Collaborative Task Offloading in Vehicular Edge Computing Networks

Geng Sun*, Jiayun Zhang*, Zemin Sun*‡, Long He*, and Jiahui Li*
*College of Computer Science and Technology, Jilin University, Changchun 130012, China
E-mail: sungeng@jlu.edu.cn, laurasun166@gmail.com
‡Corresponding author: Zemin Sun

***Abstract*—With the explosive increasing of various vehicle applications, vehicular edge computing (VEC) networks have emerged as a promising technology to extend the capability of the traditional vehicular networks. The computing tasks generated by the vehicles are mainly completed based on the cooperation of local and mobile edges in the past, which may not be able to meet the increasing computing resource requirements of the tasks. In this study, we propose a VEC architecture that groups vehicles with idle computing resources into a vehicular cloud (VC) to process tasks generated by vehicles. In order to reduce the delay and energy consumption of the VEC system, we formulate an optimization problem to find the optimal task assignment decision. Since the problem is NP-hard, two joint task offloading decision algorithms are proposed to solve the problem. Simulation results show that the algorithm is able to effectively reduce the latency and energy consumption of the system. Moreover, it is also demonstrated that the VC can be effectively constructed to further improve the utility of the system.**

***Index Terms*—Vehicular edge computing, task offloading, vehicular cloud, cluster formation, relay selection.**

I. Introduction

Vehicular network system have attracted much attention in both academia and industry. With the rapid development of artificial intelligence and the internet of vehicles, intelligent vehicles have long become synonymous with the times. The emergence of intelligent vehicles has also led to the emergence of various vehicular applications, and these applications typically generate various latency-sensitive computing tasks, such as monitoring the position of vehicles on the road and analyzing the traffic information. These tasks are not only applied to one vehicle, but also the whole vehicular network, and the traditional task processing method is not enough to support the high timeliness of the applications [1]. As the number of vehicles increases, interactions between vehicles become more frequent, and more and more vehicles choose to cooperate to deal with tasks. The way of cooperation and the choice of cooperative vehicles have been mentioned in many studies. On this basis, a challenge facing the vehicle network system is to optimize the timeliness of the task requesting vehicles to achieve better system performance.

In order to reduce the delay and energy consumption of task offloading in vehicular edge computing (VEC) networks, various task offloading methods and computing resource allocation decisions are proposed. In order to reduce the load pressure of task offloading, a load balancing scheme for task offloading based on software defined network technology is proposed in [2]. Furthermore, Zhao et al. [3] propose a collaborative offloading approach that integrates the capabilities of mobile edge computing (MEC) and vehicular cloud (VC) computing via vehicle-to-infrastructure (V2I) and vehicle-to-vehicle (V2V) communications. For the collaborative offloading, VC construction and relay path finding are essential for efficient VC offloading decisions. First, the vehicle that is out of the communication of the VC may ask for assistance from other vehicles to relay the tasks to the VC. Although relay nodes selection for task transmitting has been studied by a lot of early works [4], [5], it is challenging to select the optimal relay path due to the mobility of both vehicles and VC as well as the dynamic of vehicular networks. In addition, dynamic collaborative VC is difficult to construct due to the different mobility patterns of vehicles. The cooperation between VC and MEC for task offloading are considered by several studies [2], [3], [6]. However, the VC construction method in [6] keeps static with the movement of vehicles, leading to the reprocession of the uncompleted tasks. Although the dynamic of VC construction is considered by [3], [7], the local offloading strategies are ignored.

In this work, a collaborative task offloading approach is studied by jointly considering the local, MEC, and VC offloading strategies. The main contributions of this paper are summarized as follows:

- We propose a collaborative task offloading approach by integrating the local, MEC and VC strategies [8] to minimize the delay and energy consumption of the vehicular system.
- We construct a VC formation method for collaborative task assignment by employing the K-means clustering method to classify vehicles and select the VC leader. Moreover, the optimal offloading decision is obtained by using the Nelder-Mead (NM) algorithm and differential evolution (DE) algorithm.
- Simulation results show that the collaborative offloading approach can effectively improve the performance of the system in terms of offloading delay and energy consumption.

The rest of this paper is organized as follows. Section II presents the Models and preliminaries. Section III formulates the offloading problem. Section IV shows the the methods to the problem. Section V gives the simulation results. Finally,

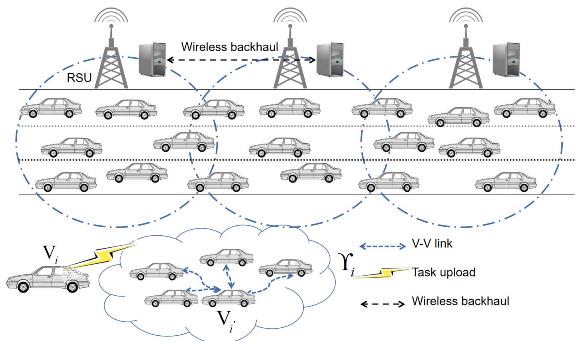

Fig. 1. The system model.

this paper is summarized in Section VI.

II. Models and Preliminaries

A. System Model

As shown in Fig. 1, there are three traffic flows on a one-way road and a set of vehicles $\mathcal{V} = \{1, 2, \ldots, V\}$ moving on the road. The characteristics of V_i is defined as $\{L_i, v_i, \varphi_i, F_i\}$, where L_i denotes the location of V_i, v_i denotes the speed of V_i, φ_i denotes the task of V_i release, and F_i denotes the computing power of V_i. The characteristics of task φ_i is defined as $\{S_i, t_i, C_i\}$, where S_i means the size of the task, t_i means the maximum acceptable delay, and C_i means the amount of CPU resources required to perform the task. $\mathcal{M} = \{1, 2, \ldots, M\}$ represents the set of MEC server equipped on the road side unit (RSU) deployed alongside the road. The characteristics of MEC server M_j is defined as $\{f_j, L_j^{mec}, R_j^{mec}\}$, where f_j is the computing power of M_j, L_j^{mec} is the location of M_j and it includes $\{x_i^{mec}, y_i^{mec}\}$, and R_j^{mec} is the communication range of M_j. For convenience, the symbols and their meanings are listed in Table. I.

The requesting vehicle can choose to process the task locally, offload it to the MEC server via V2I communication, or offload it to the VC [9] via V2V communication. The VC is formed by a group of vehicles with idle resources, and we use a variable $\theta_i = \{\theta_{i,-1}, \theta_{i,0}, \theta_{i,1}, \ldots, \theta_{i,j_{max}}\}$ ($\sum\limits_{j=-1}^{j_{max}} \theta_{i,j} = 1$) to determine the offloading decision of V_i, where $\theta_{i,0}$, $\theta_{i,j} (1 \leq j \leq j_{max})$, and $\theta_{i,-1}$ means that the task is processed locally, offloaded to MEC server M_j, and offloaded to VC Υ_i, respectively. It should be noted that vehicle i' is selected as the leader of VC Υ_i to assign the task to the other members in the VC.

B. Communication Model

The communication rate between sender i and receiver j can be expressed as:

$$R_{i,j} = B_{i,j} \log_2 \left(1 + \gamma_{i,j}\right), \tag{1}$$

where $B_{i,j}$ denotes the channel bandwidth between the sender and receiver, and $\gamma_{i,j}$ denotes the signal to interference plus noise ratio (SINR) of the communication. This is the famous Shannon theorem, and other vehicles may use the same spectrum as sender V_i to transmit information to receiver, therefore $\gamma_{i,j}$ can be expressed as:

$$\gamma_{i,j} = \frac{\frac{P_i H_{i,j} c^2}{(4\pi G_{i,j})^2 d_{i,j}^{\tau}}}{\alpha k_{i,j} z_{i,j} B_{i,j} + \sum\limits_{k \in \mathcal{V}, k \neq i} \frac{P_k H_{k,j} c^2}{(4\pi G_{k,j})^2 d_{k,j}^{\tau}}}, \tag{2}$$

where P_i denotes the transmit power of sender V_i, $H_{i,j}$ is the channel gain between sender and receiver, $G_{i,j}$ is the carrier frequency of the sender, $d_{i,j}$ is the distance between the sender and receiver, and c is the speed of light. Furthermore, $\alpha, k_{i,j} z_{i,j} B_{i,j}$ represents the noise power, where $z_{i,j}$ is the receiver noise figure, $k_{i,j}$ is the reference receiver temperature in degrees Kelvin, α is the Boltzmann constant, and τ is the path loss coefficient. Besides, $\sum\limits_{k \in \mathcal{V}, k \neq i} \frac{P_k H_{k,j} c^2}{(4\pi G_{k,j})^2 d_{k,j}^{\tau}}$ denotes the interference induced by the transmission of others.

The signal experiences attenuation over the distance of wireless transmission. The signal can be correctly received by the receiver only if the received SINR at the receiver exceeds the threshold of the SINR, i.e., $\gamma_{i,j} \leq \gamma_0$. Therefore, the maximum communication range between V_i and V_j can be obtained from Eq. (2) as:

$$d_{i,j}^{max} = \left(\frac{P_i H_{i,j} c^2}{(4\pi G_{i,j})^2 \gamma_0 \left(\alpha k_{i,j} z_{i,j} B_{i,j} + \sum\limits_{k \in \mathcal{V}, k \neq i} \frac{P_k H_{k,j} c^2}{(4\pi G_{k,j})^2 d_{k,j}^{\tau}} \right)} \right)^{\frac{1}{\tau}}. \tag{3}$$

In the following sections, $R_{i,j}^{v2i}$ and $R_{i,k}^{v2v}$ are used to represent the communication rates of V2I and V2V communications, respectively.

C. Delay Model

The total delay of task offloading consists of the offload transmission delay, processing delay, and download transmission delay. The download delay is not considered in this work since the result of the task is much smaller. The total offloading delay can be represented as:

$$T_i = \begin{cases} T_{i,i}^{loc}, \ \theta_{i,0} = 1, \\ T_{i,j}^{tran} + T_{i,j}^{exe}, \ \theta_{i,j} = 1, \ 1 \leq j \leq j_{max}, \\ T_{i,\Upsilon_i}^{tran} + T_{i,\Upsilon_i}^{exe}, \ \theta_{i,-1} = 1. \end{cases}, \tag{4}$$

where $\theta_{i,0} = 1$, $\theta_{i,j} = 1$, and $\theta_{i,-1} = 1$ denote that the task is processed locally, offloaded to MEC server M_j, and offloaded to VC Υ_i, respectively.

1) Local Offloading: The delay for local offloading (i.e., $\theta_{i,0} = 1$) can be expressed as $T_{i,i}^{loc} = \frac{C_i}{F_i}$.

2) MEC Offloading: For MEC offloading, the task of vehicle V_i is first offloaded to the MEC server on RSU-m^i that the vehicle can directly connects wirelessly. Then, the task is transmitted to the target MEC server M_j via wireless backhaul if M_i is not the target MEC server. Therefore, the total transmission delay can be given as $T_{i,j}^{tran} = \frac{S_i}{R_{i,m^i}^{v2i}} + S_i T_{m^i,j}^{backhaul}$, where $T_{m^i,j}^{backhaul}$ means the transmission delay from RSU-m^i to RSU-j. In addition, the processing delay of M_j to execute the task can be given as $T_{i,j}^{exe} = \frac{C_i}{f_j}$.

3) VC Offloading: Similarly, the transmission delay for VC offloading (i.e., $\theta_{i,-1} = 1$) includes the transmission delay from the requesting vehicle V_i to VC leader vehicle $V_{i'}$, and the maximum task assignment delay from the leader vehicle $V_{i'}$ to each member vehicle in VC [7], which can be expressed as $T_{i,\Upsilon_i}^{tran} = \frac{S_i}{R_{i,i'}^{v2v}} + \max\limits_{V_k \in \Upsilon_i} \left\{ \frac{S_{i,k}}{R_{i',k}^{v2v}} \right\}$, where V_k is the set of vehicular members in VC Υ_i, $S_{i,k}$ is the task size that the leader vehicle assigns to the member vehicle k in VC, and $\frac{S_{i,k}}{R_{i',k}}$ is the task assignment delay from the leader vehicle to the member vehicles.

The processing delay of the task in VC Υ_i can be expressed as $T_{i,\Upsilon_i}^{exe} = \max\limits_{V_k \in \Upsilon_i} \left\{ \frac{C_{i,k}}{F_k} \right\}$, where $C_{i,k}$ denotes the CPU resource required for computing the task assigned to VC member V_k, and F_k denotes the computing capability of V_k. It should be noted that the tasks that assigned to the VC members can be processed in parallel [9].

Based on above discussions, the delay for task offloading can be finally expressed as:

$$\begin{aligned} T_i = \max_{1 \le j \le j_{max}} \{ & \frac{\theta_{i,0} C_i}{F_i}, \frac{\theta_{i,j} S_i}{R_{i,m^i}^{v2i}} + \theta_{i,j} S_i T_{m^i,j}^{backhaul} + \frac{\theta_{i,j} C_i}{f_j}, \\ & \frac{\theta_{i,-1} S_i}{R_{i,i*\prime}^{v2v}} + \max_{V_k \in \Upsilon_i} \left\{ \frac{\theta_{i,-1} S_{i,k}}{R_{i',k}^{v2v}} \right\} \\ & + \max_{V_k \in \Upsilon_i} \left\{ \frac{\theta_{i,-1} C_{i,k}}{F_k} \right\} \}. \end{aligned} \tag{5}$$

D. Energy Consumption Model

Similar to Eq. (4), the energy consumption for task offloading can be expressed as:

$$E_i = \begin{cases} E_{i,i}^{loc}, \ \theta_{i,0} = 1, \\ E_{i,\Upsilon_i}^{tran} + E_{i,\Upsilon_i}^{exe}, \ \theta_{i,-1} = 1, \\ E_{i,j}^{tran} + E_{i,j}^{exe}, \ \theta_{i,j} = 1, \ 1 \le j \le j_{max}. \end{cases} \tag{6}$$

1) Energy Consumption of Local Offloading: The energy consumption for local offloading (i.e., $\theta_{i,0} = 1$) can be expressed as $E_{i,i}^{loc} = \xi_i F_i^{\chi_i} T_{i,i}^{loc} = \xi_i F_i^{\chi_i - 1} C_i$, where $\xi_i \ge 0$ and χ_i are V_i CPU-dependent parameters.

2) MEC Offloading Energy Consumption: The energy consumption for offloading the task to the MEC server mainly incurred by task transmission and task execution, it can be expressed as $E_{i,j}^{tran} = \frac{P_i S_i}{R_{i,m^i}^{v2i}} + S_i E_{m^i,j}^{backhaul}$, where RSU-$m^i$ denotes the RSU that vehicle V_i can directly connect, and $E_{m^i,j}^{backhaul}$ denotes the energy consumption of the transmission from RSU-m^i to RSU-j.

The energy consumption of M_j to execute the task can be expressed as $E_{i,j}^{exe} = \xi_j^{mec} f_j^{\chi_j^{mec}} T_{i,j}^{exe} = \xi_j^{mec} f_j^{\chi_j^{mec} - 1} C_i$, where ξ_j^{mec} and χ_j^{mec} are CPU parameters of MEC server M_j.

3) VC Offloading Energy Consumption: Similarly, the transmission energy consumption of vehicle to transmit the task to VC Υ_i can be expressed as $E_{i,\Upsilon_i}^{tran} = \frac{P_i S_i}{R_{i,i'}^{v2v}} + \sum\limits_{V_k \in \Upsilon_i} \frac{P_{i'} S_{i,k}}{R_{i',k}^{v2v}}$, where $V_{i'}$ denotes the leader vehicle of VC Υ_i, $\sum\limits_{k \in \Upsilon_i} \frac{P_{i'} S_{i,k}}{R_{i',k}^{v2v}}$ denotes the total transmission energy consumption in Υ_i, and V_k denotes the vehicle in Υ_i. The processing energy consumption of Υ_i can be expressed as $E_{i,\Upsilon_i}^{exe} = \sum\limits_{V_k \in \Upsilon_i} \xi_k F_k^{\chi_k} T_{i,\Upsilon_i}^{exe} = \sum\limits_{V_k \in \Upsilon_i} \xi_k F_k^{\chi_k - 1} C_{i,k}$.

According to above discussions, the energy consumption of offloading φ_i can be finally expressed as:

$$\begin{aligned} E_i = & \xi_i \theta_{i,0} F_i^{\chi_i - 1} C_i + \sum_{1 \le j \le j_{max}} \{ \frac{\theta_{i,j} P_i C_i}{R_{i,m^i}^{v2i}} + \theta_{i,j} S_i E_{m^i,j}^{backhaul} \\ & + \xi_j^{mec} \theta_{i,j} f_j^{\chi_j^{mec} - 1} S_i \} + \frac{\theta_{i,-1} P_i S_i}{R_{i,i'}^{v2v}} \\ & + \sum_{V_k \in \Upsilon_i} \frac{\theta_{i,-1} P_{i'} S_{i,k}}{R_{i',k}^{v2v}} + \sum_{V_k \in \Upsilon_i} \xi_k \theta_{i,-1} F_k^{\chi_k - 1} C_{i,k}. \end{aligned} \tag{7}$$

TABLE I
MAIN NOTATION

Notation	Description
$\mathcal{V}$	set of vehicles
$\mathcal{M}$	set of MEC
$\{L_i, v_i, \varphi_i, F_i\}$	set of vehicle-i's characteristics
$\{S_i, t_i, C_i\}$	set of task-i's characteristics
$\{f_j, L_j^{mec}, R_j^{mec}\}$	set of MEC-j's characteristics
$d_{i,j}$	the distance of vehicle-i and vehicle-j
θ_i	vehicle-i's offloading decision
T_i	the total time of task-i
E_i	the total energy of task-i
Υ_i	vehicles set of VC-i
$V_{i'}$	VC leader in Υ_i

III. Problem Formulation

The objective of this work is to minimize the task offloading overheads on delay and energy consumption by guiding the vehicle to select the optimal offloading strategy. Thus, two optimization problems should be considered as follows.

Problem 1: Minimize the total time required for all vehicular tasks, which can be formulated as follows:

$$\textbf{P1:} \min_{\{\theta_{i,j},S_{i,k},C_{i,k}\}} \sum_{i\in\mathcal{V}} T_i\left(\theta_{i,j},S_{i,k},C_{i,k}\right) \tag{8}$$

$$\text{s.t. } C1: T_i \le t_i,\ \forall i\in\mathcal{V} \tag{8a}$$

$$C2: \sum_{-1\le j\le j_{max}} \theta_{i,j}=1,\ i\in\mathcal{V} \tag{8b}$$

$$C3: \sum_{k\in\Upsilon_i} C_{i,k}=\theta_{i,-1}C_i \tag{8c}$$

$$C4: C_{i,k},S_{i,k}>0,\ \forall V_k\in\Upsilon_i \tag{8d}$$

$$C5: \sum_{k\in\Upsilon_i} S_{i,k}=\theta_{i,-1}S_i \tag{8e}$$

Problem 2: Minimize the energy consumption generated by all vehicle tasks, which can be formulated as follows:

$$\textbf{P2:} \min_{\{\theta_{i,j},S_{i,k},C_{i,k}\}} \sum_{i\in\mathcal{V}} E_i\left(\theta_{i,j},S_{i,k},C_{i,k}\right) \tag{9}$$
$$\text{s.t.} \quad \text{(8a)} \sim \text{(8e)}$$

IV. Algorithm

A. Vehicular Cloud Formation

If the vehicle within the communication range of RSU-m releases a task φ_0, it will send a request to RSU-m. After RSU-m receives the message, RSU-m obtains the information of all vehicles within its communication range, including speed, location and the remaining computing resources. Furthermore, the K-means clustering method is used to construct multiple vehicle clusters for VC formation according to the following steps. First, for each cluster, the vehicle with the highest reference variable is selected as the leader of the cluster. Then, the vehicles that directly connect to the leader are selected as the VC candidates. Finally, the only VC with the highest average similarity dependency [10] is choosed to process task φ_0.

The following similarities are considered in the K-means method in this work.

1) Location Similarity: The location difference among vehicles is considered as a criteria for vehicle clustering since the task transmission among vehicles with similar location causes lower overheads on transmission delay and energy consumption. The location similarity between V_i and V_k within the communication range of RSU-j can be given as:

$$sim_{i,k}^{LS}=1-\frac{dis_{i,k}}{\max\limits_{p,q\in\mathcal{G}_j} dis_{p,q}}\in[0,1], \tag{10}$$

where $dis_{i,k}$ denotes the Euclidean distance between vehicles V_i and V_k.

2) Speed Similarity: Relative consistent speeds between vehicles lead to more stable connection link for V2V communication and relative stable vehicular cluster. Therefore, the similarity of speed can be calculated as:

$$sim_{i,k}^{SS}=\frac{\min\{v_i,v_k\}}{\max\{v_i,v_k\}}\in[0,1]. \tag{11}$$

Algorithm 1: K-means-based VC formation and VC leader selection

```
   Input: The set $s$ of all vehicles excepts $V_i$ within the
          range of RSU-$m^i$ that can be communicated
          from $V_i$
   Output: $k$ VC leader candidates and $k$ clusters
1  Initialize: Randomly select $k$ vehicles from $S$ to join
     the set $s_k$, name each vehicle in $s_k$ as $ve_1,\dots,ve_k$,
     and create a corresponding cluster $se_1,\dots,se_k$ for
     them;
2  while $s_k$ is not convergent do
3      foreach $V_p \in s \cap \overline{s_k}$ do
4          if $sim_{p,k'} = \max\{sim_{p,ve_1},\dots,sim_{p,ve_k}\}$
             then
5              add $V_p$ to $se_{k'}$;
6          end
7      end
8      for $p = 1$ to k do
9          Calculate the average similarity of each vehicle
             to other vehicles in $se_p$;
10         Choose the vehicle with the maximum average
             similarity as new $ve_p$;
11         Delete vehicles in $se_p$ besides new $ve_p$;
12     end
13     Delete vehicles in $s_k$ and add new vehicles
         $ve_1,\dots,ve_k$ to $s_k$;
14 end
15 foreach $V_p \in s \cap \overline{s_k}$ do
16     if $sim_{p,k'} = \max\{sim_{p,ve_1},\dots,sim_{p,ve_k}\}$ then
17         add $V_p$ to $se_{k'}$;
18     end
19 end
20 for $p = 1$ to k do
21     foreach $V_q \in se_p$ do
22         Obtain the directly communicable vehicle set $\mathcal{Q}$
             of vehicle $V_q$ in cluster $se_p$ through SINR and
             calculate the average similarity $\overline{sim_q}$ by
             $\frac{\sum_{V_{q'}\in\mathcal{Q}} sim_{q,q'}}{\mathcal{Q}.size()}$;
23     end
24     if $\overline{sim_{p'}} = \max_{q\in se_p}\{\overline{sim_q}\}$ then
25         Let $V_{p'}$ become a candidate for the leader of
             VC $\Upsilon_i$;
26     end
27 end
```

3) Direction Similarity: The cosine similarity is applied to measure the similarity of the speed direction, which can be expressed as:

$$sim_{i,k}^{DS}=\begin{cases}\cos_{i,k},\ \cos_{i,k}>0,\\ 0,\ \cos_{i,k}\le 0,\end{cases}\in[0,1], \tag{12}$$

where $\cos_{i,k}$ denotes the cosine of the angle between vehicles

Algorithm 2: Optimal DE algorithm for task allocation decision

```
Input: Evolutionary algebra $G$, mutation operator $S$, the number of decision individuals in the population $N$, the number of genes in the population $M$, and objective function $F$
Output: Optimal decision vector $D$
1  Initialize: Initialize random decision population $X$;
2  Standardize every individual in the population;
3  Calculate the corresponding objective function value $obx = \{obx_1, \ldots, obx_N\}$ of each decision-making individual in $X$;
4  $D_0$ is the dicision corresponed by $\min\{obx\}$;
5  for $i = 1$ to $G$ do
6      $V = X$;
7      for $j = 1$ to $N$ do
8          Randomly select three individuals $r_1, r_2, r_3$ in the population except $i$ decision-making individuals;
9          Update the $i$-th decision in $V$ to $r_1 + S \times (r_2 - r_3)$ in $X$;
10     end
11     for $j = 1$ to $M$ do
12         $X$ and $V$ cross with random number $k$ as probability;
13     end
14     Calculate the corresponding objective function value $obv = \{obv_1, \ldots, obv_N\}$ of each decision-making individual in $V$;
15     for $j = 1$ to $N$ do
16         if $obv_j < obx_j$ then
17             $x_j = v_j$;
18         end
19     end
20     Update $obx$ with the new $X$;
21     $D_i$ is the dicision corresponed by $obx$;
22 end
23 $D$ is the dicision corresponed by $\min\{D_0, \ldots, D_G\}$;
```

Algorithm 3: Optimal NM algorithm for task allocation decision

```
Input: Maximum number of iterations $G$, objective function $F$, number of variables in a decision $N$
Output: Optimal decision vector $D$
1  Initialize: Initialize a random decision $X$;
2  Standardize $X$;
3  $S$=eye($N$);
4  for $i = 1$ to $N - 1$ do
5      $abc = [X; X + S(i,:); X + S(i+1,:)]$;
6      for $j = 1$ to $G$ do
7          Sort the $abc$ row matrix according to the size of its $F(abc)$;
8          $m = (abc(1) + abc(2))/2$;
9          $n = 3m - 2c$;
10         if $F(n) < F(abc(2))$ then
11             $abc(3) = n$
12         end
13         else
14             $r = (m + n)/2$;
15             if $F(r) < F(abc(3))$ then
16                 $abc(3) = r$;
17             end
18             else
19                 $s = (c + m)/2$;
20                 if $F(s) < F(abc(3))$ then
21                     $abc(3) = s$;
22                 end
23                 else
24                     $abc(2) = m$;
25                     $abc(3) = (abc(1) + abc(2))/2$;
26                 end
27             end
28         end
29     end
30     $X = abc(1)$;
31 end
32 $D = X$;
```

v_i and v_k, which can be given as:

$$\cos_{i,k} = \frac{v_i \cdot v_k}{\mid v_i \mid \times \mid v_k \mid} \in [-1, 1] \,. \tag{13}$$

Taking the above mentioned three metrics into account, the similarity between vehicles V_i and V_k can be evaluated as:

$$sim_{i,k} = (sim_{i,k}^{LS})^{\beta_1} \cdot (sim_{i,k}^{SS})^{\beta_2} \cdot (sim_{i,k}^{DS})^{\beta_3}, \tag{14}$$

where parameters β_1, β_2, and β_3 denote the weights of the three metrics. Targeting at maximizing $sim_{i,k}$ of a cluster, Algorithm 1 describes the process of VC formation and VC leader selection.

B. V2V Communication Path Selection

After constructing the VC, it is essential to find the shortest path from the requesting vehicle to the VC leader. In order to know whether vehicles V_a and V_b can communicate directly, the maximum communication distance of V_a and V_b can be obtained by Eq. (3). This work considers a weighted graph among vehicles for the shortest path finding, where the link weight between nodes V_a and V_b communicated directly can be given as the can be quantified by the transmission delay, i.e., $W_{a,b} = T_{a,b}^{tran} = \frac{S_i}{R_{a,b}^{v2v}}$. The shortest path algorithm can undoubtedly find the shortest path from the requesting vehicle to the VC leader.

C. Optimal Solution of Task Allocation

In this subsection, the methods of DE and NM are employed for obtaining the optimal offloading strategy. The optimal decision is obtained by the evolutionary iteration from the random initial decision through DE algorithm and NM algorithm [11], respectively, to obtain the optimal decisions. As a population-based heuristic global optimization algorithm, DE algorithm can quickly obtain the global optimal solution. For the NM algorithm, it is a simple graphical convergence algorithm with fast convergence speed and is suitable for most problems. The details of these two algorithms are described in Algorithms 2 and 3, respectively.

TABLE II
SIMULATION PARAMETERS

Symbol	Meaning	Default value
R_j^{mec}	the communication range of MEC-j	20-50 m
t_i	the maximum time of task-i	0.8-1.5 s
B	bandwidth	10-20 MHz
P_i	transmission power of vehicle-i	-85-44.8 dBm
H_i	channel gain of vehicle-i	1.25-1.75
c	speed of light	3.0×10^8 m/s
G_i	carrier frequency of sender	5.9×10^9 GHz
τ	path loss coefficient	2.5
α	Boltzmann constant	1.38×10^{-23}
k_i	reference receiver temperature	250-350 K
z_i	receiver noise figure	5.5-6.5 dB
ξ	CPU parameters 1	7.8^{-21} [12]
χ	CPU parameters 2	2 [12]

V. SIMULATION RESULTS

In the simulation, a 500 m one-way three-lane road is constructed, and 10 RSUs are deployed alongside the road. The task size is set as [400 KB, 1000 KB], the maximum delay allowed by the task is set as [0.5 s, 1 s], the computational resource of the each vehicle is set as [0.8 Mbps, 1.6 Mbps], and the computational resource of each MEC server is set as [1.5 Mbps, 4.5 Mbps]. Other parameters used in the simulation are listed in Table. II.

A. Performance Of VC Construction Scheme

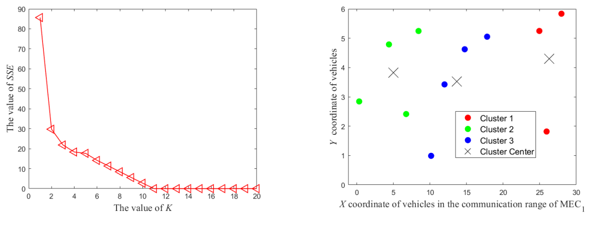

Fig. 2. (a) SEE image varying with K value, (b) VC construction by k means cluster algorithm.

Fig. 2(b) shows the example of VC construction of the vehicles within the communication range of RSU 1. The symbol '$\times$' represents the clustering center, and all vehicles are clustered into three clusters. The value K of the number of clusters can be obtained by the elbow method, which further determines the optimal K value through the sum of squared error (SSE) [13] value for clusters. The determination of K value can be observed in Fig. 2(a), where the curvature of the SSE value curve reaches to the largest at $K = 3$. After obtaining the K value, the cluster of the vehicles within the coverage of RSU 1 can be formed by Algorithm 1. The simulation result shows that the execution time of K-means clustering algorithm is minimal, because of the limited number of vehicles in a RSU. Based on the above simulation, the VC leader and VC members can be selected from the clusters.

B. Comparison Results of Offloading Scheme

The algorithms used in this study (i.e., DE and NM) are compared with the following three benchmark offloading strategies i.e., local offloading, MEC offloading, and VC offloading.

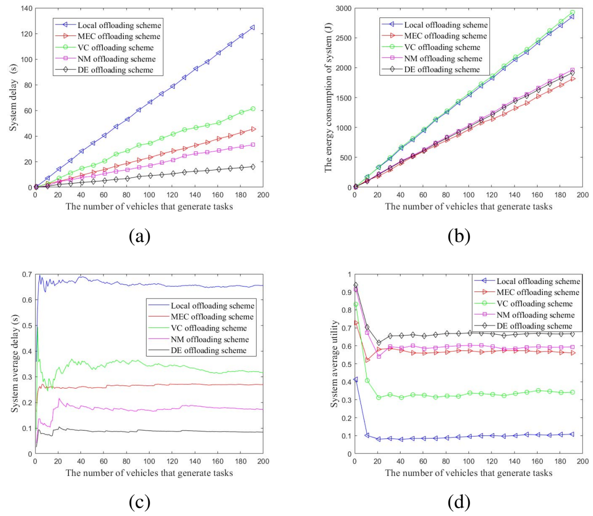

Fig. 3. (a) The effect of the number of requesting vehicles on system delay, (b) The effect of the number of requesting vehicles on system energy consumption, (c) The effect of the number of requesting vehicles on system energy consumption, (d) The effect of the number of requesting vehicles on system average utility.

Fig. 3(a) shows effect of the number of requesting vehicles on system delay. First, it can be observed from the figure that the total delay of the system gradually increases with the increasing requesting vehicles. Obviously, this is because the offloading workload of the system becomes heavier with the increasing number of tasks, leading to increased overheads on system delay. Furthermore, it can be seen that DE used by this study achieves lower delay compared to the benchmark approaches. The reason is that DE can dynamically decide the optimal decision by jointly considering the local and MEC offloading strategies. Besides, DE achieves the lowest system delay among the other schemes. This is because DE fully exploits the system resources by collaboratively integrating the local offloading, MEC offloading, and VC offloading strategies. The results in Fig. 3(a) demonstrate that DE can adapt to the varying requesting vehicles and achieves lower system delay for task offloading.

Fig. 3(b) shows the effect of the number of requesting vehicles on system energy consumption. First, it can be observed that the total energy consumption of the system for task offloading gradually increase with the increase of the requesting vehicles. This is obvious because the task workload becomes heavier with the increasing number of tasks. Furthermore, it can be seen that MEC, NM, and DE offloading schemes cause approximately similar energy consumption, which is lower than that of the local and VC offloading schemes. This is because the local and VC offloading schemes consider to process all of the tasks locally on the CPUs of vehicles, which causes higher energy consumption due to the limited computational capabilities of vehicles. Besides, it can be observed that NM and DE employed by this work is sightly higher than the MEC offloading scheme. This is reasonable because both NM and DE consider the collaborative offloading strategies by integrating the capabilities of vehicles and MEC servers.

Fig. 3(c) illustrates the effect of the number of requesting vehicles on average system delay. First, it can be observed that the average delay of the system shows fluctuations and tends to be stable gradually. The task average processing time is uncertain when there are few vehicles because of the randomness of the task size and the heterogeneity among the computing capabilities of MEC servers and vehicles. Furthermore, it can be observed that when the number of requesting vehicles excesses 20, the average system delay tends to be stable, which is because the size of tasks and the computing power of vehicles and MECs are limited. Besides, it can be observed that DE achieves the lowest average system delay among the other schemes, which is benefited from the efficient utilization of the system resources through collaborative offloading strategies. In conclusion, the set of results demonstrate that the proposed approach is able to achieves superior average system delay by dynamically adjusting the offloading strategies to the available resource of the system.

Fig. 3(d) shows the effect of the number of requesting vehicles on average system utility. First, it can be observed the average system utility drops significantly at first and then stabilizes with the increase of the requesting vehicles. This is because the system computation resources are sufficient to process the tasks when there are few vehicles. Furthermore, it can be seen that DE and NM proposed by this work achieve higher system utility compared to the benchmark schemes due to the joint offloading strategies. Specifically, DE obtains the significant superior system utility among the other schemes. This is because DE makes use of the system resources by collaboratively integrating local, MEC, and VC strategies for efficient task offloading. Consequently, the approach proposed by this study can improve the system utility by adaptive the offloading strategies to the increasing task workload.

VI. Conclusion

This study proposes a collaborative task offloading approach by jointly considering the local, MEC and VC strategies. A collaborative task allocation approach is proposed by integrating the strategies of local, MEC and VC offloading. The K-means clustering scheme is employed for optimal VC construction, and the DE and NM methods are applied for deciding the optimal offloading strategies. Besides, experimental results demonstrate that the proposed scheme achieves superior system performances than the benchmark offloading schemes in terms of delay and energy consumption.

Acknowledgment

This study is supported in part by the National Natural Science Foundation of China (62002133, 61872158, 61806083).

References

[1] M. Dai, Z. Su, Q. Xu, and N. Zhang, "Vehicle assisted computing offloading for unmanned aerial vehicles in smart city," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1932–1944, 2021.

[2] J. Zhang, H. Guo, J. Liu, and Y. Zhang, "Task offloading in vehicular edge computing networks: A load-balancing solution," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 2092–2104, 2020.

[3] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7944–7956, 2019.

[4] Q. Chen, H. Li, B. Yang, and R. Chai, "A utility based relay vehicle selection algorithm for vanet," in *2012 International Conference on Wireless Communications and Signal Processing (WCSP)*, 2012, pp. 1–6.

[5] R. Chai, Y. Lv, B. Yang, and Q. Chen, "Cooperative game based relay vehicle selection algorithm for vanets," in *2014 14th International Symposium on Communications and Information Technologies (ISCIT)*, 2014, pp. 30–34.

[6] F. Zeng, Q. Chen, L. Meng, and J. Wu, "Volunteer assisted collaborative offloading and resource allocation in vehicular edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3247–3257, 2021.

[7] S. Fei, H. Fen, C. Nan, W. Miao, Z. Haibo, G. Lin, and S. Xuemin, "Cooperative task scheduling for computation offloading in vehicular cloud," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 11 049–11 061, 2018.

[8] C. Lin, G. Han, X. Qi, M. Guizani, and L. Shu, "A distributed mobile fog computing scheme for mobile delay-sensitive applications in sdn-enabled vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5481–5493, 2020.

[9] J. Xie, Y. Jia, Z. Chen, Z. Nan, and L. Liang, "Efficient task completion for parallel offloading in vehicular fog computing," *China Communications*, vol. 16, no. 11, pp. 42–55, 2019.

[10] H. Wu, F. Lyu, and X. Shen, "Optimal uav caching and trajectory design in the agvn," in *Mobile Edge Caching in Heterogeneous Vehicular Networks*. Springer, 2022, pp. 61–88.

[11] H. Xia, "An improved wolf search algorithm introduced by nelder-mead operator," in *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, 2018, pp. 11–14.

[12] G. Zhang, W. Zhang, Y. Cao, D. Li, and L. Wang, "Energy-delay tradeoff for dynamic offloading in mobile-edge computing system with energy harvesting devices," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4642–4655, 2018.

[13] D. Marutho, S. Hendra Handaka, E. Wijaya, and Muljono, "The determination of cluster number at k-mean using elbow method and purity evaluation on headline news," in *2018 International Seminar on Application for Technology of Information and Communication*, 2018, pp. 533–538.