Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8823173, 15 pages
https://doi.org/10.1155/2020/8823173

Research Article

Joint Task Offloading and Resource Allocation Strategy for DiffServ in Vehicular Cloud System

Ya Kang, Zhanjun Liu, Qianbin Chen, and Yingdi Dai

School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China

Correspondence should be addressed to Ya Kang; 417184514@qq.com

Received 25 July 2020; Revised 3 October 2020; Accepted 9 October 2020; Published 2 November 2020

Academic Editor: Hongwei Wang

A vehicular cloud (VC) can reduce latency and improve resource utilization of the Internet of vehicles by effectively using the underutilized computing resources of nearby vehicles. Although the task offloading of the VC enhances road safety and traffic management on the Internet of vehicles and meets the low-latency requirements for driving safety services on the Internet of vehicles business, there are still some key challenges such as the resource allocation mechanism of differentiated services (DiffServ) and task offloading mechanism of improving user experience. To address these issues, we study the task offloading and resource allocation strategy of the VC system where tasks generated by vehicles can be offloaded and executed cooperatively by vehicles in VC. Specifically, the computing task is further divided into independent subtasks and executed in different vehicles in VC to maximize the offloading utility. Considering the mobility of vehicles, the deadline of tasks, and the limited computing resources, we propose the optimization problem of task offloading in the VC system in the cause of improved user experience. To characterize the difference in service requirements resulting from the diversity of tasks, a DiffServ model focusing on the pricing of a task is utilized. The initial pricing of a task is tailored by the characteristics of the task and the uniqueness of the network status. In this model, tasks are sorted and processed in order according to task pricing, so as to optimize resource allocation. Numerical results show that the proposed scheme can effectively increase the resource utilization and task completion ratio.

1. Introduction

In a vehicle network [1], the cloud composed of vehicles with strong computing power is called VC [2]. The task offloading of the VC [3] has a significant impact on the performance of the Internet of vehicles [4], which enhances vehicle road safety and traffic management by effectively utilizing the underutilized computation resources of nearby vehicles and reduces traffic congestion and accidents, thus improving the network performance of the Internet of vehicles. The processing of emerging message-type safety applications in the vehicle network requires a large amount of computing resources, such as 4 K/8 K ultrahigh real-time traffic video streaming, multiview video stitching processing of autonomous vehicles, and AR-assisted applications related to safety warnings. However, the collaborative scheduling of information and data by offloading computing tasks to the vehicular cloud can effectively enhance the utilization of computing resources due to the limited computing resources of vehicles. At the same time, due to its proximity to users and rich computing resources, the vehicular cloud can meet the low delay requirements for safe driving on the Internet of vehicles business, which is a key technology in the existing Internet of vehicles.

Owing to the fast-moving characteristics of vehicles, computing resources of vehicles are dynamic. For example, the connection of mobile vehicles can be connected or disconnected at any time, which will bring about unstable computing resource availability. Accordingly, in order to ensure the offloading of the task, the task should be offloaded within vehicle contact time [5]. In vehicular cloud computing, task offloading is inseparable from efficient resource allocation technology. Resources involved in the entire process of offloading the task of the vehicle to the vehicular cloud are mainly divided into the communication resources and the computing resources [6]. [7–9]. For the task offloading of

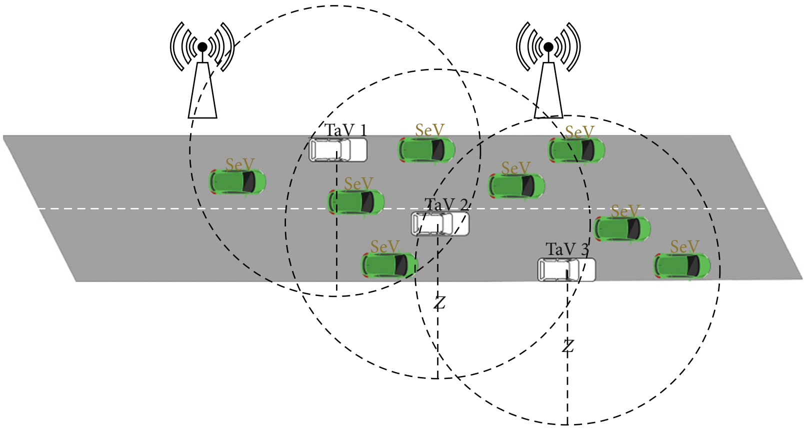


Figure 1: Vehicular cloud system model.

VC, Wu et al. assume that each vehicle task can be divided into several independent subtasks, and then, the task allocation problem is solved [10]. Under the condition of ensuring user service experience, Liu et al. study how to allocate limited computing resources to maximize the economic benefits of MEC service providers when vehicles offload tasks to MEC servers [11]. In order to solve the users' selfishness, Liu et al. have established a kind of new type of computing sharing market in the vehicle network [12]. Several other works, such as [13, 14], use the method of task replication to allow a task to be executed by multiple vehicles so as to maximize the possibility of completing the task before a given deadline and improve the reliability.

In recent years, game theory has become an incentive mechanism to analyze and allocate resources. A game model can effectively capture the interaction between users and servers [15], so game theory has become a way to analyze and allocate resources [16]. For the sake of creating greater social and economic benefits in the process of resource management, a pricing strategy as an effective way of network resource management on the basis of game theory is proposed, in [16]. The forementioned research believes that the pricing mechanism is an effective way to improve resource utilization. However, how to use the pricing mechanism to distinguish different services in the VC system in game theory has not been well explored. It also can be seen from the above literatures that most of the resource allocation process VC system does not DiffServ. That is to say, the relationship between default services is equal. Unlike the resource management [17] scheme that considers the priority of a task in the general VC computing scenario, we propose a game theory-based DiffServ resource allocation mechanism that DiffServ through a task's differentiated pricing. In other words, the higher the initial pricing of the task, the higher the service requirements of the business. The initial pricing of a task mainly involves two factors, namely, the urgency of the task vehicle and the network condition around the task. Finally, we maximize the offloading utility of tasks in the VC system and consider the mobility of vehicles and meet task delay constraints to maximize the completion of tasks. The main contributions of our paper are summarized as follows.

(i) To meet the requirements of the diversity of tasks while effectively using computing resources, we propose a resource allocation mechanism for differentiated services and offload each task according to the pricing order

(ii) To improve the completion ratio of whole tasks in the vehicular cloud system, a task offloading mechanism that maximizes the offloading utility is proposed and considers the mobility of vehicles and the deadline of tasks

(iii) To address the problem of task offloading and resource allocation, a distributed resource allocation algorithm and Lagrange dual method are used, respectively

2. System Model

We consider a dynamic traffic scenario in which the vehicle can be either a service requester or a service provider in a finite two-way straight road. Let $i \in (1, 2, \cdots, M)$ be the group of M TaVs (task vehicles). We use three parameters $(d_i, c_i, T_i^{\text{deadline}})$ to indicate the generated task of TaV i, by which d_i (in bits) specifies the data size of the task, where $c_i = \kappa d_i$, c_i (in CPU cycles per bit) is the computing resource required to processing the task, κ is the mapping between CPU cycles and task size, and T_i^{deadline} is the deadline for task execution. When the task processing delay exceeds the deadline, the task processing fails. f_{i0} indicates the local computing ability of TaV i. At the same time, $j \in (i, 2, \cdots, N)$ is denoted as a group of vehicles with free resources, which is called SeVs (service vehicles), where the computing resource of SeV j is defined as $F_j^{\max}$. Figure 1 depicts a simple example of a vehicular cloud system. Here, TaV 1, TaV 2, and TaV 3 form a vehicular cloud with SeVs within their covering radius, respectively. It is assumed that the communication radius of each vehicle is Z.

2.1. Task Offloading Model. Since the size of the output task is usually much smaller than the size of the input task, we can ignore the return time in offloading [18, 19]. Each TaV first offloads the computing task to one or more surrounding

SeVs for processing and then handles the size of the remaining task by itself to ensure that the task is completed within the deadline. The total offloading delay for SeV j to process the size of a task assigned by TaV i is given as

$$T_{ij} = \frac{d_{ij}}{R_{ij}} + \frac{c_{ij}}{f_{ij}}, \tag{1}$$

where d_{ij} is the task size of the TaV i offloading to SeV j, R_{ij} is the transmission rate of TaV i to SeV j, c_{ij} is the required computation resource to complete task d_{ij}, and f_{ij} is the computing resource provided by SeV j to TaV i.

2.2. Local Processing Model. When the task is not completely offloaded, the size of the remaining task is processed locally, and the local processing delay is expressed as follows:

$$T_{i0} = \frac{c_i - \sum_{j=1}^{N} c_{ij}}{f_{i0}}. \tag{2}$$

2.3. Mobility Model. Assuming that the initial position of TaV i and the initial position of SeV j is, respectively, $(x_{\mathrm{TaV}_i}, y_{TaV_i})$ and (x_{SeV_j}, y_{SeV_j}), the speed of each vehicle is defined as $v_{\mathrm{TaV}_i}, v_{\mathrm{SeV}_j} \in [v_{\min}, v_{\max}]$ that the vehicle maintains a uniform linear motion during task processing, so the motion angle $\theta = \{0, \pi\}$, after moving the time slot Δt, the position of the TaV i is changed to $x_{\mathrm{TaV}_i}^{\Delta t} = x_{\mathrm{TaV}_i} + v_{\mathrm{TaV}_i}\Delta t \cos\theta$, $y_{\mathrm{TaV}_i}^{\Delta t} = y_{\mathrm{TaV}_i} + v_{\mathrm{TaV}_i}\Delta t \sin\theta$. The position of SeV j is changed to $x_{\mathrm{SeV}_j}^{\Delta t} = x_{\mathrm{SeV}_j} + v_{\mathrm{SeV}_j}\Delta t \cos\theta$, $y_{\mathrm{SeV}_j}^{\Delta t} = y_{\mathrm{SeV}_j} + v_{\mathrm{SeV}_j}\Delta t \sin\theta$. According to Euclid's theorem, after TaV i and SeV j move through the time slot Δt, the distance between the two vehicles is

$$s = \sqrt{\left(x_{\mathrm{TaV}_i}^{\Delta t} - x_{\mathrm{SeV}_j}^{\Delta t}\right)^2 + \left(y_{\mathrm{TaV}_i}^{\Delta t} - y_{\mathrm{SeV}_j}^{\Delta t}\right)^2}. \tag{3}$$

Communication between each other can be achieved only when SeV j is within the communication radius Z of TaV i. The connection time between vehicles is predicted as follows:

$$Z^2 = \left(x_{\mathrm{TaV}_i}^{T_{ij}^{\mathrm{re}}} - x_{\mathrm{SeV}_j}^{T_{ij}^{\mathrm{re}}}\right)^2 + \left(y_{\mathrm{TaV}_i}^{T_{ij}^{\mathrm{re}}} - y_{\mathrm{SeV}_j}^{T_{ij}^{\mathrm{re}}}\right)^2. \tag{4}$$

2.4. Offloading Utility Function. Considering all the TaVs in the VC system, the total offloading utility obtained of offloading TaVs to SeVs can be defined as

$$U = \sum_{i=1}^{M} (u_i - C_i), \tag{5}$$

where u_i is denoted as the offloading revenue of TaV i and C_i is the offload cost of the TaV i.

The total offloading revenue obtained by TaV i for offloading tasks to the VC is expressed as follows [20]:

$$u_i = \alpha_i \sum_{j=1}^{N} \log_2\left(1 + d_{ij}\right), \tag{6}$$

where α_i represents the task offloading revenue weight parameter of TaV j, and its value is greater than zero.

The transmission cost required by TaV i to offload tasks to the VC is expressed as follows:

$$C_i = \chi \sum_{j=1}^{N} d_{ij}, \tag{7}$$

where χ is the unit transmission cost required for transmiting 1 Mb data to any SeV.

With the aim of maximizing the task offloading of TaVs in the VC system, the following optimization is formulated P1:

$$\begin{aligned} &\max_{d_{ij}} U \\ \text{s.t.}\quad & \mathrm{C1}: T_{ij} \le T_i^{\mathrm{deadline}}, \forall i \in M, \forall j \in N \\ & \mathrm{C2}: \sum_{j=1}^{N} d_{ij} \le d_i, \forall i \in M, \forall j \in N \\ & \mathrm{C3}: d_{ij} \ge 0, \forall i \in M, \forall j \in N \\ & \mathrm{C4}: T_{ij} \le T_{ij}^{\mathrm{re}}, \forall i \in M, \forall j \in N. \end{aligned} \tag{8}$$

C1 means that the processing delay of each subtask is limited by the deadline of the task. C2 and C3 ensure that the size of total tasks offloading is not greater than the generated task size and the size of offloading is positive. C4 indicates that TaV i and SeV j should not spend more time on task processing than the connection time between the two vehicles.

3. Computing Resource Allocation Based on Game Theory

For the sake of enabling SeVs to share their own resources to others, we propose an incentive method which is based on game theory as well as introduce a paid resource allocation strategy. The offloading delay for TaV i to offload the task to SeV j in the process of task offloading is as follows:

$$T_{ij} = \frac{d_i}{R_{ij}} + \frac{c_i}{f_{ij}}. \tag{9}$$

With the purpose of simplifying the problem, we will consider R_{ij} as a fixed value. According to (9), the delay for task offloading is affected by computing resources provided by SeV j to TaV i. It is impossible for SeV j to provide computing resources to TaV i for free because of the selfishness of vehicles. According to messages exchanged between vehicles, we design a distributed incentive mechanism for resource allocation. The specific details are shown as follows.

3.1. Utility and Cost Functions

(1) With the aim of maximizing utility of SeV j, the following optimization problem is formulated:

$$\begin{aligned} \max_{f_{ij}} \quad & V_j = \delta_j \ln\left(1 + p_i f_{ij}\right) - cc_j f_{ij} \\ \text{s.t.} \quad & \text{C1} : f_{ij} > 0 \\ & \text{C2} : f_{ij} \le F_j^{\max}, \end{aligned} \tag{10}$$

where cc_j represents the unit resource cost price of SeV j; δ_j is the willingness of SeV j, with $\delta_j \ge 1$; p_i represents the bid for TaV i; C1 indicates that the resources provided by SeV j are larger than zero; and C2 indicates that the total resources provided by SeV j are not larger than the maximum resources provided by itself.

Theorem 1. *For any price given by each TaV, there must be a unique Nash equilibrium solution in the noncooperative game between SeVs.*

Proof. The first-order derivative of V_j with respect to f_{ij} is expressed as follows:

$$\frac{\partial V_j}{\partial f_{ij}} = \frac{\delta_j p_i}{1 + p_i f_{ij}} - cc_j. \tag{11}$$

The second-order derivative of V_j with respect to f_{ij} is expressed as follows:

$$\frac{\partial^2 V_j}{\partial {f_{ij}}^2} = -\frac{\delta_j p_i}{\left(1 + p_i f_{ij}\right)^2}. \tag{12}$$

As the values of all parameters in (12) are positive, the second-order derivative of utility function is lower than zero, namely, $\partial^2 V_j / \partial {f_{ij}}^2 < 0$. Therefore, the utility function of SeV j is a concave function with maximum value. The corresponding optimal solution $f_{ij} = \delta_j / cc_j - 1/p_i$ can be obtained by $\partial V_j / \partial f_{ij} = 0$, and there is a unique Nash equilibrium in the noncooperative game between SeVs given the price of the TaV.

(2) In order to maximize economic benefits, we should minimize offloading cost. With the aim of minimizing cost function of TaV i, the following optimization problem is formulated:

$$\begin{aligned} \min_{f_{ij}} \quad & p_i \sum_{j=1}^{N} f_{ij} \\ \text{s.t.} \quad & \text{C1} : f_{ij} \ge 0 \\ & \text{C2} : f_i^{\min} < \sum_{j=1}^{N} f_{ij} \le f_i^{\max} \\ & \text{C3} : P_{\min} \le p_i \le P_{\max}, \end{aligned} \tag{13}$$

where C2 means that the total computing resources provided by SeVs are larger than the minimum required resources $f_i^{\min}$, but it is smaller than the maximum required resources $f_i^{\max}$. C3 indicates that the bids of TaVs are larger than the lowest price $P_{\min}$, but it is smaller than the highest price $P_{\max}$.

3.2. TaV Pricing Rules. This section defines the concept of "pricing" for tasks by their service requirements, for the purpose of solving the problem of resource allocation. The pricing function is used to classify tasks requested by TaVs and further distinguish the priority of tasks. The definition of the pricing function of the TaVs task mainly involves two factors, namely, the urgency of the task and the network conditions around the TaV. Accordingly, the pricing of the TaV is obtained by the urgency of the task and the network status around the TaV. Then, the urgency of the task is related to the size of the task generated by the TaV and the deadline of the task. The network condition around the TaV is related to the number of requests received and the number of answers received in the previous slot.

3.2.1. Distribution of the Number of Vehicles within the TaV Coverage. Owing to the stability of the network, the network condition of the previous and the next moment is unchanged. Suppose that TaV i and SeV j are randomly distributed on the road. Then, it is known that the TaV $k (k = i)$ receives R_k requests and S_k responses at the previous time. According to the conditional probability, the distribution function of the number of TaVs m_k and the distribution function of the number of SeVs n_k in the coverage area is obtained.

(a) Distribution function of the number of TaVs m_k within the coverage of TaV k is expressed as follows:

$$\begin{aligned} P\left(\frac{m_k}{R_k}\right) = \frac{P(m_k, R_k)}{P(R_k)} &= \frac{P(m_k, R_k)}{\sum_{m_k=R_k}^{\infty} P(m_k, R_k)} \\ &= \frac{C_{m_k}^{R_k} {P_1}^{R_k} (1 - P_1)^{m_k - R_k}}{\sum_{m_k=R_k}^{\infty} C_{m_k}^{R_k} {P_1}^{R_k} (1 - P_1)^{m_k - R_k}}, \end{aligned} \tag{14}$$

where (m_k, R_k) indicates that there are totally m_k TaVs within the coverage of TaV k, among which TaV k receives R_k requests, and P_1 indicates the probability of any TaV sending out requests.

(b) Distribution function of the number of SeVs n_k within the coverage of TaV k is expressed as follows:

$$P\left(\frac{n_k}{S_k}\right)=\frac{P(n_k,S_k)}{P(S_k)}=\frac{P(n_k,S_k)}{\sum_{n_k=S_k}^{\infty}P(n_k,S_k)}=\frac{C_{n_k}^{S_k}P_2^{S_k}(1-P_2)^{n_k-S_k}}{\sum_{n_k=S_k}^{\infty}C_{n_k}^{S_k}P_2^{S_k}(1-P_2)^{n_k-S_k}}, \tag{15}$$

where (n_k, S_k) means there are a total of n_k SeVs within the coverage of TaV k, among which TaV k receives S_k responses, and P_2 means the probability that any SeV will respond to a TaV after receiving it.

3.2.2. Average Number of Vehicles within the Coverage of TaV k

(a) The average number of TaVs within the coverage of TaV k is expressed as follows:

$$E(m_k)=\sum_{m_k=R_k}^{\infty}m_kP\left(\frac{m_k}{R_k}\right) \tag{16}$$

(b) The average number of SeVs within the coverage of TaV k is expressed as follows:

$$E(n_k)=\sum_{n_k=S_k}^{\infty}n_kP\left(\frac{n_k}{S_k}\right) \tag{17}$$

3.2.3. Remaining Resources within the Coverage of TaV k. Suppose that the resource requirements of each TaV obey the normal distribution of $Q_1 \sim (\mu_1, \delta_1^2)$ and the resources provided by each SeV obey the normal distribution of $Q_2 \sim (\mu_2, \delta_2^2)$, the remaining resources around TaV k obtained by (16) and (17) are expressed as follows:

$$Q_k = E(n_k)Q_2 - E(m_k)Q_1. \tag{18}$$

On account of the independence of Q_1 and Q_2, we can obtain the mean and variance value of Q_k:

$$E(Q_k)=E(n_k)\mu_2-E(m_k)\mu_1, \tag{19}$$

$$D(Q_k)=E(n_k)^2\delta_2^2-E(m_k)^2\delta_1^2. \tag{20}$$

According to (19) and (20), we can get the distribution of the remaining resources Q_k. In other words, a new normal distribution is expressed as follows:

$$F(Q_k)=\frac{1}{\sqrt{2\pi D(Q_k)}}\int_{-\infty}^{Q_k}e^{-\left((t-E(Q_k))^2/2D(Q_k)\right)}. \tag{21}$$

3.2.4. Initial Pricing of TaV k. According to (21), the remaining resources covered by TaV k are expressed as follows:

$$E(Q_k)=E(n_k)\mu_2-E(m_k)\mu_1. \tag{22}$$

The initial pricing of TaV k [21] is

$$P_k=\begin{cases}\lambda_k(P_{\text{mid}}-P_{\text{min}})+P_{\text{min}}, & E(Q_k)\ge 0,\\ -\lambda_k(P_{\text{max}}-P_{\text{mid}})+P_{\text{mid}}, & E(Q_k)<0,\end{cases} \tag{23}$$

$$\lambda_k=\frac{2}{\pi}\arctan\left(\frac{f_k^{\text{min}}}{E(Q_k)}\right), \tag{24}$$

where f_k^{min} means the minimum resources obtained by TaV k , $f_k^{\text{min}} = d_k/T_k^{\text{deadline}}$. From the pricing function, we can see that when the surrounding remaining resources are greater than zero, the price will be raised by the lowest price; otherwise, the price will be raised by the middle price.

After getting the best initial pricing, we develop a distributed resource allocation algorithm. The algorithm dynamically changes the price of the TaV to ensure the maximum utility of the SeV, and the TaV can obtain resources to meet its needs. The algorithm is introduced in detail in Algorithm 1. Firstly, the TaV sends the business request (including the price); meanwhile, all vehicles within the coverage of the TaV will receive the request message. Then, the SeV that received the request message maximizes its utility to obtain the size of computing resources that can be provided (Steps 4-14 of Algorithm 1). Next, the SeV sends a response (resources that the SeV can be provided) to the TaV; meanwhile, the TaV receives the response and adds up resources provided by SeVs. Finally, if the sum of the total resources provided is greater than the maximum resource demand of the TaV, the price will be reduced; otherwise, the price will be increased if the total resources provided are less than the minimum resource demand of the TaV. Otherwise, the price is the optimal price, and the optimal resource allocation result of the SeV is obtained (Steps 15-30 of Algorithm 1).

4. Task Allocation of Vehicular Cloud System

This section is aimed at solving the problem of task allocation of the VC system. Because the second-derivative of U for d_{ij} in equation (5) can be obtained $\partial^2 U/\partial {d_{ij}}^2 < 0$, the objective function of optimization problem P1 is the concave function of variable d_{ij}. Since the constraints C1, C2, C3, and C4 in P1 are all convex sets, the optimization problem P1 is a convex optimization problem. In the end, the optimal d_{ij} is obtained by Lagrange theory.

```
1:  Initialization: The cost price $cc_j$ of the SeV $j$ and the maximum resource it can provide are $\mathrm{F}_j^{\max}$; the TaV $i$ the required computing
    resource is $f_i^{\min} < f_i \le f_i^{\max}$
2:  For t=0, TaV $i$ to set the price $p_i(t)$, and send it to the vehicles within the coverage
3:  for $t = 1, 2, 3.\cdots$ do
4:    for $i = 1, 2, \cdots, M$ do
5:      for $j = 1, 2, \cdots, N$ do
6:        According to $\partial V/\partial f_{ij} = 0$, compute $f_{ij}$
7:        for $k = 1 : i$ do
8:          $a = a + f_{kj}$
9:        end for
10:        If $a > F_j^{\max}$ then
11:          $f_{ij} = 0$
12:        end if
13      end for
14:    end for
15:    If $\sum_{j=1}^{N} f_{ij} \ge f_i^{\max}$ then
16:      decrease the price, $p_i(t+1) = p_i(t) - v$
17:    else
18:      if $\sum_{j=1}^{N} f_{ij} \le f_i^{\min}$ the
19:        increase the price, $p_i(t+1) = p_i(t) + v$
20:      else
21:        $p_i(t+1) = p_i(t)$
22:      end if
23:      If $p_i(t+1) == p_i(t)$ then
24:        break
25:      else
26:        report the new price $p_i(t+1)$
27:        back to 3
28:      end if
29:    end if
30:  end for
```

Algorithm 1: Distributed resource allocation algorithm.

The Lagrangian function of optimization problem P1 is defined as follows (without considering $d_{ij} \ge 0$):

$$L\left(d_{ij}, \varpi, \sigma, \psi\right) = U + \sum_{i=1}^{M} \sum_{j=1}^{N} \varpi_{ij} \left[T_i^{\text{deadline}} - \frac{d_{ij}}{R_{ij}} - \frac{\kappa d_{ij}}{f_{ij}} \right] + \sum_{i=1}^{M} \sigma_i \left(d_i - \sum_{j=1}^{N} d_{ij} \right) + \sum_{i=1}^{M} \sum_{j=1}^{N} \psi_{ij} \left(T_{ij}^{\text{re}} - \frac{d_{ij}}{R_{ij}} - \frac{\kappa d_{ij}}{f_{ij}} \right), \quad (25)$$

where ϖ, σ, ψ are the Lagrangian multipliers.

According to the original optimization problem P1, a Lagrangian dual problem can be obtained as follows:

$$\text{P2}: \min_{\varpi_{ij}, \delta_i, \psi_{ij}} \max_{d_{ij}} L\left(d_{ij}, \varpi, \delta, \psi\right). \quad (26)$$

For the optimization problem P2, it can be decomposed into a maximum task assignment problem and a minimum Lagrangian factor update problem, which can then be solved through iteration. The maximized part of the optimization problem P2 is expressed as follows:

$$\text{P21}: \max_{d_{ij}} L(\varpi, \sigma, \psi). \quad (27)$$

Since the optimization problem P1 is a convex optimization problem, according to the KKT condition, the optimal solution of the optimization problem P21 is equal to the optimal solution of the optimization problem P1. By solving equation $\partial L(\varpi, \sigma, \psi)/\partial d_{ij} = 0$ and considering constraint $d_{ij} \ge 0$, the optimal task assignment d_{ij} is obtained as follows:

$$d_{ij} = \begin{cases} 0, & d_{ij} \le 0, \\ \dfrac{\alpha}{\ln 2\left(\chi + \left(1/R_{ij} + k/f_{ij}\right)\left(\varpi_{ij} + \psi_{ij}\right) + \delta_i\right)} - 1, & d_{ij} > 0. \end{cases} \quad (28)$$

```
Require:$\alpha$, $d_i$, $\chi$
Ensure:$d_{ij}$, $\forall d$
1:  Define $t$ for counting iteration, initialize $t = 0$, define $\varepsilon$ as convergence threshold
2:  Initialization $\varpi_{ij}, \sigma_j, \psi_{ij}$
3:  According to $\varpi_{ij}, \sigma_j, \psi_{ij}$ and (28), compute $d_{ij}$
4:  According to (25), compute $L(d_{ij}^{*(t)}, \varpi^{(t)}, \sigma^{(t)}, \psi^{(t)})$ , let $L^{(t)} = L(d_{ij}^{*(t)}, \varpi^{(t)}, \sigma^{(t)}, \psi^{(t)})$
5:  According to (30)(31)(32), compute $\varpi_{ij}^{(t+1)}$,$\sigma_j^{(t+1)}$, $\psi_{ij}^{(t+1)}$
6:  According to (25), compute $L(d_{ij}^{*(t)}, \varpi^{(t)}, \sigma^{(t)}, \psi^{(t)})$, let $L^{(t)} = L(d_{ij}^{*(t)}, \varpi^{(t+1)}, \sigma^{(t+1)}, \psi^{(t+1)})$
7:  while$|L^{(t+1)} - L^{(t)}| > \varepsilon$do
8:    According to $\varpi_{ij}^{(t+1)}$, $\sigma_j^{(t+1)}$, $\psi_{ij}^{(t+1)}$, compute $d_{ij}^{(t+1)}$
9:    $L^{(t)} = L^{(t+1)}$
10:     $d_{ij}^{*(t)} = d_{ij}^{*(t+1)}$
11:     Repeat steps 5, 6
12:     $t = t + 1$
13:     if$\sum_{j=1}^{N} d_{ij}^{*} \sim = d_i$then
14:       $d_{i0}^{*} = d_i - \sum_{j=1}^{N} d_{ij}^{*}$
15:       if$T_{i0} \leq \mathrm{T}_i^{\max}$then
16:         $d_{i0}^{*}$ and $d_{ij}^{*}$
17:       else
18:         $d_{i0}^{*} = 0$ and $d_{ij}^{*} = 0$
19:       end if
20:     else
21:       $d_{i0}^{*} = 0$ and $d_{ij}^{*}$
22:     end if
23:   end while
```

Algorithm 2: Task assignment algorithm based on Lagrange theory.

The Lagrange multipliers updating problem of P2 is expressed as

$$\mathrm{P22}: \quad \min_{\varpi_{ij}, \sigma_i, \psi_{ij}} L\left(d_{ij}, \varpi, \sigma, \psi\right). \tag{29}$$

It can be seen from the above analysis that P22 is also a convex problem, so the gradient descent method is used to update the Lagrangian multipliers $\varpi_{ij}, \sigma_i, \psi_{ij}$, and the Lagrangian multipliers $\varpi_{ij}, \sigma_i, \psi_{ij}$ are all nonnegative real numbers.

We denote $\varpi_{ij}^{(t)}, \sigma_i^{(t)}, \psi_{ij}^{(t)}$ as the Lagrangian multipliers of the tth iteration, and then, the Lagrange multipliers $\varpi_{ij}^{(t+1)}$, $\sigma_i^{(t+1)}, \psi_{ij}^{(t+1)}$ of the $(t+1)$th iteration can be obtained as follows:

$$\varpi_{ij}^{(t+1)} = \max\left\{0, \varpi_{ij}^{(t)} - \tau^{(t)}\left(T_i^{\max} - \frac{d_{ij}}{R_{ij}} - \frac{c_{ij}}{f_{ij}}\right)\right\}, \tag{30}$$

$$\psi_{ij}^{(t+1)} = \max\left\{0, \psi_{ij}^{(t)} - \tau^{(t)}\left(T_{ij}^{\mathrm{re}} - \frac{d_{ij}}{R_{ij}} - \frac{c_{ij}}{f_{ij}}\right)\right\}, \tag{31}$$

$$\sigma_i^{(t+1)} = \max\left\{0, \sigma_i^{(t)} - \tau^{(t)}\left(d_i - \sum_{j=1}^{N} d_{ij}\right)\right\}, \tag{32}$$

where $\tau^{(t)} = \tau^{(0)}/\sqrt{t}$ is the tth update step length. In order to ensure the convergence of the algorithm, this update step length [22] is considered.

The task allocation algorithm can be divided into the following four steps: (1) Initialize the Lagrange multipliers. (2) According to the given Lagrange multipliers, solve the optimization problem P21 through KKT conditions to obtain the optimal task allocation result; update the Lagrange multipliers with the method in formula (30)–(32). (3) Get the task allocation result by the convergence condition of Lagrange function. (4) Judge whether the task has been completely allocated; if it has been completely allocated, return the allocation result; otherwise, judge whether the remaining unallocated task can be executed locally. If possible, the rest of the tasks are executed locally and the task assignment results are returned. Otherwise, the task cannot be completed on time, and the task assignment will fail. In Algorithm 2, the task allocation algorithm will be described in detail.

5. Numerical Results

The numerical results given in this section show that the proposed algorithm improves the resource utilization and total task completion ratio in the VC system. We evaluate and compare the effects of different vehicle speed, vehicle density, task size, and number of TaVs on the performance of the proposed scheme and other baseline schemes. We will compare the proposed resource allocation scheme in this paper (PRAS) with two benchmark schemes, including (i) LCS (Lowest Cost Scheme): the server bids by the network conditions, and the buyer always chooses the server with the lowest unit price, and (ii) HRS (Highest Revenue Scheme): the buyer bids by resource requirements, and the server always provides services to the buyer with the highest unit bid. In addition, the proposed offloading scheme in this paper (POS) is compared with the benchmark scheme by three different comparison methods: (i) LPS (Local Processing Scheme); (ii) AOS (Average Offloading Scheme): TaVs distribute tasks to the surrounding SeVs equally; and (iii) UOS (Unified Offloading Scheme): TaVs consider its own computing resources and assign the task equally to the surrounding SeVs and itself.

In this paper, the MATLAB simulation platform is used to verify the proposed algorithm. Vehicles are randomly distributed to a two-way lane with a length of 1000 m and a width of 50 m. Furthermore, the computation capacity and task size of the vehicle follow normal distribution. Relevant simulation parameters are given in Table 1.

5.1. Effect of the Number of TaVs. Figure 2(a) shows resource utilization for different algorithms when the number of TaVs varies from 10 to 20, where the resource utilization ratio is equal to the ratio of the resources used by TaVs to the total resources of SeVs in the system. It is observed from Figure 2(a) that the resource utilization monotonically increases with the number of TaVs, and the growth rate gradually decreases; in other words, increasing the number of TaVs from 10 to 14 makes more TaVs benefit from the resources of SeVs than that from 16 to 20. The reason is that the available resources are no longer enough to respond to the requests of new TaVs with limited computation resources. As a result, the resource utilization gradually increased slowly. Furthermore, it can also be observed that PRAS can obtain higher resource utilization than HRS and LCS. When the number of TaVs is more, it can better reflect the superiority of PRAS. This is because the number of TaVs increases, and the more TaVs compete for limited computing resources, the probability of successful acquisition of resources by TaVs is reduced. For the sake of obtaining better resource utilization, compared with HRS and LCS, PRAS further optimizes the network status and DiffServ of different tasks based on resource allocation. In other words, PRAS adopts a differentiated pricing strategy. Compared with HRS which only considers network condition and LCS which only considers its own condition, PRAS considers the condition of the TaV, including whether the TaV has urgent tasks and whether the network around the TaV is busy. Therefore, the total resource utilization rate of the network is improved. In addition, when the number of TaVs is less than 12, the HRS and PRAS have the same resource utilization as the computing resources are sufficient and both two algorithms can provide the resources required for TaVs.

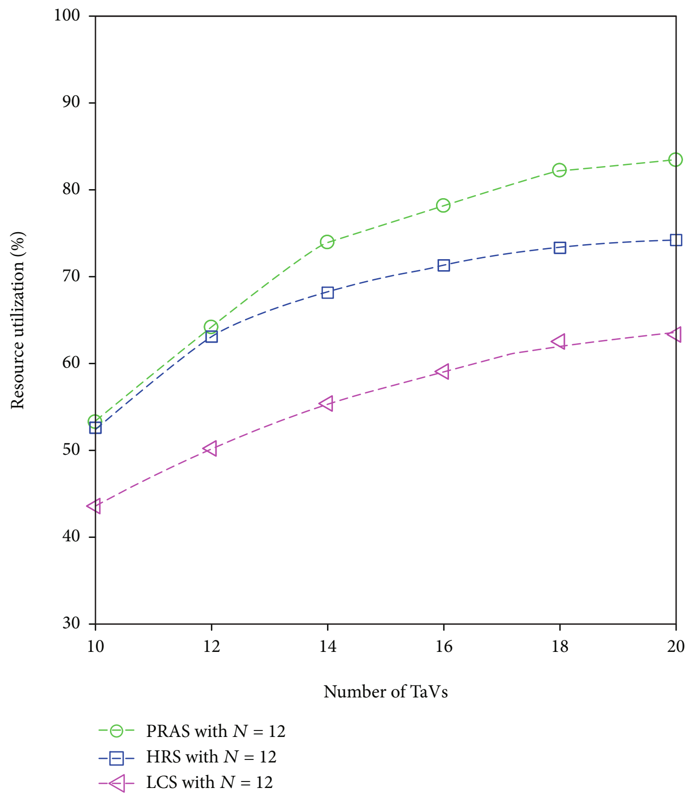
Figure 2(b) depicts the task completion ratio of different algorithms where the number of TaVs varies from 10 to 20, where the task completion ratio is equal to the number of tasks successfully completed within the deadline over the total number of tasks in the system. It can be seen from Figure 2(b) that POS can achieve a relatively higher task completion ratio compared with the three baseline schemes. This is because UOS and AOS do not consider mobility of vehicles and task deadline constraints. However, the proposed scheme jointly optimizes mobility of vehicles and the deadline of the task, which is conducive to making full use of resources. Hence, more tasks can be offloaded and task completion is significantly increased. In addition, since LPS does not consider the choice of SeVs, the increase in the number of TaVs does not affect the task completion ratio of LPS.

Table 1: Simulation parameter.

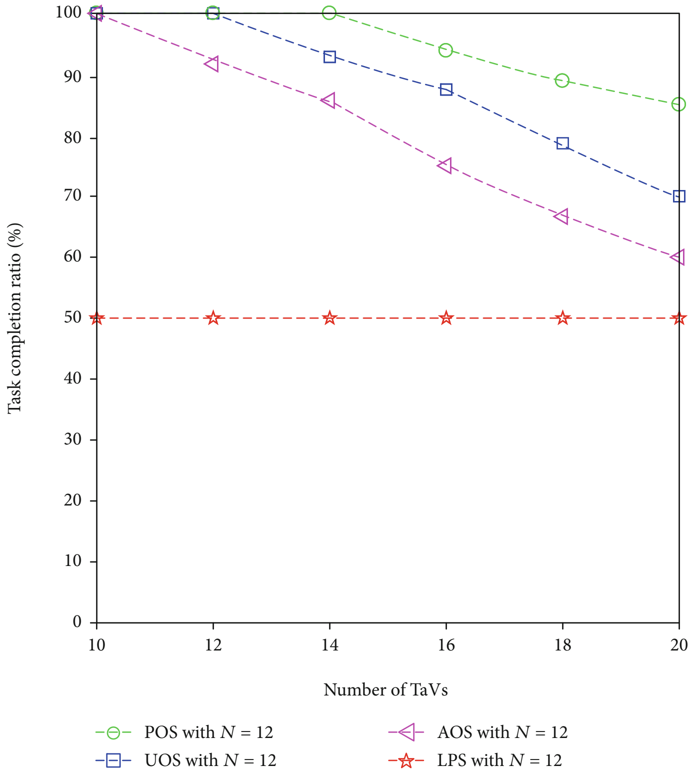
Parameter	Value
v	$[80 - 120]$ km/h
Z	300 m
R	5 Mb
$T_{\max}$	$1 \sim 3$ s
$F_j^{\max}$	$N(4 * 10^{10}, 0.5 * 10^{10})$ (CPU cycle/s)
f_i	$N(2 * 10^{10}, 1 * 10^{9})$ (CPU cycle/s)
d	$[3, 7]$ (Mb)
k [15]	18000 cycles/bit
$M + N$	$[20 - 30]$

Figure 3 shows the price for different task sizes of the TaV when the number of TaVs varies from 10 to 20. It is observed from Figure 3 that the price monotonically increases with the number of TaVs, and the price also monotonically increases with task size. The reason is that increasing the size of tasks and the number of TaVs will reduce the TaV surrounding idle resources. The price function of the TaV is a piecewise function of the surrounding remaining resources. When the remaining resources of the TaV are greater than zero, the price is increased based on the lowest price. When the remaining resources around the TaV are greater than zero, the price is increased based on the middle price.

5.2. Effect of Vehicle Density. Figure 4 shows the resource utilization and task completion ratio of the VC system versus the vehicle density. Numerical results demonstrate that vehicle density is inversely related to resource utilization and positively related to the task completion ratio. This is because the number of SeVs increases with increasing vehicle density, and resource utilization decreases due to the constant number of TaVs. Moreover, with the increase in the number of SeVs, there will be more available resources around TaVs and more computation tasks can be offloaded to SeVs, which will lead to the improvement of the task completion ratio. On the contrary, when the vehicle density in the system is constant, the resource utilization increases and the task completion ratio decreases with the increase of the number of TaVs. As the number of TaVs increases, more TaVs compete for limited resources, which will lead to the increase in resource utilization. The reason is that when the number of TaVs increases to a certain extent, the resources of SeVs are fully utilized but still cannot meet resources required for the excessive growth of the number of TaVs. In the end, the size of total offloading is no longer changed and the task completion ratio is reduced.



(a)



(b)

FIGURE 2: (a) Resource allocation versus the number of TaVs. (b) Task completion ratio versus the number of TaVs.

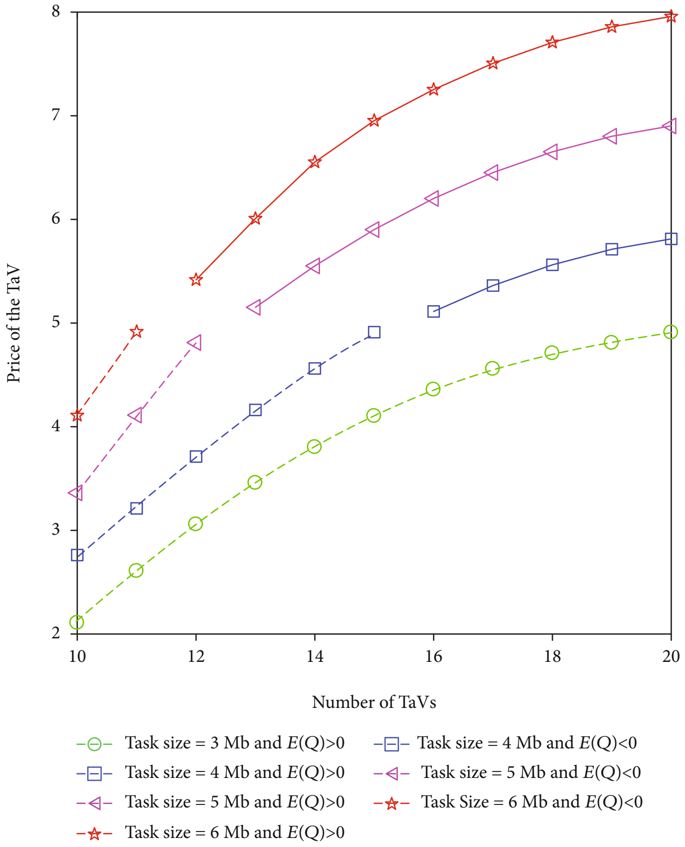
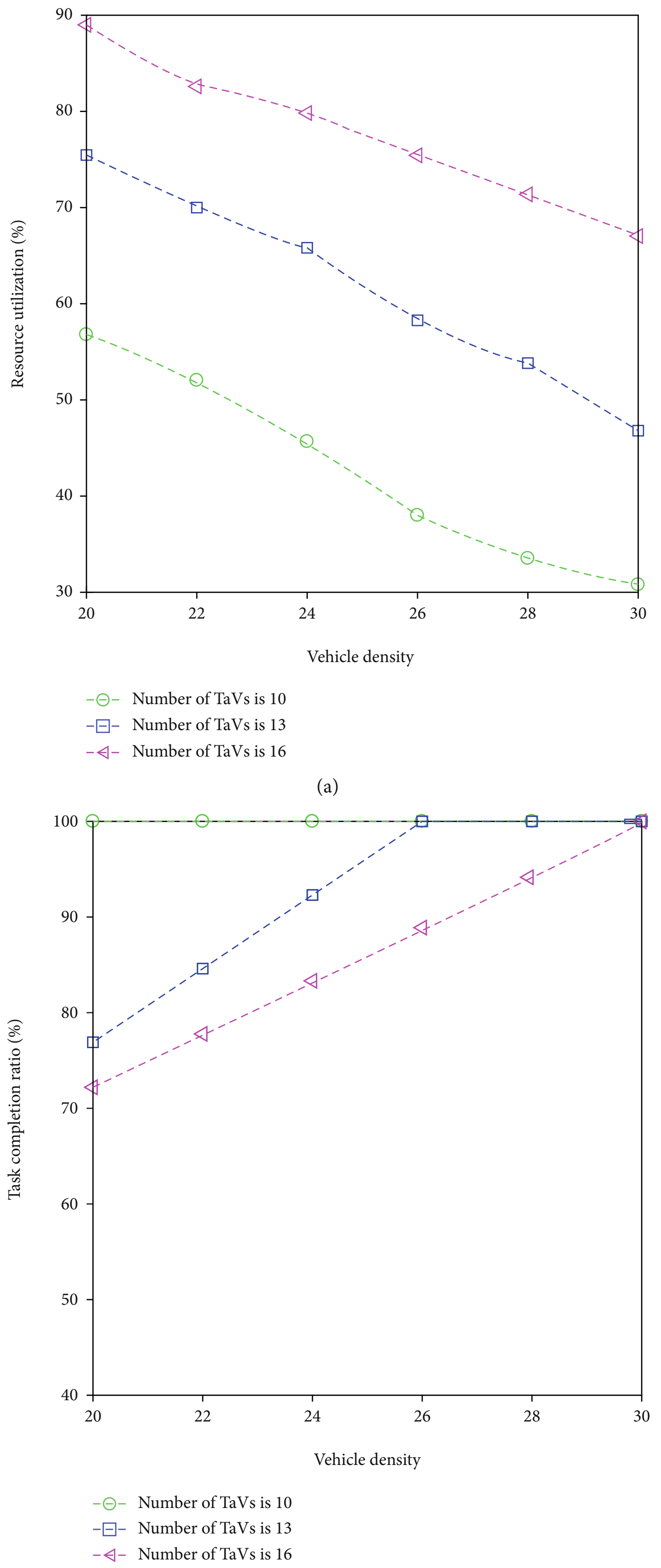


Figure 3: Price versus the number of TaVs.

5.3. Effect of TaV Speed. Here, we make comparison of TaV speed performance under different schemes. Figure 5 shows offloading utility and task completion ratio for different algorithms when TaV speed varies from 80 km/h to 120 km/h. It can be concluded that the task completion ratio and offload utility of POS, UOS, AOS, and LPS all decrease with increasing speed from Figures 5(a) and 5(b). As the speed of TaVs increases, the contact time between TaVs and SeVs becomes shorter, and the size of offloading task data decreases. Therefore, the offloading utility and task completion ratio decrease. Then, by comparing POS with other schemes, it can be seen that the task completion ratio and offload utility of POS are optimal. This is because POS optimizes the mobility of vehicles, the deadlines of TaVs, local computing resources of TaVs, and the service differentiation of different types of tasks. The resource allocation is superior to other schemes, and task allocation is also more reasonable than other schemes. Hence, the task completion ratio and offloading utility of POS are superior to other solutions. In addition, with the decrease of the task size processed by SeVs, the task offloading time is shortened and the success rate of task offloading is increased accordingly. What is more, since the choice of SeVs is not considered, LPS is not sensitive to the increase of speed on the task completion ratio of LPS.

5.4. Effect of Task Size. Figure 6(a) shows offloading utility versus the task size of TaVs. Numerical results demonstrate that task size is positively related to offloading utility. The numerical results show that when the task size is less than 4 Mb, the offloading utility of AOS is optimal and the offloading utility of POS is close to UOS. Since POS and UOS process a part of tasks locally, the size of offloading tasks is reduced, resulting in lower offloading utility than AOS. Moreover, it can be seen from Figure 6(a) that when the sizes of tasks are greater than 4 Mb, the offloading utility of POS is the highest. It can also be obtained that with the increase of task size of TaVs, the offloading utility of POS, HRS, and LCS first increases rapidly and then slowly increases. This is because the number of SeVs remains unchanged. When the task size of TaVs begins to increase, the offloading utility increases rapidly. However, with the increase of the size of the task to a certain extent, the resources of SeVs are fully utilized, which still cannot meet the resource requirements of the excessive increase in TaVs' task size, so the growth trend of offloading utility gradually slows down.

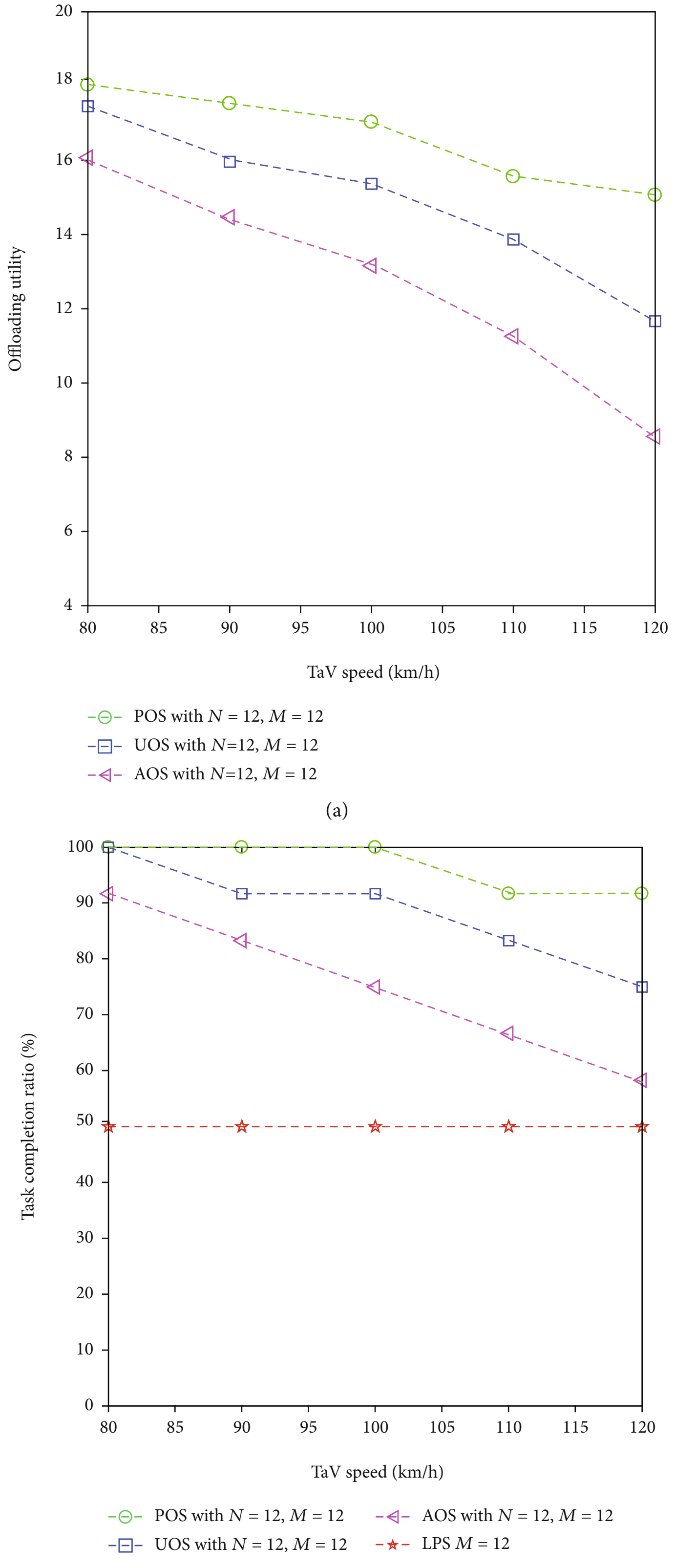
We also compare the task completion ratio in terms of the task size in Figure 6(b). It is interesting to see that the task completion ratio decreases with the increase of the task size. The reason is that as the sizes of the tasks processed by each vehicle increases, the risk of task exceeding the deadline also increases due to the constant deadline. As a result, the task completion rate is reduced. In addition, it can be seen that the UOS and AOS methods do not take the task processing time beyond the deadline into account, which results in the



(a)

(b)

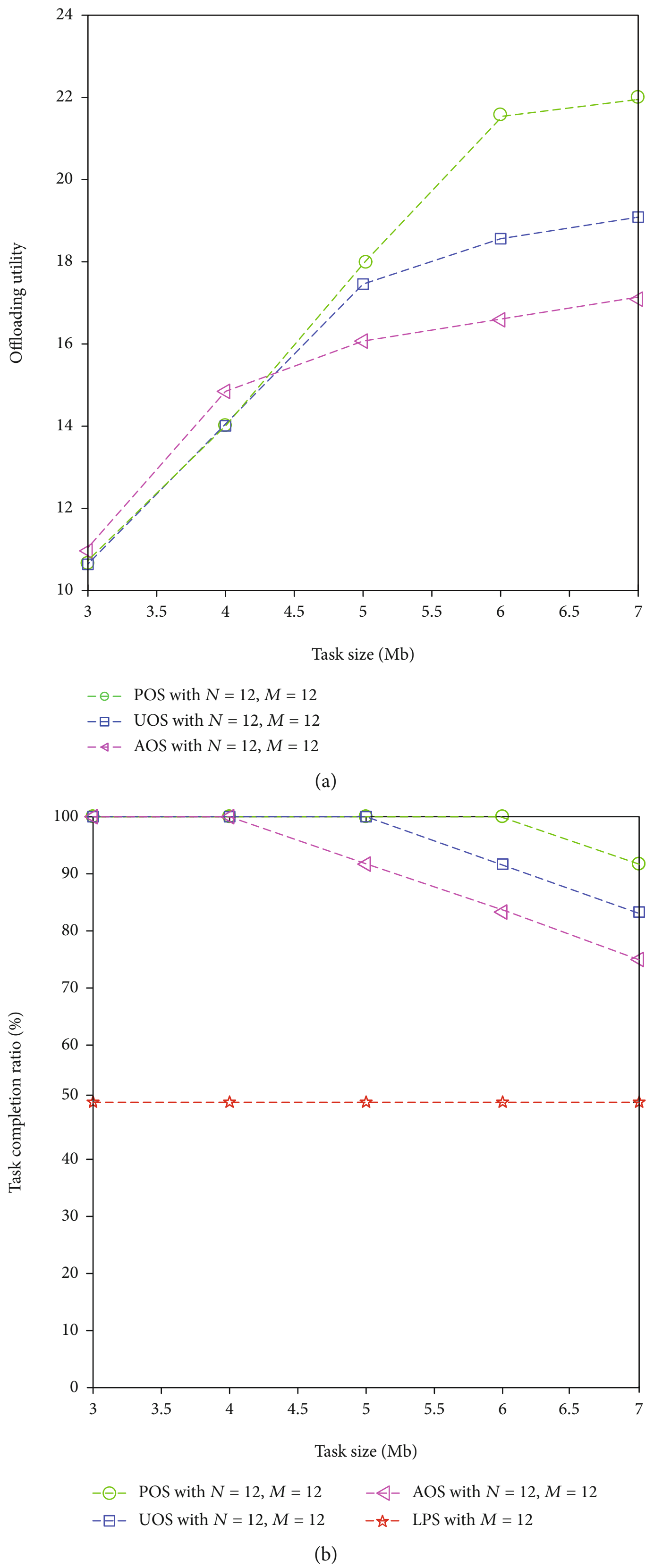
Figure 4: (a) Resource allocation versus vehicle density. (b) Task completion ratio versus vehicle density.



(a)

(b)

Figure 5: (a) Offloading utility versus TaV speed. (b) Task completion ratio versus TaV speed.



(a)

(b)

Figure 6: (a) Offloading utility versus task size. (b) Task completion ratio versus task size.

task offloading success rate lower than the POS, and the task completion rate is lower than the POS. What is more, LPS is executed locally without computational offloading, so it takes more time than the offloading mode.

6. Conclusion

In this paper, we investigate the joint task offloading and resource allocation mechanism in the VC system and maximize the total offloading revenue of tasks under the constraints of task completion deadlines and rapid vehicle movement characteristics. Considering the different types of services required by tasks in VC, this paper proposes a resource allocation model based on differentiated services and designs the pricing function of a task. Its pricing relies on the size of the task, the deadline for the task, and the number of requests and responses received by the vehicle that generated the task in the previous time slot. Furthermore, considering the mobility of vehicles, the deadline of tasks, and the limited computing resources, the centralized Lagrange dual method is used to solve the task offloading problem of the VC system. The simulation results show that the optimization effect of DiffServ offloading strategy is significantly improved, and the resource utilization and task completion ratio are significantly increased.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest

None of the authors have any conflicts of interest

Acknowledgments

This work was supported in part by the Chongqing University of Posts and Telecommunications Doctoral Fund project, E010A2019017, and the other part by the Science and Technology Research Program of Chongqing Municipal Education Commission (Grant No.KJQN201900645).

References

[1] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, *The Next Paradigm Shift: From Vehicular Networks to Vehicular Clouds*, Mobile Ad Hoc Networking, 2013.

[2] R. E. Sibai, T. Atechian, J. B. Abdo, R. Tawil, and J. Demerjian, "Connectivity-aware service provision in vehicular cloud," in *2015 International Conference on Cloud Technologies and Applications (CloudTech)*, Marrakech, Morocco, June 2015.

[3] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," *Journal of Network Computer Applications*, vol. 40, pp. 325–344, 2014.

[4] F. Yang, S. Wang, J. Li, Z. Liu, and Q. Sun, "An overview of internet of vehicles," *China Communications*, vol. 11, no. 10, pp. 1–15, 2014.

[5] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: a survey," *Mobile Networks and Applications*, vol. 25, pp. 1–42, 2020.

[6] J. Sun, Q. Gu, T. Zheng, P. Dong, and Y. Qin, "Joint communication and computing resource allocation in vehicular edge computing," *International Journal of Distributed Sensor Networks*, vol. 15, no. 3, Article ID 155014771983785, 2019.

[7] H. R. Arkian, R. E. Atani, and A. Pourkhalili, "A cluster-based vehicular cloud architecture with learning-based resource management," in *2014 IEEE 6th International Conference on Cloud Computing Technology and Science*, pp. 162–167, Singapore, Singapore, December 2014.

[8] L. Hou, K. Zheng, P. Chatzimisios, and Y. Feng, "A continuous-time markov decision process-based resource allocation scheme in vehicular cloud for mobile video services," *Computer Communications*, vol. 118, pp. 140–147, 2018.

[9] R. I. Meneguette, A. Boukerche, A. H. M. Pimenta, and M. Meneguette, "A resource allocation scheme based on semi-Markov decision process for dynamic vehicular clouds," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, Paris, France, May 2017.

[10] S. Wu, W. Xia, W. Cui, Q. Chao, and L. Shen, "An efficient offloading algorithm based on support vector machine for mobile edge computing in vehicular networks," in *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6, Hangzhou, China, October 2018.

[11] Q. Liu, Z. Su, and Y. Hui, "Computation offloading scheme to improve qoe in vehicular networks with mobile edge computing," in *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–5, Hangzhou, China, October 2018.

[12] M. Liwang, S. Dai, Z. Gao, Y. Tang, and H. Dai, "A truthful reverse-auction mechanism for computation offloading in cloud-enabled vehicular network," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4214–4227, 2019.

[13] Y. Sun, J. Song, S. Zhou, X. Guo, and Z. Niu, "Task replication for vehicular edge computing: a combinatorial multi-armed bandit based approach," in *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, Abu Dhabi, United Arab Emirates, December 2018.

[14] Z. Jiang, S. Zhou, X. Guo, and Z. Niu, "Task replication for deadline-constrained vehicular cloud computing: optimal policy, performance analysis, and implications on road traffic," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 93–107, 2018.

[15] M. LiWang, J. Wang, Z. Gao, X. Du, and M. Guizani, "Game theory based opportunistic computation offloading in cloud-enabled iov," *IEEE Access*, vol. 7, pp. 32551–32561, 2019.

[16] M. Liu and Y. Liu, "Price-based distributed offloading for mobile-edge computing with computation capacity constraints," *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 420–423, 2018.

[17] T. Zhao, S. Zhou, X. Guo, Y. Zhao, and Z. Niu, "A cooperative scheduling scheme of local cloud and internet cloud for delay-aware mobile cloud computing," in *2015 IEEE Globecom Workshops (GC Wkshps)*, pp. 1–6, San Diego, CA, USA, December 2015.

[18] J. Liu and Q. Zhang, "Offloading schemes in mobile edge computing for ultra-reliable low latency communications," *IEEE Access*, vol. 6, pp. 12825–12837, 2018.

[19] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint load balancing and offloading in vehicular edge computing and

networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4377–4387, 2019.

[20] C.-K. Tham and B. Cao, "Stochastic programming methods for workload assignment in an ad hoc mobile cloud," *IEEE Transactions on Mobile Computing*, vol. 17, no. 7, pp. 1709–1722, 2018.

[21] A. N. Toosi, R. N. Calheiros, R. K. Thulasiram, and R. Buyya, "Resource provisioning policies to increase iaas provider's profit in a federated cloud environment," in *2011 IEEE International Conference on High Performance Computing and Communications*, pp. 279–287, Banff, AB, Canada, September 2011.

[22] W. Yu and R. Lui, "Dual methods for nonconvex spectrum optimization of multicarrier systems," *IEEE Transactions on Communications*, vol. 54, no. 7, pp. 1310–1322, 2006.