until they travel to the next road section. In addition, when the task from the task vehicle is transferred to the RSU u_i, the task vehicles move out of the communication coverage area of the RSU u_i during the execution of the tasks, and the result of the task will be forwarded by the RSU u_i to the current RSU u_{i+1}. Then, the current RSU u_{i+1} will send the result of the task to the task vehicle.

2.3. Vehicular Mobility Model. In AASCR, different vehicles are participating in the traffic, and their moving directions, angles, and speeds are distinct. To maintain the stability of V2V communication and better build subnetworks of resources, we propose the vehicular mobility model (VMM) to model and analyze the mobility of vehicles and predict the trajectory of the vehicles. Dividing the subnetworks according to the predicted trajectory will make task offloading more reasonable, thereby improving the efficiency and performance of the system.

As shown in Figure 2, a stationary reference coordinate is utilized to represent the position information of the vehicles in AASCR. The x-axis is the direction of road movement, and the y-axis is the position of the different lanes on the road. This allows our model to be independent of how the vehicle trajectory is acquired, especially in the case of onboard sensors for autonomous vehicles. This also makes the model independent of road curvature and can be applied anywhere on the highway as long as the vehicle-mounted lane estimation algorithm is available. The blue dotted lines in Figure 2(a) represent the trajectory of vehicles on the road. The red lines in Figure 2(b) represent the prediction of trajectory when the vehicle moves. The width of each lane is 3.5 m, and the length of each road section is 200 m.

We set the historical trajectory of the vehicles as the input for the VMM. The historical trajectory of the vehicles on the road are represented in $A = \{a^{t-h}, \cdots, a^{t-1}, a^t\}$, where $a^t = \{[x_0^t, y_0^t], [x_1^t, y_1^t], \cdots, [x_l^t, y_l^t]\}$ is the x and y coordinates of the vehicle being predicted and all the vehicles on the road, and h is the number of frames on the vehicular historical trajectory, and l is the quantity of vehicles on the road. The output of VMM is a probability distribution as $B = \{b^{t+1}, b^{t+2}, \cdots, b^{t+g}\}$, where $b^{t+1} = \{x_0^{t+1}, y_0^{t+1}\}$ is the predicted coordinates of the vehicle, and g is the number of frames on the predicted trajectory. The conditional distribution in VMM is $P(B \mid A)$, and it is depicted in

$$P(B|A) = \sum_i P_\Theta(B|\mathrm{op}_i, A) P(\mathrm{op}_i|A), \tag{1}$$

where $\Theta = \{\Theta^{t+1}, \Theta^{t+2}, \cdots, \Theta^{t+g}\}$ are the parameters of the binary Gaussian distribution for each time step in the future, corresponding to the mean and variance of the future position, and $\mathrm{op}_i (i = 1, 2, \cdots, n_{\mathrm{op}})$ represent the operations of vehicles on the road. We consider one kind of lateral operation and three kinds of longitudinal operations as conveyed in Figure 2(b). Lateral operation is the vehicle moves forward. Longitudinal operations include left and right lane changes and lane maintenance.

Table 1: List of crucial notations in AASCR.

Parameter	Value
S	The set of sections on the road
s_i	The road section i
m	The quantity of road section
U	The set of RSUs
u_i	The RSU i
V	The set of vehicles on the road
V^i	The set of vehicles on the section s_i
V^{i-j}	The set of vehicles on the subnetwork in section s_i
v_k^{i-j}	The vehicle in the subarea V^{i-j}
o	The quantity of vehicles

The LSTM encoder is employed to learn the mobility of the vehicles. For each moment, the most recent h frames of the trajectory histories are inputted into the LSTM encoder to predict the target vehicle and all the nearby vehicles around it. In addition, although the LSTM encoder captures the mobility of the vehicles, it cannot capture the interdependence of all vehicle motions in the scene. Therefore, in order to solve this problem, we adopted social pooling [20] to pool the LSTM states of all predicted vehicles around the target vehicle into a social tensor. This is achieved by defining a spatial grid around the predicted target and filling the grid with LSTM states according to the spatial configuration of the vehicles in the scene. There is the structure of VMM in Figure 3.

In Figure 3, firstly, the LSTM encoder is used to learn the mobility of the vehicles; then, the social tensor is employed as the input of the model and coupled with the LSTM state of the predicted vehicle to improve the accuracy of future operation prediction. The model can now access the motion state of surrounding vehicles and their spatial configuration. Lastly, the LSTM-based decoder is used to generate the prediction distribution of the future motion of the next frame. We solve the inherent multimodality problem of driver operation by predicting the distribution for each of the above three kinds of operations and the probability of each operation. Using VMM, we can clearly predict the trajectory of the vehicle and construct more reasonable subnetworks.

As the distances among vehicles or distances between vehicles and RSUs, we assume two devices have the coordinate, respectively, are $co_1 = (x_1, y_1)$ and $co_2 = (x_2, y_2)$. The distance Dis between co_1 and co_2 is obtained by

$$\mathrm{Dis} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \tag{2}$$

There are main parameters and corresponding values in the VMM listed in Table 2.

In AASCR, some task vehicles will generate some onboard calculation tasks in certain periods. We denote these generated calculation tasks as $R = \{r_1, r_2, \cdots, r_p\}$, where r_i represents one of the tasks and p is the quantity of tasks in certain periods. For each r_i, we employ $r_i = \{rd_i, cd_i, t_i, tv_i, l_i, lr_i\}$ to represent the specific information about the onboard task, where rd_i represent the size of raw data to execute the r_i, which is the size of the data when task uploading, and cd_i represents the size of result data about the task, which is the size of the data when results are downloaded, and t_i represents the arrival time of r_i. The tv_i represents the task vehicle, and l_i represents the provider of computing resources, which could be RSU or service vehicle. The execution of r_i need the computing resources from l_i, where $tv_i \in V$ and $l_i \in U \cup V$. The lr_i will be set as 0 when the task is executed in RSU. Otherwise, it will be set as 1 if the task is executed in the service vehicle. In this paper, the computing resources of the vehicle are limited. Therefore, we set that a service vehicle can only handle one task.

2.4. Communication Model. In AASCR, the channel model of V2I and V2V communication is based on orthogonal frequency division multiplexing (OFDM). For V2I communication, each RSU can evenly divide the bandwidth into equal sizes, thereby realizing the simultaneous service of multiple vehicles. For V2V communication, one service vehicle can only serve one task vehicle at a time period. So, its bandwidth cannot be divided. In addition, we assume that there is no channel interference between different subnetworks. The Bw_r is set as the channel bandwidth of V2I communication, and the Bw_v is the channel bandwidth of V2V communication. Therefore, the channel bandwidth Bw of r_i offloading is depicted in

$$\mathrm{Bw} = \begin{cases} \mathrm{Bw}_r, & lr_i = 0, \\ \mathrm{Bw}_v, & lr_i = 1. \end{cases} \tag{3}$$

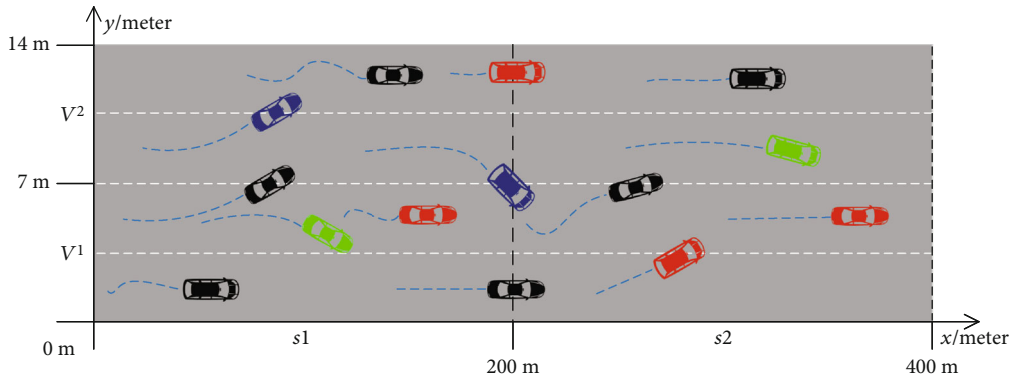
About the channel loss, we employ $\mathrm{cl}(t_i, tv_i, l_i)$ represent the channel loss between tv_i and l_i at the time t. The distance at the time t between tv_i and l_i can be indicated as $\mathrm{Dis}(t_i, tv_i, l_i)$. For V2V, the channel loss can be approximately equal to the determined path loss [21]. Therefore, $cl(t_i, tv_i, l_i)$ can be obtained via

$$\mathrm{cl}(t_i, tv_i, l_i) = \begin{cases} 103.4 + 24.2 \log_{10}(\mathrm{Dis}(t_i, tv_i, l_i)), & lr_i = 0, \\ X(\mathrm{Dis}(t_i, tv_i, l_i))^{-\alpha}, & lr_i = 1. \end{cases} \tag{4}$$

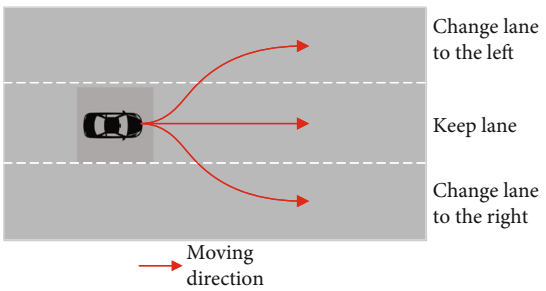
where X represents the path loss coefficient and α is the path loss exponent. We assume that the transmission power of each vehicle is the same and fixed. According to the Shannon theorem, at time t_i, the uplink transmission rate $\mathrm{tsr}^{\mathrm{up}}$ of the tv_i offload tasks to l_i can be obtained by

$$\mathrm{tsr}^{\mathrm{up}}(tv_i, l_i, E) = \mathrm{Bw} \bullet \log_2\left(1 + \frac{P \bullet \mathrm{cl}(t_i, tv_i, l_i)}{N + \sum P \bullet cl(t_i, e, l_i)}\right), \tag{5}$$

where E represents the set of devices connected to l_i through the same wireless channel at the time t_i and $e \in \{E - \{tv_i\}\}$. N represents the white Gaussian noise.



(a) Trajectory of vehicles on the road



(b) Prediction of vehicular trajectory

Figure 2: Mobility of the vehicles.

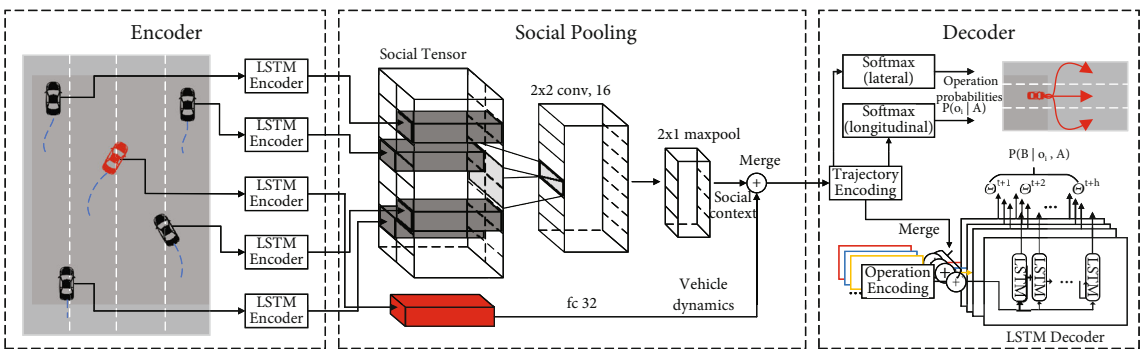


Figure 3: The structure of VMM.

Table 2: List of crucial notations in VMM.

Parameter	Value
A	The historical trajectory of the vehicles
a^t	The x and y coordinates of the vehicles
h	The number of frames on the vehicular historical trajectory
B	The output of VMM
b^{t+1}	The predicted coordinates of the vehicle
g	The number of frames on the predicted trajectory
$P(B \mid A)$	The conditional distribution in VMM
Θ	The parameters of the binary Gaussian distribution
op_i	The operations of vehicles on the road
n_op	The quantity of operations
Dis	The distances among vehicles or distances between vehicles and RSUs

Input: v_k: the vehicle in V on the road;
A: the track histories.
$cc_k = [x_k^t, y_k^t]$: the current coordinate of v_k.
$pc_k = [x_k^{t+1}, y_k^{t+1}]$: the predictive coordinate of v_k.
o: the quantity of vehicles;
$cr_{i\_j} = [sx_{i\_j} \sim ex_{i\_j}, ey_{i\_j} \sim sy_{i\_j}]$:
The coordinate range of subarea s_{i_j}.
Output: the subnetwork set V^{i_j} of vehicles
1: Build the VMM;
2: Input track histories A of V;
3: for i=1 to epoch do
4: Neural network training;
5: end for
6: for k=1 to o do
7: if $cc_k \in cr_{i\_j}$ do
8: $V^{i\_j} = V^{i\_j} \cup v_k$;
9: end if
10: Predicting the trajectory of v_k;
11: if $pc_k \in cr_{i+1\_j+1}$ do
12: $V^{i+1\_j+1} = V^{i+1\_j+1} \cup v_k$;
13: end if
14: end for
15: return V^{i_j}.

Algorithm 1: The construction of subnetworks.

So, the time delay D_i^{up} of task r_i upload is calculated via

$$D_i^{\mathrm{up}} = \frac{rd_i}{\mathrm{tsr}^{\mathrm{up}}(tv_i, l_i, E)}. \tag{6}$$

According to formulas (6) and (11), time point t_i^f when the results of the task return from l_i to vehicle tv_i is expressed as

$$t_i^f = t_i + D_i^{\mathrm{up}} + D_i^E. \tag{7}$$

Therefore, the downlink transmission rate during the result of task return is depicted by

$$\mathrm{tsr}^{\mathrm{dn}}(tv_i, l_i, F) = \mathrm{Bw} \bullet \log_2\left(1 + \frac{P \bullet cl\left(t_i^f, tv_i, l_i\right)}{N + \sum P \bullet cl\left(t_i^f, tv_i, f\right)}\right), \tag{8}$$

where F represents the set of devices connected to tv_i through the same wireless channel at the time t_i^f, and $f \in \{F - \{l_i\}\}$. N represents the white Gaussian noise.

So, the time delay D_i^{dn} of task r_i result download is calculated via

$$D_i^{\mathrm{dn}} = \frac{rd_i}{\mathrm{tsr}^{\mathrm{dn}}(tv_i, l_i, F)} \tag{9}$$

The communication delay D_i^{ud} of r_i, including the communication delay of offloading tasks and downloading calculation results, which can be defined in

$$D_i^{\mathrm{ud}} = D_i^{\mathrm{up}} + D_i^{\mathrm{dn}}. \tag{10}$$

2.5. Computation Model. In AASCR, there are three ways to perform tasks, tasks are executed locally, tasks are offloaded to the service vehicles in subnetwork, and tasks are offloaded to the RSUs. We set the computing power of vehicles as homogeneous. C represents the number of CPU cycles required for addressing with a bit of data. So, the CPU clock cycle required for the task is set as $C \bullet rd_i$, and f_r and f_v, respectively, denote the CPU frequency of the edge server equipped in RSU and vehicle. Therefore, the computing delay D_i^E of r_i can be formulated in

$$D_i^E = \begin{cases} \dfrac{C \bullet rd_i}{f_r}, & lr_i = 0, \\ \dfrac{C \bullet rd_i}{f_v}, & lr_i = 1. \end{cases} \tag{11}$$

2.6. Adaptive Allocation Scheme. In AASCR, we assume that the tasks generated in each task vehicle are indivisible. The vehicle can perform at most one task at the same time. To evaluate the performance of the scheme regarding the allocation of computing resources, we consider the construction of subnetworks. The algorithm of subnetwork construction is as follows. Firstly, the VMM is prebuilt and trained with the track histories of the vehicles on the road. Secondly, during the construction of the subnetworks, the current and just past tracks of the vehicles are employed to predict the vehicle trajectories for a period of time in the future. Thirdly, the resource subnetworks are divided according to the predicted results and the location of the vehicle for a period of time in the future. The subnetworks are constructed reasonably through the algorithm and employed for resource allocation of task vehicles on the corresponding road section.

We can formulate the allocation scheme of computing resources as an optimization problem. According to the communication delay and computing delay, the average response time of the task is obtained via formula (12). The smaller response time indicates better performance of the method. So, the average response time can be regarded as one of the objective functions waiting for optimization.

$$T\_r = \frac{1}{p}\sum_{i=1}^{p}\left(D_i^{\mathrm{ud}} + D_i^E\right). \tag{12}$$

To further evaluate the performance of the allocation scheme, we designed another evaluation index. The resource utilization rate of vehicles is a crucial point for the allocation scheme, and it can reflect performance in terms of load. The resource utilization rate of vehicles is defined in

$$R_u = \frac{t_v}{n_V}, \tag{13}$$

where t_v are the tasks that execute with computing

Input: R_t: the set of tasks which arrive at the time t;
V^{j_k}: the subnetwork set of vehicles;
iter: maximum number of iterations;
Output: the set L_t of providers for computing resources.
initial $L_t = \varnothing$;
for r_i in R_t do
  for V^{j_k} in V do
    if $tv_i \in V^{j\_k}$ do
      initial $a = 0$;
      initial population pop;
      while $a <$ iter do
        $\text{pop}_1 = \text{Crossover\&mutation}(\text{pop})$;
        Calculate T_r and R_u;
        $F = \text{Fast-nondominated-sort}(\text{pop}_1)$;
        $\text{pop}_2 = \text{Crowding-distance}(F)$;
        $\text{pop} = \text{pop}_2$;
        $a = a + 1$;
      end while
    end if
  $L_t = L_t \cup \text{pop}$;
  end for
  end for
  return L_t.

Algorithm 2: The allocation scheme of computing resources.

resources in vehicles and n_V is the quantity of vehicles on the road.

So, the formulated problem can be depicted in

$$\begin{cases} \min T\_r, \\ \max R\_u, \end{cases} \quad \text{e.t.} \begin{cases} lr_i \in \{0, 1\}, \\ l_i \in \left(V^{i_j} \setminus \{tv_i\}\right) \cup u_i. \end{cases} \tag{14}$$

When a task arrives, the task is first offloaded to the nearest and resource-rich vehicle subnetworks, thereby minimizing the average response time T_r of the task. The communication delay between vehicles is far smaller than that between vehicles and RSU. Therefore, in order to reduce the average response time of the task, we tend to transfer the task to the nearest and resource-rich vehicle subnetworks. The allocation scheme of computing resources is formulated in Algorithm 2 as follows. The optimization algorithm employed in this paper is NSGA-II. The $R_t \subseteq R$ represent the set of tasks that arrive at the time t. The RSU has its own covered road information, which includes the position of the vehicle on the road and the state of the computing resources in the vehicle. The L_t represent the set of providers for computing resources, which could be RSU or service vehicle.

3. Results and Discussion

In this section, the simulation of urban mobility (SUMO) is employed to simulate the realistic scenario for evaluating the VMM, AASCR, and the corresponding algorithms. The more detailed introduction is as follows.

3.1. Datasets and Experimental Parameters. The publicly available NGSIM US-101 is utilized as the training dataset for VMM. Each dataset consists of trajectories of real freeway traffic captured at 10 Hz over a time span of 45 minutes. Each dataset consists of 15 min segments of mild and congested traffic conditions. We use SUMO and VMM to simulate the prediction of vehicle trajectory. The status information of vehicles is generated through SUMO. Then the construction of resource subnetworks is also carried out under this simulation software. The notations of the simulation are depicted in Table 3. The datasets of evaluation from [22] are shown in Table 4.

3.2. The Performance Comparison of Different Schemes. In this paper, the performance of our proposed scheme AASCR is compared with three different methods. The first is IDM which is the vehicular network model only with V2I communication. The second is OWV which is the vehicular network model only with V2V communication. The last one is DVNM [23] which combines the V2I and V2V communication.

The proportion of task vehicles is represented as prop_tv in formula (15), which is the proportion of vehicles that require services in all vehicles in a road section at the time t.

$$\text{prop}_{tv} = \frac{n_{tv}}{n_{V^i}}, \tag{15}$$

where n_tv is the number of tasks in s_i. n_V^i is the number of

Table 3: List of simulation notations.

Parameters	Value
Bw_r	30 MHz
Bw_v	10 MHz
f_r	4 GHz
f_v	2 GHz
P	0.1 W
N	10^{-13} W
X	-21.06 dBm
α	1.68
Length of the road section	200 m
Time segment size	50 s

Table 4: The parameters of evaluation.

Datasets	Data 1	Data 2	Data 3	Data 4	Data 5
$C \bullet rd_i$ (cycles)	200	400	600	800	1000
rd_i (KB)	200	400	600	800	1000
cd_i (KB)	40	80	120	160	200

vehicles in s_i. The density of vehicles is defined as follows. If $20 \le n\_V^i \le 25$, the density is low. If $60 \le n\_V^i \le 65$, the density is high.

To compare the performance of IDM, OWV, DVNM, and AASCR in terms of the average response time of tasks, we conduct the relative experiments on different kinds of datasets for low vehicle density and prop_$tv = 0.5$. In Figure 4, we can see that the average response time of tasks for AASCR is always less than IDM, OWV, and DVNM for all kinds of datasets. Simultaneously, with the increase of data size for each task, the average response time of AASCR is growing slower than that of IDM, OWV, and DVNM. Therefore, the performance of computing resources allocation on AASCR is better than IDM, OWV, and DVNM.

To further compare the performance of IDM, OWV, DVNM, and AASCR in terms of the average response time of tasks, we conduct the relative experiments on the different proportions of task vehicles and vehicle densities with the dataset of Data 2.

In Figure 5, it depicts the performance for IDM, OWV, DVNM, and AASCR in terms of the average response time of tasks on different vehicle densities. When the proportion of task vehicles is greater than 90%, IDM, DVNM, and AASCR tend to increase linearly, and OWV tends to decrease continuously. In Figure 5(a), the average response time of tasks for AASCR is always less than IDM and DVNM in low vehicle density for all proportions of task vehicles. In addition, AASCR outperforms OWV in terms of the average response time of tasks, when the proportion of task vehicles is less than 70%. Besides, the average response time of tasks on OVM is less than that of other schemes when the proportion of task vehicles is more than 70%. This is because only the computation resources in vehicles are utilized in OVM, and the number of vehicles is less than the number of tasks when the proportion of task vehicles is greater than 40%. Therefore, there are not enough computation resources in OVM for completing all tasks. As a result, a large number of tasks can not be completed. Therefore, we can conclude that our scheme outperforms OVM in terms of the average response time of tasks. In Figure 5(b), the performance for IDM, OWV, DVNM, and AASCR in terms of the average response time of tasks on high vehicle density. Same as in (a), the average response time of tasks for AASCR is always less than IDM and DVNM in low vehicle density for all proportions of task vehicles. Otherwise, AASCR outperforms OWV in terms of the average response time of tasks when the proportion of task vehicles is less than 60%. Besides, the average response time of tasks on OVM is less than that of other schemes when the proportion of task vehicles is more than 70%. Because of the unsuccessful tasks of OWV, we can conclude that our scheme outperforms OVM in terms of the average response time of tasks.

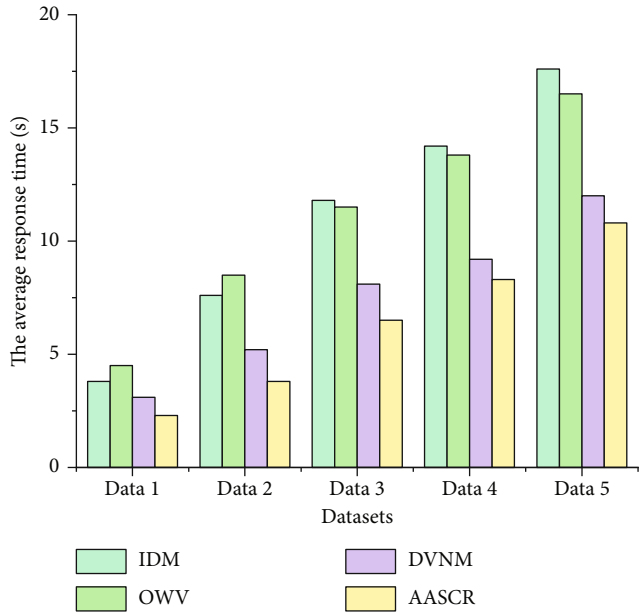


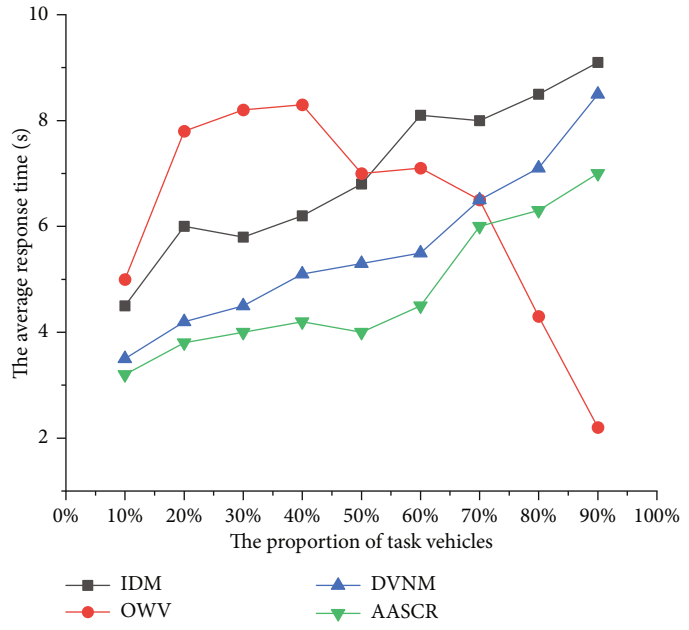
Figure 4: The average response time for different datasets.

For the proportion of successful tasks which is represented as prop_st in formula (16), which is the proportion of tasks successfully executed at the time t. We set n_st as the quantity of tasks which are lost or uncompleted and n_tv as the number of tasks in s_i.

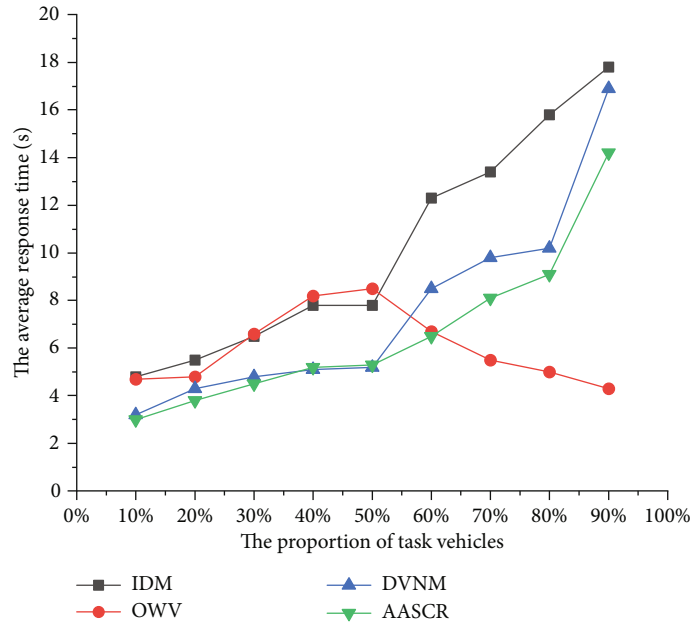
$$\text{prop}_{st} = \frac{n_{st}}{n_{tv}}. \tag{16}$$

To compare the performance of IDM, OWV, DVNM, and AASCR in terms of the proportion of successful tasks, we conduct the relative experiments on the different proportions of task vehicles and vehicle densities with the dataset of Data 2.

In Figure 6, it depicts the performance for IDM, OWV, DVNM, and AASCR in terms of the proportion of successful



(a) Low vehicle density



(b) High vehicle density

Figure 5: The average response time for the proportion of task vehicles with different vehicle densities.

tasks on different vehicle densities. When the proportion of task vehicles is greater than 90%, IDM, DVNM, and AASCR tend to stabilize, and OWV tends to decrease continuously. In Figures 6(a) and 6(b), the proportion of successful tasks for OVM is getting lower and lower. This is due to the increase in the proportion of task vehicles; there are not enough computation resources in OVM for completing all tasks. In Figure 6(a), we can see that the AASCR is better than other methods when the proportion of task vehicles is greater than 25%. In Figure 6(b), we can see that the IDM is relatively better than other schemes. This is due to the IDM being the vehicular network model only with V2I communication, the computing resources are provided by RSUs and we assume that RSUs have enough resources to execute tasks. Therefore, as the quantity of task vehicles increases, the proportion of successful tasks will remain relatively high. However, other methods consider the computing resources possessed by the vehicles, so the increase in vehicle density will affect the proportion of successful tasks. It can be seen from Figure 6(b) that our scheme AASCR is very close to the results of IDM and even exceeds IDM in some proportion of task vehicles. Therefore, our scheme AASCR has a good performance in terms of the proportion of successful tasks compared to other methods.

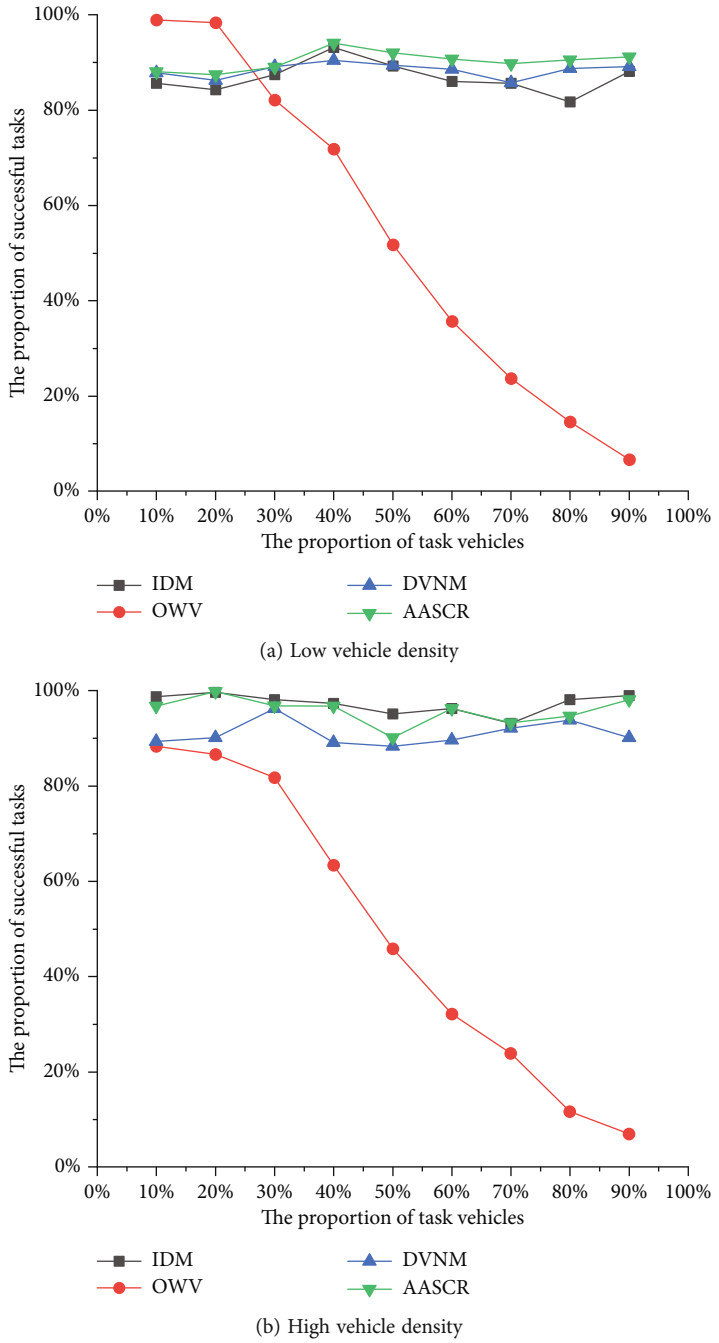


Figure 6: The proportion of successful tasks for the proportion of task vehicles with different vehicle densities.

To compare the performance for IDM, OWV, DVNM, and AASCR in terms of resource utilization rate, we conduct the relative experiments on the different proportions of task vehicles and low vehicle density with dataset Data 2.

In Figure 7, the OWV, DVNM, and AASCR outperform IDM in terms of the resource utilization rate of vehicles for all cases. When the proportion of task vehicles is greater than 90%, the results of all methods tend to stabilize. In addition, the value for OVM is 100% when the proportion of task vehicles. This is because OVM only utilizes the computation resources in vehicles. Therefore, the OVM performs better than other models in terms of the resource utilization rate of vehicles. In summary, OWV, DVNM, and AASCR make full advantage of the computation resources in vehicles in comparison with IDM. The results of AASCR are better than DVNM on the resource utilization rate of vehicles.

In conclusion, the AASCR outperforms IDM, OWV, and DVNM in terms of the average response time of tasks on different datasets, and the AASCR outperforms IDM and DVNM in terms of the average response time of tasks on different vehicle densities. The AASCR also outperforms DVNM in the proportion of successful tasks, and it has higher resource utilization rate of vehicles compared with IDM and DVNM.

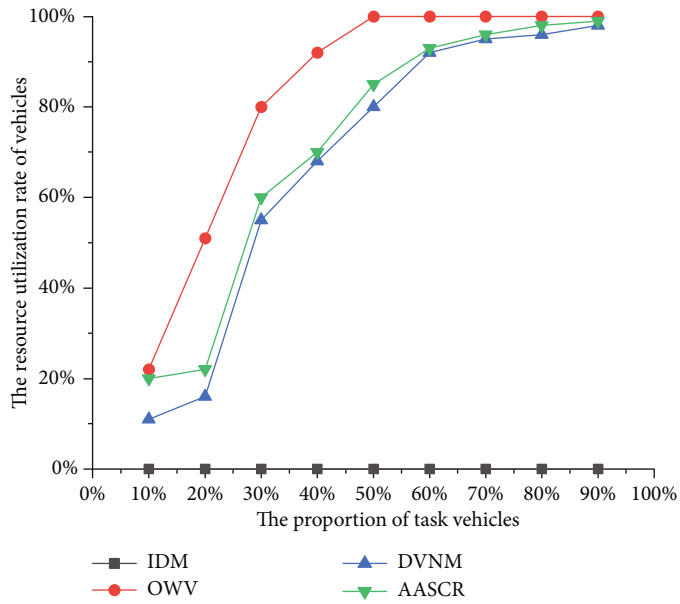


Figure 7: The resource utilization rate of vehicles in different proportions of task vehicles.

The OWV has the lower average response time than other schemes when the proportion of task vehicles is greater than 70%, and it also has better performance than other schemes on the resource utilization rate of vehicles. However, the proportion of successful tasks of OWV is much less than other schemes. Therefore, with trading off and comprehensively considering various evaluation indicators, we can see that our scheme has better performance than others.

4. Conclusions

In this paper, we focused on the allocation problem of computing resources in IoV and proposed an efficient scheme AASCR to address with computing resources from multiple service providers. The AASCR can effectively schedule computing resources under a dynamic environment. With the proposed VMM, the behavior and trajectory of vehicles are effectively analyzed and predicted, and the subnetworks of computing resources from vehicles are constructed reasonably. Considering the average response time of the vehicular task and the resource utilization rate of vehicles, the multiobjective optimization algorithm leads to a better scheme of resource allocation and enables to improve resource utilization while reducing latency. We conducted experiments in consideration of various factors and analyzed the results for a better trade-off. Based on the comprehensive analysis, our scheme outperforms recent relative schemes and has better performance.

In the future, we will consider the allocation of computing resources under more complex road conditions, the task segmentation, and the switching of resource equipment under the IoV.

Data Availability

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

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Foundation of Beijing (No. L201002)and the MIIT of China 2020 (Identification Resources Search System for Industrial Internet of Things).

References

[1] E. Ahmed and H. Gharavi, "Cooperative vehicular networking: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 996–1014, 2018.

[2] S. Raza, S. Wang, M. Ahmed, and M. Anwar, "A survey on vehicular edge computing: architecture, applications, technical issues, and future directions," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 3159762, 19 pages, 2019.

[3] A. Thakur and R. Malekian, "Fog computing for detecting vehicular congestion, an internet of vehicles based approach: a review," *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 2, pp. 8–16, 2019.

[4] X. L. Xu, H. Y. Li, W. J. Xu, Z. Liu, L. Yao, and F. Dai, "Artificial intelligence for edge service optimization in internet of vehicles: a survey," *Tsinghua Science and Technology*, vol. 27, no. 2, pp. 270–287, 2022.

[5] Y. X. Sun, X. Y. Guo, J. H. Song et al., "Adaptive learning-based task offloading for vehicular edge computing systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3061–3074, 2019.

[6] J. Zhou, F. Wu, K. Zhang, Y. Mao, and S. Leng, "Joint optimization of offloading and resource allocation in vehicular networks with mobile edge computing," in *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*, Hangzhou, China, 2018.

[7] J. Du, F. R. Yu, X. Chu, J. Feng, and G. Lu, "Computation offloading and resource allocation in vehicular networks based on dual-side cost minimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1079–1092, 2019.

[8] Y. J. Liu, S. G. Wang, J. Huang, and F. Yang, "A computation offloading algorithm based on game theory for vehicular edge networks," in *2018 IEEE International Conference on Communications (ICC)*, Kansas City, MO, USA, 2018.

[9] K. Zhang, Y. M. Mao, S. P. Leng, S. Maharjan, and Y. Zhang, "Optimal delay constrained offloading for vehicular edge computing networks," in *2017 IEEE International Conference on Communications (ICC)*, Paris, France, 2017.

[10] L. Liang, H. Peng, G. Y. Li, and X. Shen, "Vehicular communications: a network layer perspective," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1064–1078, 2019.

[11] J. Zhang and K. B. Letaief, "Mobile edge intelligence and computing for the internet of vehicles," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 246–261, 2020.

[12] X. W. Hou, Z. Y. Ren, J. J. Wang et al., "Reliable computation offloading for edge-computing-enabled software-defined IoV," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7097–7111, 2020.

[13] S. C. Hung, X. Zhang, A. Festag, K. C. Chen, and G. Fettweis, "Vehicle-centric network association in heterogeneous vehicle-to-vehicle networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5981–5996, 2019.

[14] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software defined networking-based vehicular adhoc network with fog computing," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 1202–1207, Ottawa, ON, Canada, 2015.

[15] C. Tang, X. Wei, C. Zhu, Y. Wang, and W. Jia, "Mobile vehicles as fog nodes for latency optimization in smart cities," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 9364–9375, 2020.

[16] S. Raza, W. Liu, M. Ahmed et al., "An efficient task offloading scheme in vehicular edge computing," *Journal of Cloud Computing*, vol. 9, no. 1, p. 28, 2020.

[17] T. Ye, X. Lin, J. Wu, G. Li, and J. Li, "Processing capability and QoE driven optimized computation offloading scheme in vehicular fog based F-RAN," *World Wide Web*, vol. 23, no. 4, pp. 2547–2565, 2020.

[18] C. Ma, J. Zhu, M. Liu, H. Zhao, N. Liu, and X. Zou, "Parking edge computing: parked-vehicle-assisted task offloading for urban VANETs," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 9344–9358, 2021.

[19] X. Huang, L. He, X. Chen, L. Wang, and F. Li, "Revenue and energy efficiency-driven delay constrained computing task offloading and resource allocation in a vehicular edge computing network: a deep reinforcement learning approach," 2020, http://arxiv.org/abs/2010.08119.

[20] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: human trajectory prediction in crowded spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 961–971, Las Vegas, NV, USA, 2016.

[21] M. Abdulla, E. Steinmetz, and H. Wymeersch, "Vehicle-to-vehicle communications with urban intersection path loss models," in *2016 IEEE Globecom Workshops (GC Wkshps)*, Washington, DC, USA, 2016.

[22] T. Taleb and K. B. Letaief, "A cooperative diversity based handoff management scheme," *IEEE Transactions on Wireless Communications*, vol. 9, no. 4, pp. 1462–1471, 2010.

[23] Y. Wu, J. Wu, L. Chen, G. Zhou, and J. Yan, "Fog computing model and efficient algorithms for directional vehicle mobility in vehicular network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 2599–2614, 2021.