IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 15, 2019, accepted December 19, 2019, date of publication December 23, 2019,
date of current version January 3, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2961802

A Task Offloading Scheme in Vehicular Fog and Cloud Computing System

QIONG WU[1,2], (Member, IEEE), HONGMEI GE[1], HANXU LIU[1], QIANG FAN[3], ZHENGQUAN LI[1], (Member, IEEE), AND ZIYANG WANG[2]

[1]Key Laboratory of Advanced Process Control for Light Industry, School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China
[2]Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China
[3]Advanced Networking Laboratory, Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA

Corresponding author: Qiong Wu (qiongwu@jiangnan.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61701197 and Grant 61571108, in part by the Project Funded by the China Postdoctoral Science Foundation under Grant 2018M641354 and Grant 2019M650677, in part by the 111 Project under Grant B12018, and in part by the Wuxi Science and Technology Development Fund under Grant H20191001.

ABSTRACT Vehicular fog and cloud computing (VFCC) system, which provides huge computing power for processing numerous computation-intensive and delay sensitive tasks, is envisioned as an enabler for intelligent connected vehicles (ICVs). Although previous works have studied the optimal offloading scheme in the VFCC system, no existing work has considered the departure of vehicles that are processing tasks, i.e., the occupied vehicles. However, vehicles leaving the system with uncompleted tasks will affect the overall performance of the system. To solve the problem, in this paper, we study the optimal offloading scheme that considers the departure of occupied vehicles. We first formulate the task offloading problem as an semi-Markov decision process (SMDP). Then we design the value iteration algorithm for the SMDP to maximize the total long-term reward of the VFCC system. Finally, the numerical results demenstrate that the proposed offloading scheme can achieve higher system reward than the greedy scheme.

INDEX TERMS Vehicular fog computing, cloud computing, task offloading, semi-Markov decision process.

I. INTRODUCTION

Intelligent connected vehicles (ICVs) are considered as a promising technology to solve a series of traffic-related issues such as traffic congestion, traffic accident and environmental pollution [1]. ICVs can facilitate many smart applications such as autonomous driving [2], augmented reality [3] and natural language processing [4], which can assist both the drivers and passengers in the vehicular environment [5]. The implementation of those applications needs powerful computing capability to process numerous computation-intensive and delay sensitive tasks generated by the vehicle sensors. It is predicted that there are around 4000 GB data generated by vehicles every day in 2020 [6]. However, the computing capabilities of vehicles are limited. If all tasks are processed by the vehicles, it is difficult to meet the requirements of those smart applications [7].

The concept of cloud computing has been proposed to process enormous computational tasks [8]–[10]. The cloud is a distant centralized server with powerful computing capability. Vehicles are connected to the remote cloud (RC) through the advanced communication technologies such as fifth-generation (5G) [11]–[13], software defined networking (SDN) [14], [15], and cognitive radio [16], [17]. Then vehicles can enjoy the powerful computing resource by transmitting tasks to the cloud server. However, with the increasing data transmitted to the RC, the collision probabilities of data transmission between vehicles and the cloud server may increase dramatically, incurring poor network performance [18]. Moreover, the long distance between the cloud server and vehicles may result in high latency, thus failing to process the delay sensitive tasks. In this case, vehicular fog (VF) computing has been introduced to push the computing resources close to the computational tasks via Vehicle-to-Vehicle (V2V) communication and Vehicle-to-Infrastructure (V2I) communication [19]–[22]. As a result, the vehicles can obtain the analytical results in a very short time. Thus VF is envisioned as a promising method to process low-latency tasks and support the Internet of Thing (IoT) applications [23], [24]. However, due to the limited communication bandwidth and computing capability of on-board computer, it is difficult for the VF system to process such amount of tasks and meet the requirements of smart applications.

The associate editor coordinating the review of this manuscript and approving it for publication was Junhui Zhao.

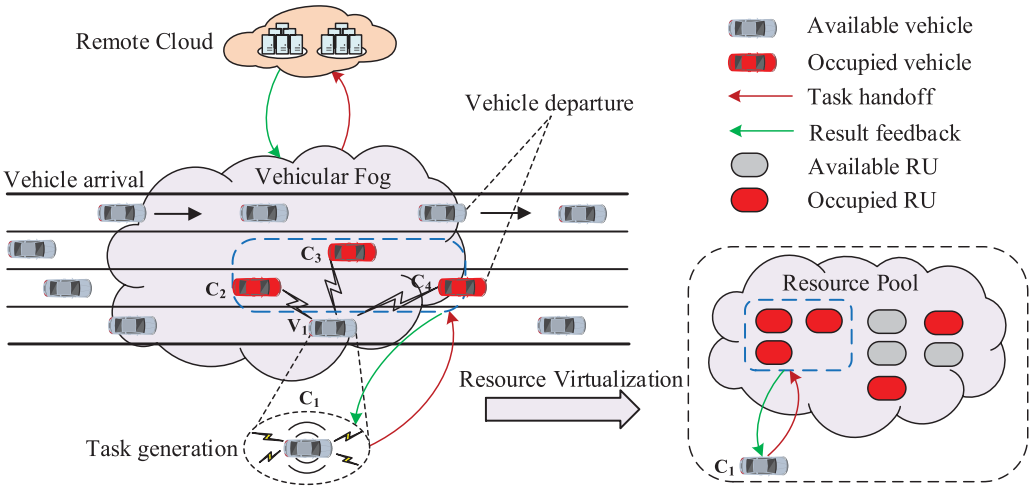


FIGURE 1. Illustration of VFCC system.

Vehicular fog and cloud computing (VFCC) system is a new computing architecture, that has been proposed to provide huge computing capability for processing numerous computation-intensive and delay sensitive tasks [25]. In the VFCC system, each vehicle is considered as not only the source of tasks but also a processor with a computing resource unit (RU). In other words, the vehicles can offload its tasks between each other. Owing to the dynamics of the VFCC system, we need to consider different factors when designing an offloading scheme: 1) tasks arrive at the system; 2) tasks are completed and then depart from the system; 3) vehicles join the system; 4) vehicles which are not processing tasks, i.e., the available vehicles, leave the system; and 5) vehicles which are processing tasks, i.e., the occupied vehicles, depart from the system. A simple example is given to explain the departure of occupied RUs (i.e., occupied vehicles) in Fig. 1. Vehicle C_1 generates a task and the system allocates three RUs, i.e., C_2, C_3 and C_4, to process the task. However, vehicle C_4 departs from the system before the task is completed. Due to the dynamic properties of VFCC system, the number of computing resources changes frequently. Thus when the above five features are considered, how to offload the computational tasks to maximize the benefit of VFCC system becomes a challenging problem. Note that the main benefits of the VFCC come from reducing the energy consumption and processing time of tasks [26], [27].

To the best of our knowledge, most of existing works mainly considered four factors in conducting task offloading [28], [29]. No existing work has taken into account the departure of occupied vehicles. However, an accurate model should consider all major factors. Furthermore, the occupied vehicles departing from the VFCC system will interrupt computational tasks, and thus degrade the overall efficiency of the system. Thus it is necessary to propose an optimal task offloading scheme that takes into consideration the departure of occupied vehicles to maximize the long-term reward of the VFCC system, which motivates us to do this work.

In this paper, we propose an optimal task offloading scheme. The optimal policy considers all major factors that affect the total long-term expected reward of the VFCC system. Our main contributions are summarized as follows.

1) We consider the departure of occupied vehicles in performing task offloading. Specifically, the task offloading problem is formulated as an semi-Markov decision process (SMDP) model. Then we define and analyze the states, actions, discounted rewards and transition probabilities of the SMDP model. Finally, an iterative algorithm is used to solve the task offloading problem.
2) To evaluate the performance of the proposed task allocating scheme, we conduct extensive experiments and attain numerical results. Numerical results illustrate that the proposed policy is able to improve the total long-term reward of VFCC system.

The paper is outlined as follows. Section II discusses the related work on task offloading problem in fog and cloud computing system. Section III describes the system model. The task offloading problem is formulated as an SMDP in Section IV. Section V introduces the iteration algorithm to find the optimal task offloading policy. Section VI evaluates the performance of proposed scheme through numerical results. Section VII concludes this paper.

II. RELATED WORK

A few works have been conducted to solve the task offloading problem in fog and cloud computing system.

Zheng *et al.* [29], investigated an optimal offloading scheme to maximize the long-term reward of the VFCC system. Specifically, they formulated the optimal resource allocation problem as an SMDP model and solved it by the value iteration algorithm. Liu *et al.* [30] took into account the different size of tasks and computation capacities of resources and proposed an integrated fog and cloud computing approach with non-orthogonal multiple access

to optimize the offloading scheme. They further solved the optimization problem by a low-complexity method which was called alternating direction method of multipliers based algorithms. Du *et al.* [31] considered the user fairness and maximal tolerable delay and investigated the joint optimization problem including the offloading scheme and transmit power assignment in a mixed fog/cloud computing system. They proposed a low-complexity suboptimal algorithm which was called computation offloading and resource allocation algorithm to solve the optimization problem. Deng *et al.* [32] took into account the constrained service delay and investigated the workload allocation problem in the fog-cloud computing system. They further divided the problem into three subproblems. Each subproblem was solved via exiting optimization techniques, i.e., convex optimization techniques, generalized Benders decomposition algorithm and Hungarian algorithm. Meng *et al.* [33] considered the different computation and communication capabilities of cloud server and fog computing and investigated the task offloading problem. Specifically, they defined a concept of computation energy efficiency and then solved the offloading problem by decomposing it into four subproblems. They further gave a closed-form computation offloading solution for each subproblem. Lin *et al.* [34], considered the heterogeneity of vehicles and roadside units and proposed an SMDP model for the VFCC system. They further developed the value iteration algorithm to find the optimal allocation scheme. Shah-Mansouri and Wong [35] proposed a computation offloading game to investigate the allocation of computing resources among IoT users in a hierarchical fog-cloud computing paradigm. They further proposed a near-optimal resource allocation algorithm to reduce the time complexity of reaching the pure Nash equilibrium. Du *et al.* [36] investigated the joint optimization problem including offloading scheme, resource allocation and power distribution in a mixed fog and cloud computing system. The offloading scheme was obtained by Binary Tailored Fireworks Algorithm based joint computation offloading and resource allocation algorithm (FAJORA) which was a general iterative algorithm with low-complexity. Li *et al.* [37], proposed two SMDP-based virtual machine allocation methods for the cloud-fog computing system. Specifically, they proposed an SMDP-based planning method to find the optimal policy. In this model, the generic SMDP was degraded into a continuous-time Markov decision process. The optimal virtual machine allocation policy was obtained via the relative value iteration algorithm. Then they also proposed the model-free reinforcement learning method to find an approximately optimal virtual machine allocation policy. Wang *et al.* [38] took into account the different delay requirements of the applications and formulated the resource allocation problem as an SMDP model. They further solved the offloading problem via iteration algorithm.

As described above, a few works have studied the task offloading problem in the VFCC system. However, no existing work takes into account the departure of occupied vehicles, which motives us to do this work.

III. SYSTEM MODEL

In this section, the system model will be described in detail. The scenario considering the departure of occupied vehicles is illustrated in Fig. 1. Vehicles moving on the highway form a dynamic vehicular fog. When a computational task is generated by a vehicle (i.e., a request vehicle), the request vehicle can transfer the task to VF via V2V communication. By this way, the VFCC system can reduce the energy consumption and processing time of tasks. We assume that the computing capacity of each vehicle is the same. Vehicles arrive at the system and depart from the system according to Poisson distribution with parameter λ_v and μ_v, respectively. Note that the maximal number of vehicles that the system can support is M. Tasks generation follows the Poisson distribution with parameter λ_p. The processing rate that a computing RU handles tasks is μ_p. Thus the aggregated processing rate of i RUs is $i\mu_p$. If a vehicle in the VFCC system generates a task, the system may decide to transfer it to the cloud server or the VF. Specifically, if the system determines to process the task in the VF. According to the number of available RUs (i.e., available vehicles), the VFCC system needs to make the decision to assign how many computing units to process the task. Finally, the analytical results will be sent back to the request vehicle from VF. If the task is transmitted to the RC, the vehicle receives results from the remote cloud. We use an example in Fig. 1 to further illustrate the VFCC system. Vehicle C_1 generates a compute-intensive task. Since the system has sufficient available RUs, the task is accepted by the VF and three RUs are allocated to process the task, i.e., C_2, C_3 and C_4. However, vehicle C_4 departs from the VFCC system before the task is completed.

IV. PROBLEM FORMULATION

In this section, we will formulate the task offloading problem as an SMDP with the states, actions, discounted reward model and transition probabilities of the VFCC system. Notations used in the analysis are summarized in Table 1.

A. STATES

A state is composed of the number of RUs that can be used to handle tasks, the number of tasks handled by different numbers of RUs and a specific event e, where the event $e \in H = \{A, L_1, \ldots, L_i, \ldots L_N, V_{+1}, V_{-1}, V_1, \ldots, V_i, \ldots, V_N\}$. Here, A denotes that a vehicle in the VFCC system generates a compute-intensive task. L_i denotes that a task handled by i RUs leaves the system. Denote N as the maximal number of computing units that a task can be allocated. V_{+1} means that a vehicle joins the VFCC system. V_{-1} denotes that the vehicle which is not processing a task departs from the VFCC system. V_i denotes that the vehicle which is processing a task with other $(i-1)$ RUs departs from

TABLE 1. Notations used in the analysis.

Notation	Description
A	A vehicle generates a compute-intensive task.
D_1	Transmission delay between request vehicle to VF.
D_2	Transmission delay between VF to RC.
I	Immediate income of the VFCC system.
G	Cost of VFCC system between two decision epochs.
K	Number of vehicles at the state x.
L_i	A task handled by i RUs leaves the system.
M	Number of vehicles that the VFCC can accommodate.
N	Maximal number of RUs that a task can be allocated.
q	State transition probability.
R	Long-term reward of the VFCC system.
s_i	Number of tasks handled by i RUs.
T_l	Time consumed by processing the task at request vehicle.
V_{+1}	A vehicle joins the VFCC system.
V_{-1}	The vehicle which is not processing a task departs from the VFCC system.
V_i	The vehicle which is processing a task with other $(i-1)$ RUs departs from the VFCC system.
λ_v	The rate that vehicles arrive at the VFCC system.
μ_v	The rate that vehicles departs from the VFCC system.
λ_p	The rate that vehicles generate tasks.
μ_p	The processing rate that a computing RU handles tasks.
ω_e	Weight of energy income.
ω_t	Weight of time income.
ζ_e	Saved price of per energy.
ζ_t	Saved price of per time.
E_l	Energy consumed by processing the task at request vehicle.
γ	Discount rate.
$-\eta_i$	The punishment of event V_i.
α	Continuous-time discount factor.

the VFCC system. Therefore, the state set can be denoted as

$$X = \{x | x = (K, s_1, \ldots, s_N, e)\}, \tag{1}$$

where K denotes the number of RUs at the state x which cannot exceed M, s_i denotes the number of tasks handled by i RUs. Obviously, the total number of occupied vehicles should satisfy $\sum_{i=1}^{N} i \cdot s_i \leqslant K$.

B. ACTIONS

An action indicates the decision of VFCC system under a specific event. When an event e happens, based on the current state x, the VFCC system will take an action $a(x)$, where $a(x) \in \Omega = \{-1, 0, 1, 2, \ldots, N\}$. Specifically, $a(x) = -1$ means that the system takes no action; $a(x) = 0$ means that the task is transmitted to the RC; $a(x) = i$ means that the VFCC system decides to accept the task by the VF and allocates i RUs($1 \leq i \leq N$) to process it. The action space under different events e can be given as

$$A_x = \begin{cases} \{-1\}, e \in \{L_1, \ldots, L_N, V_{+1}, V_{-1}, V_1, \ldots, V_N\} \\ \{0, 1, 2, \ldots, i, \ldots, N\}, e = A. \end{cases} \tag{2}$$

C. REWARDS

A reward indicates the benefit of the VFCC system after an action is taken. The rewards consist of both the immediate income $I(x, a)$ and the system cost $G(x, a)$ before next state, i.e.,

$$R(x, a) = I(x, a) - G(x, a). \tag{3}$$

Next, the formulas for immediate income and the cost will be derived, respectively.

1) IMMEDIATE INCOME

Since the system aims to reduce the energy consumption and processing time of tasks, the immediate income should take into account both of them. Furthermore, the state is related with the action a and event e. Thus, the immediate income $I(x, a)$ can be analyzed as follows.

(a) $a(x) = i(i \geq 1), e = A$
This case represents that the VFCC system decides to assign i RUs to handle the task. Specifically, after a vehicle in the system generates a task, the VFCC system assigns i computing RUs to handle the task and finally the analytical results are sent back to the request vehicle. Denote E_l as the energy consumed by processing the task at request vehicle, D_1 as the transmission delay between request vehicle and RUs and P_l as the transmission power of request vehicle. Then the energy saved by the system can be derived as $(E_l - P_l \cdot D_1)$. Similarly, let T_l be the time of processing the task at the request vehicle, $D_p(i)$ be the time that i RUs handle a task, i.e., $\frac{1}{i\mu_p}$. Then the time saved by the system is $[T_l - D_p(i) - D_1]$. Let ω_e and ω_t be the weight of energy income and time income, which should satisfy $\omega_e + \omega_t = 1$. ζ_e and ζ_t are the saved price of per energy unit and per time unit which transform the energy and time into the income [34], and φ be the cost of one unit of transmitting time. Thus, the immediate income of the VFCC system is expressed as $[\omega_e\zeta_e(E_l - P_l \cdot D_1) + \omega_t\zeta_t(T_l - D_p(i) - D_1) - \varphi D_1]$.

(b) $a(x) = 0, e = A$
This case represents that the system decides to process the task in the RC. Specifically, the request vehicle transfers the task to the VF. However, the available RUs are insufficient, then the VF transmits the computing

$$I(x, a) = \begin{cases} \omega_e\zeta_e(E_l - P_l \cdot D_1) + \omega_t\zeta_t\left[T_l - D_p(i) - D_1\right] - \varphi D_1, & a = i, e = A(i > 0) \\ \omega_e\zeta_e(E_l - P_l \cdot D_1) + \omega_t\zeta_t(T_l - D_1 - D_2) - \varphi(D_1 + D_2), & a = 0, e = A \\ 0, & a = -1, e \in \{L_1, \ldots, L_N, V_{+1}, V_{-1}\} \\ -\eta_i, & a = -1, e \in \{V_1, \ldots, V_N\} \end{cases} \tag{4}$$

task to the RC and finally the results are sent back to the request vehicle. We assume that D_2 is the transmission delay between the VF and the RC. Thus, the immediate income of the system can be denoted as $[\omega_e\zeta_e(E_l - P_l \cdot D_1) + \omega_t\zeta_t(T_l - D_1 - D_2) - \varphi(D_1 + D_2)]$.

(c) $a(x) = -1, e \in \{L_1, \ldots L_i, \ldots, L_N, V_{+1}, V_{-1}\}$
In this case, e is one of the events including a task handled by $i(1 \leq i \leq N)$ RUs leaves the system, a vehicle joins the VFCC and a vehicle which is not processing a task departs from the system. Under these events, the system does not take any action, and there is no income for the VFCC system.

(d) $a(x) = -1, e \in \{V_1, \ldots V_i, \ldots, V_N\}$
In this case, e is the event that an occupied vehicle which is processing a task with other $(i-1)$ RUs $(1 \leq i \leq N)$ departs from the system and no action is taken. Under this situation, the task processed by the occupied vehicle will be interrupted, reducing the overall efficiency of the system. Thus, the VFCC system should be punished and the punishment of event V_i is $-\eta_i$.

Conclusively, the immediate income $I(x, a)$ under different events and actions can be expressed as Eq. (4) as shown at the bottom of previous page.

2) COST

$G(x, a)$ indicates the cost of the VFCC system during the period of two decisions. We assume that the interval between two decisions follows an exponentially distribution with parameter α which is a continuous-time discount factor. According to [39], the discounted cost of the system between decision epochs can be denoted as

$$\begin{aligned} G(x,a) &= C(x,a)E_x^a\left\{\int_0^\tau e^{-\alpha t}dt\right\} \\ &= C(x,a)E_x^a\left\{\frac{1-e^{-\alpha\tau}}{\alpha}\right\} \\ &= \frac{C(x,a)}{\alpha+\beta(x,a)}, \end{aligned} \tag{5}$$

where $\beta(x, a)$ is the expected event rate. The average event rate can be calculated by adding up the occurrence rates of all events which may happen under the specific state x and action a. $C(x, a)$ is the cost rate which is related to the number of occupied vehicles, i.e.,

$$C(x,a) = \sum_{i=1}^{N} i \cdot s_i. \tag{6}$$

D. TRANSITION PROBABILITIES

In the VFCC system, the next state is affected by the current state and action. Similar to [29], we define the transition probability as the ratio between the rate of next event (i.e., the next event occurrence rate) and the expected event rate. Let $q(j|x, a)$ denote the transition probability from state x to state j after action a, where $x = (K, s_1, \ldots, s_N, e)$ and e is the current event. Based on different current events, the transition probabilities can be divided into the following cases.

In the first situation, the current event e is that a vehicle generates a task and the VFCC system may decide to assign i computing RUs $(i = 1, \ldots, N)$ for handling the task or transmit it to the cloud server. According to the definition of transition probabilities, we first discuss the rate of next event. Then the formulas of transition probabilities will be derived.

If the next event is that a vehicle generates a compute-intensive task, i.e., A, the rate of next event is the rate that vehicles generate tasks. Since the rate that vehicles generate tasks is λ_p and the number of vehicles at the state x is K, the rate of next event is equal to $K\lambda_p$. If the next event is that a task handled by i computing units leaves the VFCC system, i.e., L_i, the rate of next event is explained as follows. When the decision of the VFCC system is transferring the task to the RC, the rate of next event can be denoted as $s_i \cdot i\mu_p$. When i RUs are assigned to handle the task, the rate of next event can be given as $(s_i+1)\cdot i\mu_p$. However, if the next event is L_m $(m \neq i)$, the rate of next event is $s_m \cdot m\mu_p$. If the next event is that a vehicle joins the VFCC system, i.e., V_{+1}, the rate of next event is the rate that vehicles arrive at the system, i.e., λ_v. The rate that vehicles departs from the system is μ_v. If the next event is that a vehicle departs from the VFCC, the rate of next event can be analyzed as following:

(a) $a = 0, e = V_i$
In this situation, the system decides to transmit the task to the cloud server and the next event is that a vehicle which is processing the task with other $(i-1)$ RUs departs from the system. Since the number of tasks handled by i computing RUs is s_i, the corresponding total number of occupied vehicles can be calculated as $i \cdot s_i$. Thus the rate of next event can be denoted as $\frac{i \cdot s_i}{K}\mu_v$.

(b) $a = 0, e = V_{-1}$
In this situation, the task is processed by the RC and the next event is that a vehicle which is not processing the task departs from the system. Since the total number of occupied vehicles is $\sum_{i=1}^{N} i \cdot s_i$, the number of vehicles which are not handling the tasks is $(K - \sum_{i=1}^{N} i \cdot s_i)$. Thus the rate of next event can be given by $\left(1 - \frac{\sum_{i=1}^{N} i \cdot s_i}{K}\right)\mu_v$.

(c) $a = i, e = V_i$
In this situation, the system decides to assign i computing units to handle the task and the next event is that an occupied vehicle departs from the system. The number of tasks handled by i computing RUs is $(s_i + 1)$ and the corresponding number of occupied vehicles is $i \cdot (s_i + 1)$. Thus the rate of next event is $\frac{i \cdot (s_i+1)}{K}\mu_v$.

(d) $a = i, e = V_m, m \neq i$
In this situation, the next event is that an occupied vehicle processing a task with other $(m-1)$ RUs departs from the system. Since the system determines to assign i RUs to handle the task, the number of compute-intensive tasks handled by m RUs has not changed and the corresponding number of occupied vehicles is $m \cdot s_m$. The rate of next event can be expressed as $\frac{m \cdot s_m}{K}\mu_v$.

(e) $a = i, e = V_{-1}$
In this situation, the next event is that an available vehicle departs from the system. Since i computing units are assigned to handle the task, the number of occupied vehicles is $(\sum_{i=1}^{N} i \cdot s_i + i)$ and the number of available vehicles is $(K - \sum_{i=1}^{N} i \cdot s_i - i)$. Thus the rate of next event can be expressed as $\left(1 - \frac{\sum_{i=1}^{N} i \cdot s_i + i}{K}\right) \mu_v$.

Since the expected event rate is $\beta(x, a)$, the transition probability is the ratio between the rate of next event and the expected event rate. Thus if the current event is that a vehicle in the system generates a task, i.e., A, the transition probabilities can be expressed as Eq. (7).

1) $x = (K, s_1, \ldots, s_N, A)$

$$q(j|x,a) = \begin{cases} \frac{K\lambda_p}{\beta(x,a)}, & a=0, j=(K, s_1, \ldots, s_N, A) \\ \frac{s_i \cdot i\mu_p}{\beta(x,a)}, & a=0, j=(K, s_1, \ldots, s_N, L_i) \\ \frac{\lambda_v}{\beta(x,a)}, & a=0, j=(K, s_1, \ldots, s_N, V_{+1}) \\ \left(1 - \frac{\sum_{i=1}^{N} i \cdot s_i}{K}\right) \cdot \frac{\mu_v}{\beta(x,a)}, & a=0, \\ & j=(K, s_1, \ldots, s_N, V_{-1}) \\ \frac{i \cdot s_i}{K} \cdot \frac{\mu_v}{\beta(x,a)}, & a=0, j=(K, s_1, \ldots, s_N, V_i) \\ \frac{K\lambda_p}{\beta(x,a)}, & a=i, j=(K, s_1, \ldots, s_i+1, \ldots, s_N, A) \\ \frac{(s_i+1) i\mu_p}{\beta(x,a)}, & a=i, j=(K, s_1, \ldots, s_i+1, \ldots, s_N, L_i) \\ \frac{s_m \cdot m\mu_p}{\beta(x,a)}, & a=i, m \neq i, \\ & j=(K, s_1, \ldots, s_i+1, \ldots, s_N, L_m) \\ \frac{\lambda_v}{\beta(x,a)}, & a=i, j=(K, s_1, \ldots, s_i+1, \ldots, s_N, V_{+1}) \\ \left(1 - \frac{\sum_{i=1}^{N} i \cdot s_i + i}{K}\right) \cdot \frac{\mu_v}{\beta(x,a)}, & a=i, \\ & j=(K, s_1, \ldots, s_i+1, \ldots, s_N, V_{-1}) \\ \frac{i \cdot (s_i+1)}{K} \cdot \frac{\mu_v}{\beta(x,a)}, & a=i, \\ & j=(K, s_1, \ldots, s_i+1, \ldots, s_N, V_i) \\ \frac{m \cdot s_m}{K} \cdot \frac{\mu_v}{\beta(x,a)}, & a=i, m \neq i, \\ & j=(K, s_1, \ldots, s_i+1, \ldots, s_N, V_m) \end{cases} \tag{7}$$

Similarly, if the current event e is the departure of a task, arrival of a vehicle, departure of an available vehicle, or departure of an occupied vehicle, i.e., L_i, V_{+1}, V_{-1} and V_i, the transition probabilities in these cases are expressed as Eqs. (8)-(11), respectively.

2) $x = (K, s_1, \ldots, s_N, L_i)$

$$q(j|x,a) = \begin{cases} \frac{K\lambda_p}{\beta(x,a)}, & a=-1, j=(K, s_1, \ldots, s_i-1, \ldots, s_N, A) \\ \frac{(s_i-1) i\mu_p}{\beta(x,a)}, & a=-1, j=(K, s_1, \ldots, s_i-1, \ldots, s_N, L_i) \\ \frac{s_m \cdot m\mu_p}{\beta(x,a)}, & a=-1, m \neq i, \\ & j=(K, s_1, \ldots, s_i-1, \ldots, s_N, L_m) \\ \frac{\lambda_v}{\beta(x,a)}, & a=-1, \\ & j=(K, s_1, \ldots, s_i-1, \ldots, s_N, V_{+1}) \\ \left(1 - \frac{\sum_{i=1}^{N} i \cdot s_i - i}{K}\right) \cdot \frac{\mu_v}{\beta(x,a)}, & a=-1, \\ & j=(K, s_1, \ldots, s_i-1, \ldots, s_N, V_{-1}) \\ \frac{i \cdot (s_i-1)}{K} \cdot \frac{\mu_v}{\beta(x,a)}, & a=-1, \\ & j=(K, s_1, \ldots, s_i-1, \ldots, s_N, V_i) \\ \frac{m \cdot s_m}{K} \cdot \frac{\mu_v}{\beta(x,a)}, & a=-1, m \neq i, \\ & j=(K, s_1, \ldots, s_i-1, \ldots, s_N, V_m) \end{cases} \tag{8}$$

3) $x = (K, s_1, \ldots, s_N, V_{+1})$

$$q(j|x,a) = \begin{cases} \frac{(K+1)\lambda_p}{\beta(x,a)}, & a=-1, j=(K+1, s_1, \ldots, s_N, A) \\ \frac{s_i \cdot i\mu_p}{\beta(x,a)}, & a=-1, j=(K+1, s_1, \ldots, s_N, L_i) \\ \frac{\lambda_v}{\beta(x,a)}, & a=-1, j=(K+1, s_1, \ldots, s_N, V_{+1}) \\ \left(1 - \frac{\sum_{i=1}^{N} i \cdot s_i}{K+1}\right) \cdot \frac{\mu_v}{\beta(x,a)}, & \\ & a=-1, j=(K+1, s_1, \ldots, s_N, V_{-1}) \\ \frac{is_i}{K+1} \cdot \frac{\mu_v}{\beta(x,a)}, & a=-1, j=(K+1, s_1, \ldots, s_N, V_i) \end{cases} \tag{9}$$

4) $x = (K, s_1, \ldots, s_N, V_{-1})$

$$q(j|x,a) = \begin{cases} \frac{(K-1)\lambda_p}{\beta(x,a)}, & a=-1, j=(K-1, s_1, \ldots, s_N, A) \\ \frac{s_i \cdot i\mu_p}{\beta(x,a)}, & a=-1, j=(K-1, s_1, \ldots, s_N, L_i) \\ \frac{\lambda_v}{\beta(x,a)}, & a=-1, j=(K-1, s_1, \ldots, s_N, V_{+1}) \\ \left(1 - \frac{\sum_{i=1}^{N} i \cdot s_i}{K-1}\right) \cdot \frac{\mu_v}{\beta(x,a)}, & \\ & a=-1, j=(K-1, s_1, \ldots, s_N, V_{-1}) \\ \frac{is_i}{K-1} \cdot \frac{\mu_v}{\beta(x,a)}, & a=-1, j=(K-1, s_1, \ldots, s_N, V_i) \end{cases} \tag{10}$$

5) $x = (K, s_1, \ldots, s_N, V_i), i = 1, \ldots, N$

$$q(j|x,a) = \begin{cases} \dfrac{(K-1)\lambda_p}{\beta(x,a)}, & a=-1, \\ & j=(K-1, s_1, \ldots, s_i-1, \ldots, s_N, A) \\ \dfrac{(s_i-1)\, i\mu_p}{\beta(x,a)}, & a=-1, \\ & j=(K-1, s_1, \ldots, s_i-1, \ldots, s_N, L_i) \\ \dfrac{s_m \cdot m\mu_p}{\beta(x,a)}, & a=-1, m\neq i, \\ & j=(K-1, s_1, \ldots, s_i-1, \ldots, s_N, L_m) \\ \dfrac{\lambda_v}{\beta(x,a)}, & a=-1, \\ & j=(K-1, s_1, \ldots, s_i-1, \ldots, s_N, V_{+1}) \\ \left(1-\dfrac{\sum_{i=1}^{N} i\cdot s_i - i}{K-1}\right)\cdot\dfrac{\mu_v}{\beta(x,a)}, & a=-1, \\ & j=(K-1, s_1, \ldots, s_i-1, \ldots, s_N, V_{-1}) \\ \dfrac{i(s_i-1)}{K-1}\cdot\dfrac{\mu_v}{\beta(x,a)}, & a=-1, \\ & j=(K-1, s_1, \ldots, s_i-1, \ldots, s_N, V_i) \\ \dfrac{ms_m}{K-1}\cdot\dfrac{\mu_v}{\beta(x,a)}, & a=-1, m\neq i, \\ & j=(K-1, s_1, \ldots, s_i-1, \ldots, s_N, V_m) \end{cases} \tag{11}$$

The average event rate $\beta(x, a)$ is the sum of rates of all events that may occur with the current state x and action a. Under different events and actions, $\beta(x, a)$ can be calculated by Eq. (12).

$$\beta(x,a) = \begin{cases} K\lambda_p + \lambda_v + \mu_v + \left(\sum_{i=1}^{N} i\cdot s_i + i\right)\mu_p, \\ \qquad e=A, a=i(0\le i\le N) \\ K\lambda_p + \lambda_v + \mu_v + \left(\sum_{i=1}^{N} i\cdot s_i - i\right)\mu_p, \\ \qquad e=L_i, a=-1 \\ (K+1)\lambda_p + \lambda_v + \mu_v + \sum_{i=1}^{N} i\cdot s_i\mu_p, \\ \qquad e=V_{+1}, a=-1 \\ (K-1)\lambda_p + \lambda_v + \mu_v + \sum_{i=1}^{N} i\cdot s_i\mu_p, \\ \qquad e=V_{-1}, a=-1 \\ (K-1)\lambda_p + \lambda_v + \mu_v + \sum_{i=1}^{N} i\,(s_i-1)\mu_p, \\ \qquad e\in\{V_1,\ldots,V_N\}, a=-1. \end{cases} \tag{12}$$

Fig. 2 shows a simple state transition diagram. The current state is expressed as $x = \{10, 1, 1, 1, A\}$, i.e., the total number of RUs is 10 and the current event is that a vehicle generates a task. Since there are four available RUs in the current state, the VFCC system may transfer the task to the RC, i.e., $a(x) = 0$, or allocate i RUs ($i = 1,2,3$) to process the task. The corresponding transition probabilities can be calculated as Eq. (7).

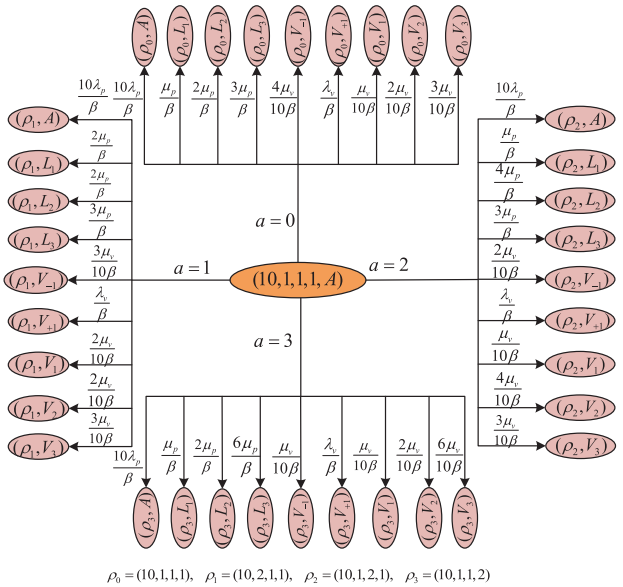


FIGURE 2. State transition diagram.

V. SOLUTION

In this section, we adopt a value iteration method to maximize the total long-term reward of the VFCC system. Firstly, the value iteration algorithm is briefly described. Then the continuous-time SMDP is transformed into a discrete-time SMDP by normalizing the reward, transition probabilities and discount rate. Finally, the detailed description of finding optimal policy π^* will be presented.

The value iteration algorithm is widely used to solve the optimal problem. For each state x, we utilize the Bellman optimal equation [39] to calculate the maximal state-value function $v_*(x)$ under different actions. The equation is shown in Eq. (13). If the maximal state-value function of each state converges, the above step terminates.

$$v_*(x) = \max_{a\in A_x}\left[R(x,a) + \gamma\sum_{j\in X} q(j|x,a)\, v_*(j)\right], \tag{13}$$

where γ is the discount rate, $0 \le \gamma \le 1$. The discount rate decides the present value of future rewards and can be calculated as $\gamma = \beta(x, a)/(\alpha + \beta(x, a))$.

According to the Bellman optimal equation [39], given a parameter $y = M\lambda_p + M \cdot N \cdot \mu_p + \lambda_v + \mu_v$, the reward, transition probabilities and discount rate can be normalized accordingly. The normalized equations are expressed as

$$\hat{r}(x,a) = R(x,a)\frac{\alpha+\beta(x,a)}{\alpha+y} \tag{14}$$

$$\hat{\gamma} = \frac{y}{(y+\alpha)} \tag{15}$$

Algorithm 1 Value Iteration, for Finding Policy π^*

1 Initialize $v_*(x) = 0$, for all state $x \in X$; set $k = 0$, and set convergence rate ε;
2 **for** *each system state* $x \in X$ **do**
3 $\quad \hat{v}_*(x) = \max_{a \in A_x} \left[\hat{r}(x,a) + \hat{\gamma} \sum_{j \in X} \hat{q}(j|x,a)\, \hat{v}_*(j) \right]$;
4 **if** $\left\| \Delta \hat{v}_*^{k+1}(x) \right\| = \max \left\| \hat{v}_*^{k+1}(x) - \hat{v}_*^{k}(x) \right\| < \frac{\varepsilon(1-\hat{\gamma})}{2\hat{\gamma}}$ **then**
5 $\quad$ **for** *each system state* $x \in X$ **do**
6 $\quad\quad \pi^*(x) = \arg\max_{a \in A_x} \left[\hat{r}(x,a) + \hat{\gamma} \sum_{x \in X} \hat{q}(j|x,a)\, \hat{v}_*^{k+1}(j) \right]$;
7 **else**
8 $\quad k++$;
9 $\quad$ go back to Line 2;
10 Return the optimal policy π^*;

$$\hat{q}(j|x,a) = \begin{cases} 1 - \dfrac{[1 - q(j|x,a)]\beta(x,a)}{y}, & j = x \\ \dfrac{q(j|x,a)\,\beta(x,a)}{y}, & j \neq x \end{cases} \tag{16}$$

Substituting Eqs. (14)-(16) into Eq. (13), the Bellman optimal equation can be written as

$$\hat{v}_*(x) = \max_{a \in A_x} \left[\hat{r}(x,a) + \hat{\gamma} \sum_{j \in X} \hat{q}(j|x,a)\, \hat{v}_*(j) \right]. \tag{17}$$

We will further describe the process of finding optimal policy π^*. Initially, the state-value function is set to be zero. Afterwards, for each state x, we calculate the maximal state-value function based on Eq. (17). Then the maximal difference is derived as

$$\left\| \Delta \hat{v}_*^{k+1}(x) \right\| = \max \left\| \hat{v}_*^{k+1}(x) - \hat{v}_*^{k}(x) \right\|, \quad x \in X, \tag{18}$$

where $v_*^{k+1}(x)$ is the maximal value function of state x in the $(k+1)th$ iteration. Then the threshold θ is calculated as

$$\theta = \frac{\varepsilon(1-\hat{\gamma})}{2\hat{\gamma}} \tag{19}$$

where ε is the convergence rate. Once the maximal difference $\Delta \hat{v}_*^{k+1}(x)$ is less than the threshold θ, the optimal policy π^* for each state x can be expressed as

$$\pi^*(x) = \arg\max_{a \in A_x} \left[\hat{r}(x,a) + \hat{\gamma} \sum_{x \in X} \hat{q}(j|x,a)\, \hat{v}_*^{k+1}(j) \right], \tag{20}$$

Otherwise, the algorithm goes into the next iteration. The pseudocode of the value iteration method is given in Algorithm 1.

VI. NUMERICAL RESULTS

In this section, we evaluate the performance of optimal policy π^* which is calculated by Eq. (20) through experimental numerical results. We conduct the experiments on MATLAB 2010a. The scenario in consideration of the departure of occupied vehicles is shown in Fig. 1 and described in Section III. The maximal number of RUs that a task can be allocated is 3. Behavior 1 denotes that one RU handles a task; behavior 2 denotes that two RUs handle a task; behavior 3 denotes that three RUs handle a task; behavior 0 denotes that the VFCC system transfers the task to the RC. Finally, the long-term reward is compared under different methods, i.e., the proposed algorithm and greedy algorithm. The greedy algorithm means that the system always allocates the maximal number of available computing units to process a task. The values of related parameters are shown in Table 2.

TABLE 2. Parameter setup.

Parameter	Value	Parameter	Value
N	3	M	7-13
λ_p	2-6	μ_p	7-11
λ_v	6-20	μ_v	8
ω_e	0.5	ω_t	0.5
ζ_e	2	ζ_t	2
E_l	20	T_l	20
D_1	2	D_2	3.5
P_l	4	η_1	18
η_2	36	η_3	42
α	0.1	φ	2

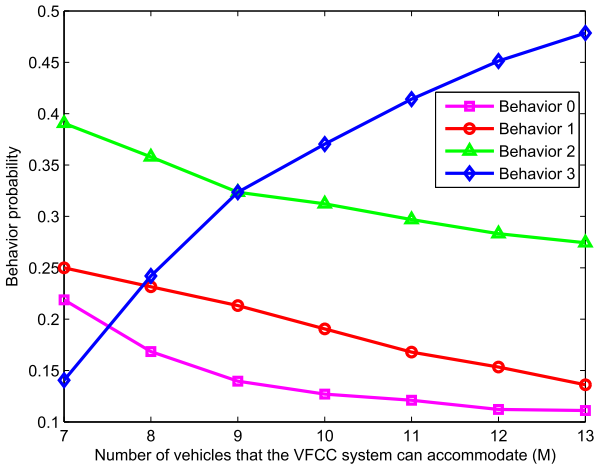


FIGURE 3. Behavior probabilities under different number of vehicles that the system can accommodate ($\lambda_p = 5$, $\mu_p = 8$, $\lambda_v = 16$).

Fig. 3 shows the behavior probabilities under different number of vehicles that the system can accommodate. Initially, the number of vehicles that the system can accommodate, i.e., M, is low. There are not sufficient computing resources to be allocated so that the probability of behavior 3 is less than those of behavior 0, behavior 1 and behavior 2. In Fig. 3, as the number of vehicles that the VFCC system can accommodate increases, the probabilities of behavior 0, behavior 1 and behavior 2 all decrease and the probability

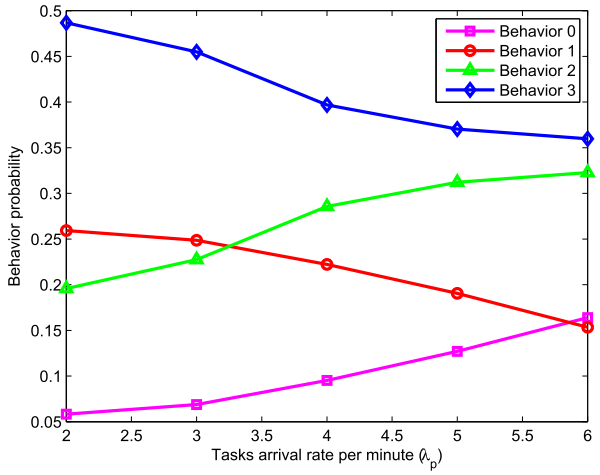


FIGURE 4. Behavior probabilities under different task arrival rates in the VFCC system ($K = 10, \mu_p = 8, \lambda_v = 16$).

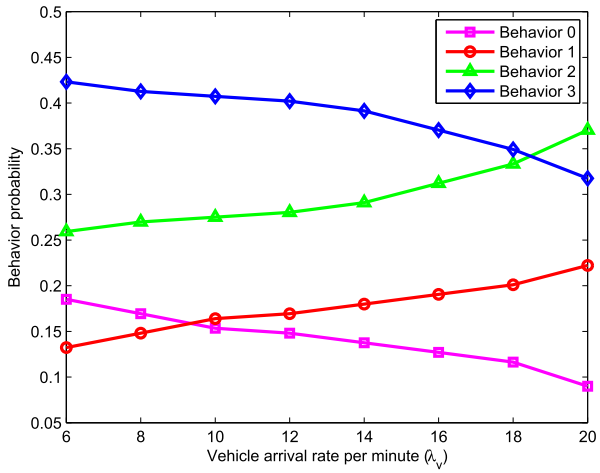


FIGURE 5. Behavior probabilities under different vehicle arrival rates in the VFCC system ($K = 10, \mu_p = 8, \lambda_p = 5$).

of behavior 3 becomes larger. This is because that there are more available computing resources for compute-intensive tasks. The VFCC system assigns as many computing units as possible to obtain the maximal long-term reward. Thus the probability of behavior 3 increases and those of behavior 0, behavior 1 and behavior 2 decrease.

Fig. 4 shows the behavior probabilities under different task arrival rates in the VFCC system. It is observed that the probability of behavior 0 is the lowest initially. This is because that when the arrival rates of tasks are low, there are sufficient RUs to be allocated. Thus the system is inclined to process the task in the VF, resulting in the situation that the probability of behavior 0 is the lowest initially. It also can be seen that as the arrival rate of tasks increases, the probability of behavior 3 gradually decreases and that of behavior 0 increases. This is because that the available computing resources is not especially abundant. The system decides to degrade the probability of behavior 3 and increases the probability of behavior 0 to reduce the energy consumption and processing time of tasks. Moreover, the VFCC system

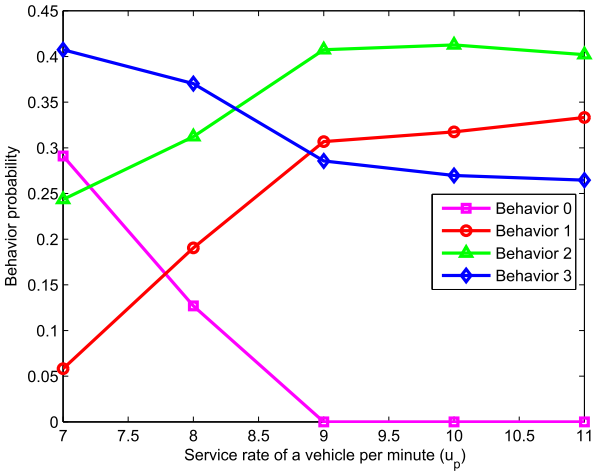


FIGURE 6. Behavior probabilities under different service rate of a vehicle in the VFCC system ($K = 10, \lambda_p = 5, \lambda_v = 16$).

trends to allocate as many RUs as possible, resulting in the increasing probability of behavior 2.

Fig. 5 shows the behavior probabilities under different vehicle arrival rates in the VFCC system. It is observed that as the arrival rates of vehicles increase, the probability of behavior 0 decreases. This is because that the available computing resources are gradually sufficient and the VF can process more tasks, causing that the probability of behavior 0 decreases. However, the probability of behavior 3 decreases and those of behavior 1 and behavior 2 increase with the vehicles arrival rate increasing. This is because that the punishment of V_3 is largest, thus to maximize the long-term reward, the probability of behavior 3 decreases. As a result, the system trends to allocate one or two RUs to process the tasks. Furthermore, the system aims at reducing the total energy consumption and processing time of tasks. Thus the system assigns as many computing units as possible, resulting in that the probability of behavior 2 is higher than that of behavior 1.

Fig. 6 shows the behavior probabilities under different service rate of a vehicle in the VFCC system. In Fig. 6, as the service rate of a vehicle increases, the probability of behavior 0 decreases. It can be explained as follows. As the processing rate that a RU handles tasks increases, the number of available computing RUs trends to become larger. To reduce the total energy consumption and processing time of tasks, the system is inclined to process the tasks in the VF, which degrades the probability of behavior 0. It also can be seen that as u_p increases, the probability of behavior 3 decreases. This is because that the punishment of V_3 is largest, which degrades the probability of behavior 3. To achieve the long-term benefit, the VFCC system will assign one or two computing resource units to process the tasks, resulting in the situation that the probabilities of behavior 1 and behavior 2 increase.

Next, we evaluate the proposed offloading algorithm of the SMDP model by comparing with GA algorithm. Fig. 7-Fig. 9 shows the tendency of long-term reward under different

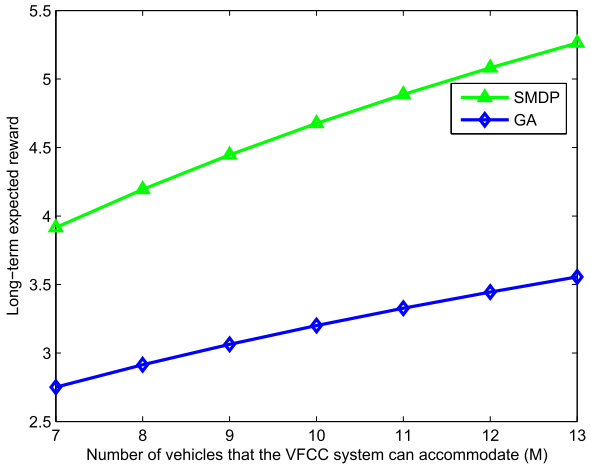


FIGURE 7. Behavior probabilities under different number of vehicles that the system can accommodate ($\lambda_p = 5$, $\mu_p = 8$, $\lambda_v = 16$).

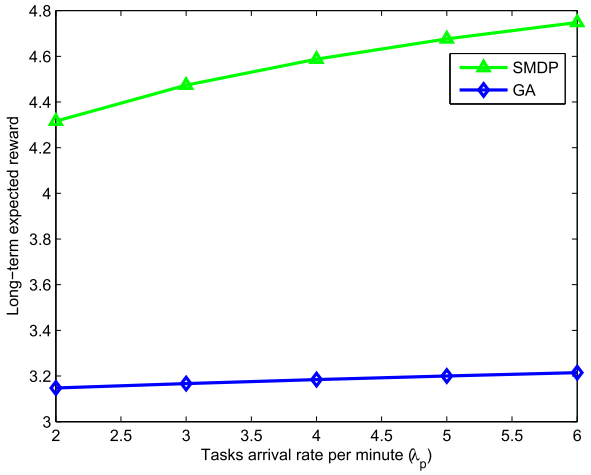


FIGURE 8. Behavior probabilities under different task arrival rates in the VFCC system ($K = 10$, $\mu_p = 8$, $\lambda_v = 16$).

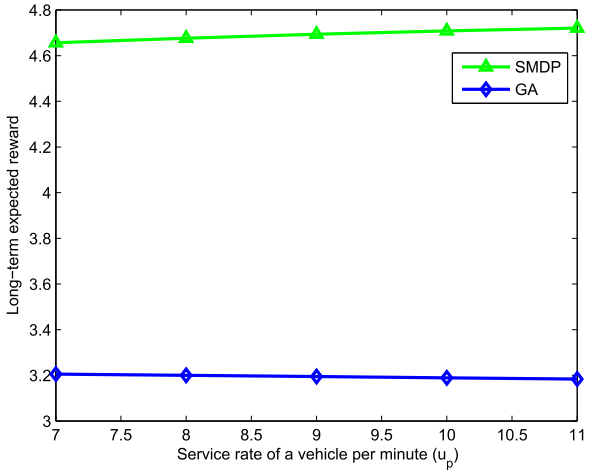


FIGURE 9. Behavior probabilities under different service rate of a vehicle in the VFCC system ($K = 10$, $\lambda_p = 5$, $\lambda_v = 16$).

situations, i.e., the number of vehicles that the VFCC system can accommodate, task arrival rate and service rate of a vehicle. It can be seen that their tendencies are similar, which can be explained as follows. In Fig. 7, as the number of vehicles that the VFCC system can accommodate increases, the long-term reward becomes higher. This is because that the proposed algorithm considers the punishments of the departure of occupied vehicles in performing task offloading, while the GA algorithm only trends to assign as many computing RUs as possible to handle tasks without considering other factors. In Fig. 8, the long-term reward becomes larger with the arrival rate of tasks increasing. This is because that more tasks are processed by the VF. In Fig. 9, as the service rate of a computing unit increases, the long-term reward is higher. This is because that the computing resources are more sufficient with the service rate increasing. In this case, the system considering the long-term reward takes actions. It is obvious that the proposed algorithm outperforms the GA algorithm.

VII. CONCLUSION

In this paper, we formulated the offloading problem in the VFCC system as an SMDP model with the defined state space, action space, discounted reward model and transition probabilities. To maximize the long-term of the system, the relative value algorithm was designed to solve the problem to find the optimal task offloading scheme. Numerical results showed that the performance of proposed scheme can achieve higher gains than the regularly used greedy scheme. In the future, the factor that vehicles periodically arrive at the system will be considered.

REFERENCES

[1] J. Han, A. Sciarretta, L. L. Ojeda, G. De Nunzio, and L. Thibault, "Safe- and Eco-driving control for connected and automated electric vehicles using analytical state-constrained optimal solution," *IEEE Trans. Intell. Veh.*, vol. 3, no. 2, pp. 163–172, Jun. 2018.

[2] Q. Wu, S. Xia, P. Fan, Q. Fan, and Z. Li, "Velocity-adaptive V2I fair-access scheme based on IEEE 802.11 DCF for platooning vehicles," *Sensors*, vol. 18, no. 12, p. 4198, Nov. 2018.

[3] S. Matsutomo, S. Manabe, V. Cingoski, and S. Noguchi, "A computer aided education system based on augmented reality by immersion to 3-D magnetic field," *IEEE Trans. Magn.*, vol. 53, no. 6, Jun. 2017, Art. no. 8102004.

[4] Q. Xie, X. Zhou, J. Wang, X. Gao, X. Chen, and C. Liu, "Matching real-world facilities to building information modeling data using natural language processing," *IEEE Access*, vol. 7, pp. 119465–119475, 2019.

[5] X. Ma, J. Zhao, Y. Gong, and X. Sun, "CSMA/CA-aware connectivity quality of downlink broadcast in vehicular relay networks," *IET Microw., Antennas Propag.*, vol. 13, no. 8, pp. 1–10, Mar. 2019.

[6] *Intel Autonomous Driving*. Accessed: Nov. 2019. [Online]. Available: https://www.driverlessguru.com/blog/intel-to-invest-250-million-into-autonomous-driving

[7] J. Zhao, Y. Liu, Y. Gong, C. Wang, and L. Fan, "A dual-link soft handover scheme for C/U plane split network in high-speed railway," *IEEE Access*, vol. 6, pp. 12473–12482, 2018.

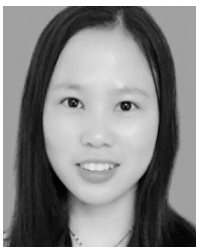
[8] S. Guo, J. Liu, Y. Yang, B. Xiao, and Z. Li, "Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 2, pp. 319–333, Feb. 2019.

[9] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7944–7956, Aug. 2019.

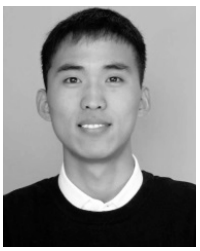
[10] J. Zheng, Y. Cai, Y. Wu, and X. Shen, "Dynamic computation offloading for mobile cloud computing: A stochastic game-theoretic approach," *IEEE Trans. Mobile Comput.*, vol. 18, no. 4, pp. 771–786, Apr. 2019.

[11] S. Ni, J. Zhao, H. H. Yang, and Y. Gong, "Enhancing downlink transmission in MIMO HetNet with wireless backhaul," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6817–6832, Jul. 2019.
[12] J. Zhao, S. Ni, L. Yang, Z. Zhang, Y. Gong, and X. Yu, "Multiband cooperation for 5G HetNets: A promising network paradigm," *IEEE Veh. Technol. Mag.*, vol. 14, no. 4, pp. 85–93, Dec. 2019, doi: 10.1109/MVT.2019.2935793.
[13] S. Ni, J. Zhao, and Y. Gong, "Optimal pilot design in massive MIMO systems based on channel estimation," *IET Commun.*, vol. 11, no. 7, pp. 975–984, May 2016.
[14] A. Lara and B. Ramamurthy, "OpenSec: Policy-based security using software-defined networking," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 1, pp. 30–42, Mar. 2016.
[15] J. Schulz-Zander, C. Mayer, B. Ciobotaru, R. Lisicki, S. Schmid, and A. Feldmann, "Unified programmability of virtualized network functions and software-defined wireless networks," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 4, pp. 1046–1060, Dec. 2017.
[16] Z. Junhui, Y. Tao, G. Yi, W. Jiao, and F. Lei, "Power control algorithm of cognitive radio based on non-cooperative game theory," *China Commun.*, vol. 10, no. 11, pp. 143–154, 2013.
[17] J. Zhao, X. Guan, and X. P. Li, "Power allocation based on genetic simulated annealing algorithm in cognitive radio networks," *Chin. J. Electron.*, vol. 22, no. 1, pp. 177–180, Jan. 2013.
[18] J. Zheng and Q. Wu, "Performance modeling and analysis of the IEEE 802.11p EDCA mechanism for VANET," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2673–2687, Apr. 2016.
[19] Q. Fan and N. Ansari, "Towards workload balancing in fog computing empowered IoT," *IEEE Trans. Netw. Sci. Eng.*, to be published, doi: 10.1109/TNSE.2018.2852762.
[20] R. Atallah, M. Khabbaz, and C. Assi, "Multihop V2I communications: A feasibility study, modeling, and performance analysis," *IEEE Trans. Veh. Technol.*, vol. 66, no. 3, pp. 2801–2810, Mar. 2017.
[21] Q. Wu, S. Nie, P. Fan, H. Liu, Q. Fan, and Z. Li, "A swarming approach to optimize the one-hop delay in smart driving inter-platoon communications," *Sensors*, vol. 18, no. 10, p. 3307, Oct. 2018.
[22] F. Zhang, J. Xi, and R. Langari, "Real-time energy management strategy based on velocity forecasts using V2V and V2I communications," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 2, pp. 416–430, Feb. 2017.
[23] Q. Wu, H. Liu, C. Zhang, Q. Fan, Z. Li, and K. Wang, "Trajectory protection schemes based on a gravity mobility model in IoT," *Electronics*, vol. 8, no. 2, p. 148, Feb. 2019.
[24] Q. Fan and N. Ansari, "Application aware workload allocation for edge computing-based IoT," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2146–2153, Jun. 2018.
[25] Q. Fan and N. Ansari, "On cost aware cloudlet placement for mobile edge computing," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 4, pp. 926–937, 2019.
[26] L. Lei, Z. Zhong, K. Zheng, J. Chen, and H. Meng, "Challenges on wireless heterogeneous networks for mobile cloud computing," *IEEE Wireless Commun.*, vol. 20, no. 3, pp. 34–44, Jun. 2013.
[27] Q. Fan, X. Sun, and N. Ansari, "Energy driven avatar migration in green cloudlet networks," *IEEE Commun. Lett.*, vol. 21, no. 7, pp. 1601–1604, Jul. 2017.
[28] Q. Wu, H. Liu, R. Wang, P. Fan, Q. Fan, and Z. Li, "Delay sensitive task offloading in the 802.11p based vehicular fog computing systems," *IEEE Internet Things J.*, to be published.
[29] K. Zheng, H. Meng, P. Chatzimisios, L. Lei, and X. Shen, "An SMDP-based resource allocation in vehicular cloud computing systems," *IEEE Trans. Ind. Electron.*, vol. 62, no. 12, pp. 7920–7928, Dec. 2015.
[30] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. M. Leung, "Distributed resource allocation and computation offloading in fog and cloud networks with non-orthogonal multiple access," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12137–12151, Dec. 2018.
[31] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee," *IEEE Trans. Commun.*, vol. 66, no. 4, pp. 1594–1608, Apr. 2018.
[32] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1171–1181, Dec. 2016.
[33] X. Meng, W. Wang, and Z. Zhang, "Delay-constrained hybrid computation offloading with cloud and fog computing," *IEEE Access*, vol. 5, pp. 21355–21367, 2017.
[34] C.-C. Lin, D.-J. Deng, and C.-C. Yao, "Resource allocation in vehicular cloud computing systems with heterogeneous vehicles and roadside units," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3692–3700, Oct. 2018.
[35] H. Shah-Mansouri and V. W. S. Wong, "Hierarchical fog-cloud computing for IoT systems: A computation offloading game," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 3246–3257, Aug. 2018.
[36] J. Du, L. Zhao, X. Chu, F. R. Yu, J. Feng, and C. L. I, "Enabling low-latency applications in LTE-A based mixed fog/cloud computing systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1757–1771, Feb. 2019.
[37] Q. Li, L. Zhao, J. Gao, H. Liang, L. Zhao, and X. Tang, "SMDP-based coordinated virtual machine allocations in cloud-fog computing systems," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1977–1988, Jun. 2018.
[38] Z. Wang, Z. Zhong, and M. Ni, "Application-aware offloading policy using smdp in vehicular fog computing systems," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, May 2018, pp. 1–6.
[39] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY, USA: Wiley, 2005.

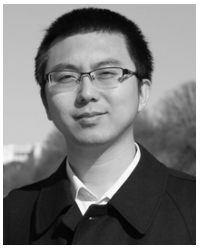
QIONG WU (S'13–M'18) received the Ph.D. degree in information and communication engineering from the National Mobile Communications Research Laboratory, Southeast University, Nanjing, China, in 2016. He is currently a Lecturer with the School of Internet of Things Engineering, Jiangnan University, Wuxi, China. He is also a Postdoctoral Researcher with the Department of Electronic Engineering, Tsinghua University. His current research interest includes autonomous driving communication technology.



HONGMEI GE received the B.S. degree from the East China University of Technology, Nanchang, China, in 2018. She is currently pursuing the M.S. degree with Jiangnan University. Her current research interests include vehicular fog and cloud computing.



HANXU LIU received the B.S. degree from Anhui Jianzhu University, in 2017. He is currently pursuing the M.S. degree with Jiangnan University. His current research interest includes vehicular cloud computing.



QIANG FAN received the B.S. degree from the Suzhou University of Science and Technology, China, in 2009, and the M.S. degree in electrical engineering from the Yunnan University of Nationalities, China, in 2013. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology (NJIT), Newark, New Jersey, USA. He is currently a Research Assistant with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology (NJIT). His research interests include mobile and cellular networks, mobile edge computing, the Internet of things, and free space optical communications.

ZHENGQUAN LI (M'17) received the B.S. degree from the Jilin University of Technology, in 1998, the M.S. degree from the University of Shanghai for Science and Technology, in 2000, and the Ph.D. degree in circuits and systems from Shanghai Jiao Tong University, in 2003. He is currently a Professor with Jiangnan University. He is also a Postdoctoral Researcher with the National Mobile Communications Research Laboratory, Southeast University. His current research interests include space time coding and cooperative communications and massive MIMO.

ZIYANG WANG was born in Henan, China, in 1991. He received the Ph.D. degree in electromagnetic wave and microwave technology from Xidian University, Xi'an, China, in 2018. He is currently a Postdoctoral Researcher with the National Laboratory for Information Science and Technology, Department of Electronic Engineering, Tsinghua University. His research interests include 360° all phase element and 1-bit active element design, phase controlled electromagnetic surface, multiple beam, vortex electromagnetic wave, other spatial beamforming, and global optimization applications such as particle swarm optimization/genetic algorithms, 5G MIMO antennas, array antenna decoupling, and antenna broadband design.