Offloading Time Optimization via Markov Decision Process in Mobile-Edge Computing

Guisong Yang, *Member, IEEE*, Ling Hou, Xingyu He, Daojing He, *Member, IEEE*, Sammy Chan, *Member, IEEE*, and Mohsen Guizani, *Fellow, IEEE*

***Abstract*—Computation offloading from a mobile device to the edge server is an emerging paradigm to reduce completion latency of intensive computations in mobile-edge computing (MEC). In order to satisfy the delay-sensitive computing tasks, offloading time, including task uploading time, task execution time, and results downloading time is adopted as the computational performance metrics for offloading nodes that perform offloaded computing tasks for mobile devices. Therefore, how to minimize the offloading time by selecting an optimal offloading node in MEC is of research importance. This work first investigates a MEC system consisting of mobile devices and heterogeneous edge severs that support various radio access technologies. Then, based on the available bandwidth of heterogeneous edge severs and the location of mobile devices, an optimal offloading node selection strategy is formulated as a Markov decision process (MDP), and solved by employing the value iteration algorithm (VIA). Finally, extensive numerical results demonstrate the effectiveness of the proposed strategy over classic strategies in terms of offloading time.**

***Index Terms*—Computation offloading, Markov decision process (MDP), mobile-edge computing (MEC), offloading time, value iteration algorithm (VIA).**

I. Introduction

IN RECENT years, with the development of Internet of Things (IoT) and mobile crowdsensing, the number of devices for information sensing, gathering, and computing has increased rapidly. According to IHS Markit forecasts, the IoT devices will grow to be more than 75 billion by 2025 [1]. Meanwhile, various novel and computing-intensive applications, such as vehicular gaming, augmented reality, and gesture recognition, are installed on the mobile devices and require diverse real-time services. Nevertheless, it becomes a bottleneck that liberating the tension between the extensive computations and ultralow latency.

To cope with the constrained computation capabilities of mobile devices, computation offloading is an approach to ease the ever-increasing computational burden of mobile devices. Offloading computing tasks to the remote cloud severs from mobile terminals is a solution in mobile cloud computing [2]. Although the remote cloud severs own powerful computing and storage capacity, considering the low bandwidth and service availability issues in mobile cloud computing [3], it will incur long transmission latency from mobile devices to remote cloud severs for data transmission.

As a new architecture, mobile-edge computing (MEC) is introduced to bridge the drawback of the remote cloud by the network edges [4], which can shorten the propagation delay and enhance the computational capability of mobile devices.

In MEC, edge servers (or offloading nodes) are closer to mobile devices than remote cloud severs for performing computing tasks in order to meet stringent quality of service requirements. Furthermore, computation offloading that performs computing tasks on edge servers is a promising technology to achieve computing agility and accomplish the demands of high computation capability in MEC. For a computing task, computation offloading involves the process of task uploading, task execution, and result downloading [5]. The offloading time is defined as the sum of the time corresponding to the three processes, which respectively, refers to task uploading time, task execution time, and task downloading time. Among offloading time, the task uploading time and result downloading time are related to the wireless access network of offloading nodes, and the task execution time is related to the computational capacity of offloading nodes [6].

In order to minimize the transmission time, including task uploading time and results downloading time, edge server-enabled heterogeneous wireless access networks are taken into consideration. Thus, as edge servers located at cellular base stations (BSs) provide nonorthogonal multiple access (NOMA) transmission, a NOMA-assisted computation offloading scheme is proposed to minimize the overall delay for finishing computation requirements [7]. However, it ignores the fact that the macro cells BSs may exist in the poor signal quality area in edge computing systems [8],

Manuscript received July 6, 2020; revised October 15, 2020; accepted October 20, 2020. Date of publication October 23, 2020; date of current version February 4, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61802257 and Grant 61602305, and in part by the Natural Science Foundation of Shanghai under Grant 18ZR1426000 and Grant 19ZR1477600. *(Corresponding author: Xingyu He.)*

Guisong Yang is with the Department of Computer Science and Engineering and the Shanghai Key Laboratory of Modern Optical system, University of Shanghai for Science and Technology, Shanghai 200093, China (e-mail: gsyang@usst.edu.cn).

Ling Hou is with the Department of Computer Science and Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China (e-mail: 172590535@st.usst.edu.cn).

Xingyu He is with the College of Communication and Art Design, University of Shanghai for Science and Technology, Shanghai 200093, China (e-mail: xy_he@usst.edu.cn).

Daojing He is with the School of Software Engineering, East China Normal University, Shanghai 200062, China (e-mail: djhe@sei.ecnu.edu.cn).

Sammy Chan is with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong (e-mail: eeschan@cityu.edu.hk).

Mohsen Guizani is with the Department of Computer Science and Computer Engineering, Qatar University, Doha, Qatar (e-mail: mguizani@ieee.org).

Digital Object Identifier 10.1109/JIOT.2020.3033285

so using small-cell BS as wireless access point (AP) is formulated to offload computation tasks for delay-sensitive applications to improve cell coverage and boost network capacity of data transmission [9]. Nevertheless, considering the heavy workloads in the cellular network, offloading computation tasks to edge servers-enabled WiFi network is also another option to ease the burden of large traffic volume [10]. Moreover, fog nodes located at network infrastructures [e.g., fifth-generation (5G) cell towers] or onboard opportunistic communication modules [e.g., WiFi and dedicated short-range communication (DSRC)] are proposed to provide latency-aware offloading strategies for computational tasks in vehicular fog computing [11]. However, the two dynamic factors are ignored in these works, which are the available network bandwidth and location of mobile devices, consistently, it will reduce the effectiveness of optimization algorithms for offloading time.

Although some prior research works have studied the computation offloading problem in MEC, most of them considered the problem from the perspective of static system. In contrast, the available network bandwidth and location of mobile devices change dynamically in practical computation offloading. In this situation, existing computation offloading strategies are not applicable. To fill this gap, in this work, the heterogeneous edge severs refer to the edge servers that are attached to either cellular networks or WiFi networks. On one hand, the available network bandwidth of the heterogeneous edge severs are not fully analyzed in previous studies for computation offloading in MEC. This means that since the data transmission rate under a specific network bandwidth in a real environment may change dynamically according to the background traffic, for offloaded computing tasks, the data transmission time depending on the available network bandwidth of the heterogeneous edge severs also changes dynamically. On the other hand, due to the limited radio radii of the offloading nodes, the mobility of mobile devices affects the strategy of the optimal offloading node selection (OONS). Therefore, this article focuses on how to select the optimal offloading node to minimize offloading time by considering both the available network bandwidth and the location of mobile devices.

In this article, we consider that edge severs are deployed at either femto cell BS or WiFi AP randomly. For instance, femto cells and WiFi hotspots are deployed to offload the real-time traffic monitored by smart camera in a highway scenario [12]. The deployment of femto cell BS in the 5G system can address the indoor coverage issue for indoor users [13]. Consequently, mobile devices can only offload computing tasks to offloading nodes through cellular or WiFi networks during the process of their movement. Then, for a mobile device, the OONS strategy is formulated as a Markov decision process (MDP) model, which incorporates the available network bandwidth of edge servers and the location of mobile devices. Finally, the minimum offloading time and OONS strategy can be obtained by employing the value iteration algorithm (VIA) to solve the MDP. What's more, this article differs from the previous works, which mainly studied computation offloading in a fixed environment. Noticing that the marco cell BS is imposed a heavy traffic load, this article considers dynamic changes in available network bandwidth of heterogeneous edge severs, which are composed of femto cell BS and WiFi AP to ensure delay-sensitive and massive connectivity. Meanwhile, mobile devices are making full use of the location prediction to minimize the offloading time.

The main contributions of this article are summarized as follows.

1) We consider both the heterogeneity of edge servers and the mobility of mobile devices in MDP formulation. Edge servers are deployed at femto cell BS to enhance the power of the poor signal quality area of macro cell BS or deployed at WiFi AP to alleviate the problem of network congestion. Besides, this article analyzes multiple possibilities for location changes of mobile devices when they offload computing tasks.
2) We propose an OONS strategy based on changes in available network bandwidth and location of mobile devices to minimize the offloading time. When calculating offloading time, this article emphasizes the importance on both the change probability of the available network bandwidth of the edge server and the change probability of location of the mobile device.
3) We conduct extensive simulations to evaluate the performances of the proposed strategy. The simulation results show that the proposed strategy can significantly minimize the offloading time compared with other four baseline strategies.

The remainder of this article is organized as follows. The related work is introduced in Section II and the system model is described in Section III. In Section IV, the offloading time optimization is explained, and the performance of the proposed strategy is evaluated in Section V. Finally, in Section VI, we summarize this article and discuss the future work.

II. Related Work

In recent years, computation offloading has attracted attention, which can augment computational capabilities and provide lower latency of mobile devices in MEC.

Researches are using the computation offloading technology to solve the limited computing resources problem of mobile devices. In vehicular edge computing, Jiang *et al.* [14] studied computation-intensive and delay-sensitive task scheduling, which offloaded multiple tasks divided by computation-intensive vehicular applications to roadside units. An efficient task scheduling algorithm is developed to solve the optimization problem of multiple tasks scheduling by prioritizing multiple applications and prioritizing multiple tasks. In addition, in fog computing systems [15], mobile devices offloaded its data or computational expensive tasks to the fog node instead of distant cloud, which utilized queuing theory to realize multiobjective optimization, including energy consumption, execution delay, and payment cost for computation offloading. More recently, the in IoT fog computing system [16], IoT device offloaded computation tasks according to the optimal solution for optimization problem, which is solved by the Lagrangian approach and the policy gradient method. These research attempts all enhanced the computing capabilities of mobile devices through computing offloading, however, how to select offloading nodes was not fully analyzed in edge computing.

Considering offloading nodes may be overloaded or broken when they are selected to perform computing tasks in computation offloading [8], it is important to design a computation offloading strategy providing pervasive and agile computation services. Alam *et al.* [17] proposed an autonomic computation offloading framework which scavenged the available edge resources by the distributed edge network controller and used a deep reinforcement Q-learning model to fit the computation resource demand for computation offloading. Considering the queue of edge resources in MEC, Mao *et al.* [18] proposed an optimal computation offloading strategy with multiple heterogeneous servers, so as to solve multivariable optimization problems by developing fast numerical algorithms which established queue models for multiple edge servers and power consumption models for the user equipment. Specifically, Li [19] proposed a Lyapunov optimization-based dynamic computation offloading policy, which jointly considered the transmission power for computation offloading and the CPU-cycle frequencies for mobile execution to reduce the execution latency and task failure. Due to the uncertainties of users and cloudlets, Le and Tham [20] first formulated the MDP model by considering the state of users and cloudlets, and then used deep reinforcement learning scheme to learn an effective method to offload computing tasks. Besides, considering the mobility of the user, an optimal offloading algorithm was developed to obtain an optimal policy in an intermittently connected cloudlet system [21]. The algorithm examined users' mobility patterns and availability of cloudlets, achieving lowest computation offloading cost for the mobile user. However, these research works failed to address the impact of changes in available network bandwidth on the computational offloading.

Some works concerned on the purpose of designing an optimal offloading strategy for computation offloading. To minimize total cost of smart mobile devices, Zhang *et al.* [22] proposed a system model in a 5G heterogeneous MEC network to gain a lower total cost. They presented a computation offloading scheme via an iterative search algorithm in scenarios of one macro cell and multismall cells. To cope with the limited battery power of smart mobile devices, Zhang *et al.* [23] proposed an energy-efficient computation offloading mechanism, which considered the energy cost of both task computing and file transmission. By this mechanism, each mobile device offloads their computation task to the MEC server with the minimal energy consumption. What is more, to satisfy the demand of latency sensitive applications, Li *et al.* [24] studied a double deep Q-network framework by virtue of deep reinforcement learning to reduce access latency by caching content in BSs. From the perspective of multiuser, Zhang *et al.* [25] considered advantages of NOMA and proposed a NOMA-enabled computation offloading scheme to minimize the overall delay.

From the perspective of a single user in MEC, Kuang *et al.* [26] proposed the partial offloading scheduling and power allocation problem to minimize the execution delay and energy consumption. To exploit the influence of mobility in computation offloading, Zhou *et al.* [27] utilized machine learning and coded computing technologies to predict mobility and reduce offloading delay throughout the computation offloading procedure. However, these aforementioned research works failed to take into account the influence of the heterogeneity of edge servers on the computational offloading.

To sum up, the related works have contributed to improve the computational performance of computation offloading, but their considerations are not comprehensive. For example, although Meng *et al.* [2] and Ko *et al.* [10] considered wireless bandwidth allocation and heterogeneous networks selection, respectively, both of them ignored the network disconnection which is caused by the movement of the mobile devices. Meanwhile, Jiang *et al.* [14] and Zhou *et al.* [27] focused on the exploration of intermittent connectivity and mobility, but did not consider the available bandwidth when transmitting data. In contrast to the existing literature, this article proposes an OONS scheme which focuses on both the mobility of mobile devices and the heterogeneity of edge servers in order to optimize offloading time of computation offloading in MEC.

III. System Model

In this section, we introduce the system model for computation offloading in MEC. As shown in Fig. 1, the system model consists of three tiers: "terminal-edge-cloud." The terminal tier consists of mobile devices (e.g., vehicles) that offload computing tasks and receive computing results from the offloading node. To simplify our analysis, we assume that the mobile devices move within a region divided into a number of grids, and each of the grid has a unique ID. The set of ID is denoted as $L = \{L_1, L_2, \ldots, L_l, \ldots, L_G\}$, $l = 1, 2, \ldots, G$, and G represents the total number of grids segmented in the region. A 2-D memoryless model [28] is adopted to describe the movement of the mobile device on the grids. During the moving process, the location information of mobile devices and the IDs of grids in which the mobile devices are located can be obtained.

The edge tier consists of edge servers serving as offloading nodes that have abundant computing resources to perform computing tasks from mobile devices. The set of edge servers is denoted as $N = \{N_1, N_2, \ldots, N_i, \ldots, N_M\}$, $i = 1, 2, \ldots, M$, and M represents the number of edge servers. Edge servers are close to mobile devices and have fixed location with a unique index as its identification. In the system, heterogeneous edge servers are deployed either at femto cell BS or WiFi AP, and different wireless networks have different radio radii. Due to the heterogeneity of edge servers, for a specific wireless access network, the available network bandwidth of it changes within its limit dynamically. Let $B = \{B_1, B_2, \ldots, B_i, \ldots, B_M\}$ represent the available network bandwidth of edge server i. In addition, the computing capability of an edge server is different from each other for having different CPU clock speeds.

The cloud tier consists of remote cloud that is generally far away from the mobile devices, thus causing a long communication delay. In this system, when the edge server cannot directly retrieve the results to the mobile device due to the

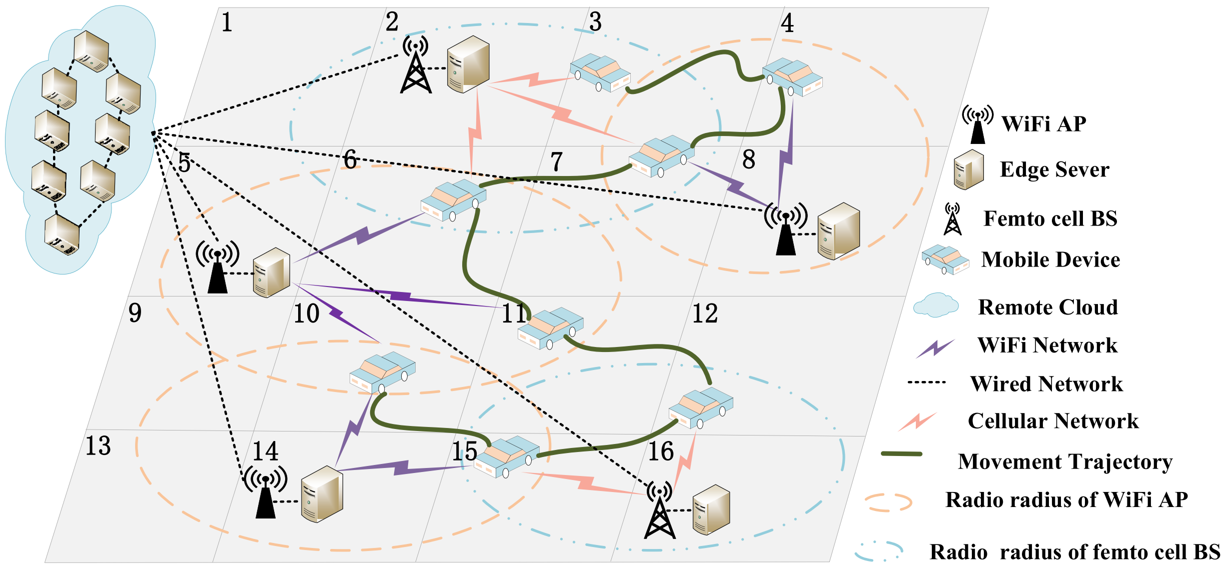

Fig. 1. System model of MEC for computation offloading.

offloading node failure, the remote cloud is used to accept the results of computing tasks from the edge server and retrieve the results to the mobile devices. Besides, this communication is done through a mix of wireless and wired connections, where mobile device connects wirelessly to an AP/BS which, in turn, have a wired connection to the remote cloud.

Computing tasks comprised of programs and data are generated by mobile devices in the system. The graph topologies of computing tasks execution can be expressed in a line, tree, or mesh. Based on the used and tested assumption that offloaded computing tasks are likely to be executed sequentially [29], thus, we use a linear topology to represent the relationship among computing tasks which are offloaded to the offloading nodes.

Since the available network bandwidth can change dynamically and the mobile device can move, the system is uncertain and varies constantly. In an uncertain dynamic system, an OONS strategy is a sequential decision, which offloads multiple computing tasks sequentially to offloading nodes for minimizing the offloading time. MDP is an effective mathematical model of the sequential decision to obtain an optimal strategy and optimize the system when outcomes are uncertain. Therefore, in the next section, how to optimize the offloading time by adopting an MDP model will be presented.

IV. Offloading Time Optimization

When a mobile device offloads computing tasks in MEC, the detail of offloading time optimization can be described in two parts: 1) MDP model formulation and 2) OONS. They are discussed in this section.

A. MDP Model Formulation

When the mobile device selects an offloading node to offload computing tasks, each selection must not be made in isolation, and needs to consider the relationship between current and future selection. MDP considers the immediate and corresponding reward and makes sequential decision under uncertainty, its model consists of five elements, namely, decision epochs, states, actions, transition probabilities, and rewards function. Considering the system model presented in Fig. 1, the detail of the five elements are described as follows.

Decision Epochs: A decision epoch denotes the period of a decision made by the mobile device. The times of successive decision epochs are represented by sequence T

$$T = \{1, 2, \dots, t, \dots, K\} \tag{1}$$

where K denotes the number of computing tasks, a specific decision epoch is described as t, $t \in T$.

States: A state includes the index and the available bandwidth of the edge servers, and the ID of grid that the mobile device may be located at. The states space S includes the total possible states for the system and defined as

$$S = N \times B \times L \tag{2}$$

where $\times$ represents cartesian product. $N = \{N_1, N_2, \dots, N_i, \dots, N_M\}$ and $B = \{B_1, B_2, \dots, B_i, \dots, B_M\}$ are M-dimension sets representing the index and the available bandwidth of the edge servers respectively, and $L = \{L_1, L_2, \dots, L_l, \dots, L_G\}$ is a G-dimension set representing ID of grid. Specifically, N_i represents the flag of edge server i, and $N_i \in \{0, 1\}$, $N_i = 1$ denotes that edge server i performs computing tasks uploaded by the mobile device, and $N_i = 0$ denotes that edge server i does not perform computing tasks uploaded by the mobile device. B_i represents the available network bandwidth of edge server i, the value of B_i depends on which wireless network the edge server is deployed (at femto cell BS or WiFi AP). L_l refers to the ID of grid l where the mobile device is currently located. At a specific decision epoch t, the current state can be defined as $s = (i, B_i, L_l)$, $s \in S$.

Action: An action is selected by mobile devices to decide which edge server to offload computing tasks. The action set

A denotes all actions that may be selected by mobile devices, and A is described by

$$A = \{a = (a_1, a_2, \ldots, a_j, \ldots, a_M)\} \tag{3}$$

where $a_j = \{0, 1, 2\}$, $j = 1, 2, \ldots, M$. $a_j = 0$ denotes that the edge server j is not selected as offloading node. $a_j = 1$ denotes that the edge server j enabled with cellular network is selected as offloading node, and $a_j = 2$ denotes that the edge server j enabled with the WiFi network is selected as the offloading node. Based on the current state s, the mobile device will select an action a at specific decision epoch t, $a \in A$.

Transition Probability: The transition probability $P(s'|s, a)$ represents the probability that current state $s = (i, B_i, L_l)$ could transfer to next state $s' = (i', B_i', L_l')$ by selecting action a, the transition probability is derived by

$$P(s'|s, a) = \begin{cases} P(L_l'|L_l)P(B_i'|B_i), & i' = i \\ 0, & i' \neq i \end{cases} \tag{4}$$

where $P(L_l'|L_l)$ represents the probability that the mobile device moves from grid L_l to grid L_l', $P(L_l'|L_l)$ is described by

$$P(L_l'|L_l) = \begin{cases} \mu, & \text{if } L_l' = L_l \\ \rho, & \text{otherwise} \end{cases} \tag{5}$$

where μ represents the probability of the mobile device staying at the same grid in two sequential decision epochs, $0 \leq \mu \leq 1$. Alternatively, ρ denotes the probability that the mobile device moves randomly to a grid L_l' adjacent to grid L_l, and ρ is expressed as follows:

$$\rho = (1 - \mu)/g \tag{6}$$

where g is the total number of adjacent grids of grid L_l.

$P(B_i'|B_i)$ represents the probability that the available network bandwidth of the offloading node varies from B_i to B_i', $P(B_i'|B_i)$ is given by

$$P(B_i'|B_i) = \begin{cases} \varphi, & \text{if } B_i' = B_i \\ \sigma, & \text{otherwise} \end{cases} \tag{7}$$

where φ is the probability that the bandwidth of the offloading node remains unchanged in two sequential decision epochs, $0 \leq \varphi \leq 1$. Alternatively, the available bandwidth of the offloading node varies randomly with probability σ, σ is expressed as follows:

$$\sigma = (1 - \varphi)/j \tag{8}$$

where j is the total number of possible network bandwidth B_i' varing from network bandwidth B_i.

The state transition diagram is shown in Fig. 2. The shaded circles indicate the next state that the mobile device cannot transit into from the current state, because the mobile device is not within the radio radii of the edge server corresponding to the shaded circle. The unshaded circles indicate the next state that the mobile device can transit into from the current state, because the mobile device is within the radio radii of the edge server corresponding to the unshaded circle. Specifically, the mobile device can only offload computing tasks to edge servers corresponding to shaded circles due to communication constraint. A directed edge indicates a transition between

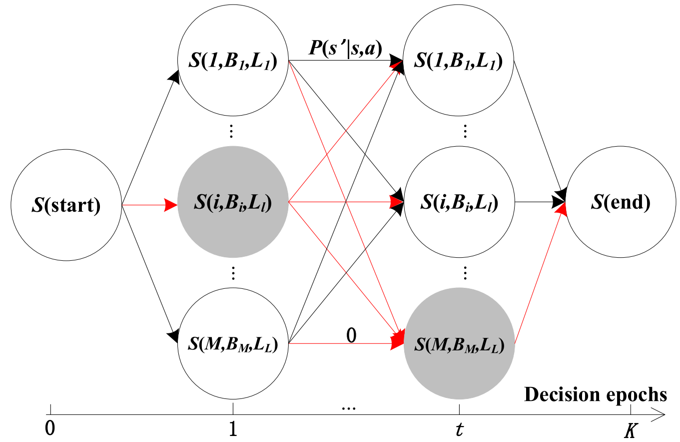

Fig. 2. Transition of the states.

states, a circle connected to the head of the arrow indicates the current state, and a circle connected to the end of the arrow indicates the next state. The weights on edges connecting circles represent the transition probability which is represented by $P(s'|s, a)$. The transition probability associated with a shaded circle equals to 0, and the transition is represented by a redline. The transition probability associated with an unshaded circle is calculated by (4), and the transition is represented by a blackline. In particular, the transition probability is only related to the action selected in the current state, and the transition probability affects the expected rewards of the mobile device. The calculation of expected rewards will be introduced in Section IV-B.

In reality, a mobile device equipped with GPS sensors can periodically update timed geo-location data [30], which can be used to calculate the probability of mobile device moving. In addition, the probability of available bandwidth of offloading node can be predicted by geo-profiles, which is a result of geo-intelligence at each location in a vehicular mobility scenario [31]. Based on this, the transition probability can be applied for computation offloading in practice.

Reward Function: The reward function $F(s, a)$ reflects the immediate reward for the mobile device when it selects an action at the current state. The offloading time is considered to describe the computational performance of the selecting action, here, the reward function $F(s, a)$ is defined as follows:

$$F(s, a) = \begin{cases} T_{\text{Cell}}, & a = 1 \\ T_{\text{WiFi}}, & a = 2 \end{cases} \tag{9}$$

where T_{Cell} and T_{WiFi} represent the offloading time of computing task offloaded to the edge server deployed at femto cell BS and WiFi AP, respectively.

Computation offloading consists of computation task uploading, computation task execution, and result downloading, thus, their corresponding time can be defined as follows.

1) For the Task Uploading Time t^u:

Case 1: If the edge server is deployed at femto cell BS, the task uploading time is given as

$$t_c^u = \frac{D_s}{\left(1 - \frac{d}{Rc_i}\right) * Bc_i}. \tag{10}$$

Case 2: If the edge server is deployed at WiFi AP, the task uploading time is given as

$$t_w^u = \frac{D_s}{\left(1 - \frac{d}{Rw_i}\right) * Bw_i} \tag{11}$$

where D_s represents the uploading data size of the computing tasks. d represents the distance between the mobile device and the edge server. Rc_i and Rw_i represent the radio radii of the edge server i deployed at femto cell BS and WiFi AP, respectively. Bc_i and Bw_i represent the available network bandwidth of edge server i deployed at femto cell BS and WiFi AP, respectively.

In fact, based on the analysis that the signal strength decreases monotonically with distance increasing [32], and the conclusion that the received bandwidth decreases sharply as the link distance increases [33], consequently, these formulas use the distance from the mobile device to the edge server to describe the impact of the location of the mobile device on the available network bandwidth of edge server, which, in turn, affects the transmission time of the computational offloading.

2) For the Task Execution Time t^c:

$$t^c = W/f_i \tag{12}$$

where W represents the total CPU cycles for executing the computing task. f_i represents the CPU clock speed of edge server i selected as offloading node.

3) For the Result Downloading Time t^d:

Case 1: If the edge server is deployed at femto cell BS, the task downloading time is given as

$$t_c^d = \frac{D_r}{\left(1 - \frac{d}{Rc_i}\right) * Bc_i}. \tag{13}$$

Case 2: If the edge server is deployed at WiFi AP, the task downloading time is given as

$$t_w^d = \frac{D_r}{\left(1 - \frac{d}{Rw_i}\right) * Bw_i} \tag{14}$$

where D_r represents the downloading data size of the result.

Therefore, when mobile devices offload the computing task to the edge server deployed at femto cell BS and WiFi AP, the offloading time can be respectively, derived as follows:

$$T_{\text{Cell}} = t_c^u + t^c + t_c^d \tag{15}$$

$$T_{\text{WiFi}} = t_w^u + t^c + t_w^d. \tag{16}$$

In the proposed MDP model, the mobile device can select an action to offload computing task at the current state during a specific decision period. Accordingly, the mobile device can obtain the transition probability associated with the current state and the selected action, then calculate the immediate reward of the specific decision period. In addition, the mobile devices will receive corresponding rewards in the next decision epochs.

B. Optimal Offloading Node Selection

In order to optimize the expected rewards, that is, minimize the offloading time, an OONS strategy is introduced by solving the MDP model employing VIA, as explained Algorithm 1.

The expected rewards consist of the immediate reward and corresponding rewards. In this article, we use the value function $v^\pi(s)$ which describes the expected rewards when the offloading node selection strategy is π and the current state is s. Thus, $v^\pi(s)$ is written as

$$v^\pi(s) = E_s^\pi\left[\sum_{t=1}^{K} \gamma^{t-1} F_t(s, a)\right] \tag{17}$$

where $E_s^\pi[\cdot]$ represents the expectation function to calculate the expected rewards when the offloading strategy is π and the current state is s. $F_t(s, a)$ represents immediate reward at decision epoch t and γ is a discount factor to weight the importance of reward sequence, $\gamma \epsilon (0, 1]$.

Specifically, our objective is minimizing the offloading time by selecting the optimal strategy in δ which is the set of π, here, $v(s)$ represents the minimum expected rewards that can be described as

$$v(s) = \min_{\pi \in \delta} v^\pi(s). \tag{18}$$

The optimization equation and it is expressed by [34]

$$v(s) = \min_{a \in A}\left\{F(s, a) + \gamma \sum_{s' \in S} P\big(s'|s, a\big)v\big(s'\big)\right\}. \tag{19}$$

The optimality equation is widely solved by VIA which has the advantages of easy implement and quick convergence. Thus, we employ VIA to solve the minimum offloading time $v(s)$ and the optimal strategy $\pi^*(s)$

$$\pi^*(s) = \underset{a \in A}{\operatorname{argmin}}\{v(s)\}. \tag{20}$$

Generally, MDP is a mathematically model of reinforcement learning, which is learning how to map states to actions to obtain an optimal reward. The problem of solving an optimality equation for MDP can be described as a linear programming problem, and it can be solved in polynomial time. VIA is devised to compute the optimality equation for MDP, and the running time for each iteration in the VIA is $O(|A||S|^2)$ [35]. In addition, for a given threshold ε which is used to stop value iteration, the VIA can quickly converge by using (19) to obtain an OONS strategy in a finite number of iterations. To cope with the expensive computational complexity of VIA, the calculation of VIA can be preperformed by edge servers or remote cloud. Then, mobile devices store the optimal strategy to offload computing tasks, which includes each state in S and the correspondingly optimal action in A.

C. Offloading Node Failure

In fact, the mobile devices may move beyond the radio radii of the optimal offloading node when they are downloading the results of the computing tasks from the optimal offloading nodes, which will make the computation offloading fail.

In this case, the mobile devices should download the results from other edge servers or remote cloud where the results can

Algorithm 1 VIA

Step 1: Initialize $\forall s \in S$, $v^0(s) = 0$.
Specify $\varepsilon > 0$, and $x = 0$, where ε is a small positive number
Step 2: For $\forall s \in S$
Step 3: For $\forall a \in A$
Step 4:
$v^{x+1}(s) = \min_{a \in A}\{F(s, a) + \gamma \sum_{s' \in S} P(s'|s, a)v^x(s')\}$
//According to (19)
Step 5: If $(|v^{x+1}(s) - v^x(s)| < \varepsilon)$, then
Go to Step 6
Else
$x = x + 1$, and go to Step 2
Step 6: Return $\pi^*(s)$ //According to (20)

be transferred by the optimal offloading nodes. Furthermore, since the offloading nodes communicate with each other or with the remote cloud by wired network, the result transferring time can be calculated as follows:

$$t^t = \frac{D_r}{B_{\text{wired}}} + \frac{d_w}{r} \tag{21}$$

where B_{wired} represents the available network bandwidth of wired network, d_w represents the transfer distance of the result on wired network between the optimal node and other edge servers or remote cloud, and r represents the wave propagation speed of the wired network.

V. Simulations

To evaluate the performance of our OONS strategy, in terms of minimizing offloading time, we use python 3.6 to perform simulations on the Dawn TC4600E Ten Billions Supercomputing System. The main simulation parameters are listed in Table I.

TABLE I
Simulation Parameters

Symbol	Values
K	6
L_G	16
Region	600 m × 600 m
W	60 G cycles
D_s	[200,1000] MB
D_r	100 MB
F	[2,3,5] GHz
Rw	70-350 m
Rc	90-300 m
Bw	5-150 Mbps
Bc	10-200 Mbps
μ	0.6
φ	0.12
γ	0.9

In the simulation, OONS is compared with on-the-spot offloading (OTSO) [36] to evaluate its efficiency. The OTSO spontaneously selects the edge server deployed at WiFi AP to offload computing tasks. Moreover, we adopt greedy strategy and random strategy (RS) as the baseline strategy. In the greedy strategy, we consider two greedy-based offloading node selection strategies. One is the minimum distance strategy (MDS) which refers to selecting offloading nodes by the minimum Euclidean distance between the mobile device and edge servers, and another is the maximum bandwidth strategy (MBS) which selects offloading nodes by the maximum available bandwidth provided by edge servers for the mobile device. The RS selects the offloading nodes randomly to offload computing tasks.

To measure the performance of minimizing offloading time in different strategies, we investigate the impact of uploading data size, number of edge servers and heterogeneity of edge severs. Regarding simulation experiments, the simulation results are calculated by averaging over 50 times.

A. Uploading Data Size

In this part, we study how the uploading data size of computing tasks impact on the offloading time. Fig. 3 shows the relationship between variance of uploading data size and the offloading time under a different ratio of the number of grids to the number of edge servers, which is used to describe the density of edge servers in the region.

As shown in Fig. 3, we can observe that the offloading time increases as the size of uploading data rises from 200 to 1000 MB under (a) ratio = 2:1, (b) ratio = 1:1, and (c) ratio = 1:2. According to the theoretical analysis in Section IV, this can be explained that increasing the size of uploading data increases the task uploading time so the offloading time shows an increasing trend.

From the simulation results, it shows that the offloading time in OONS is lower than that in other comparison strategies. This is because, the OONS considers both the mobility of the mobile device and the heterogeneity of the edge servers. In fact, OONS calculates the possible location of the mobile device and the available network bandwidth of the edge servers according to (5) and (7), respectively. The offloading time in those possible situations are compared to select the optimal offloading nodes. MDS and MBS adopting the idea of greedy algorithms calculate the Euclidean distance or the available network bandwidth separately to offload computing tasks, this is why MDS and MBS can surpass OTSO and RS.

As the ratio of the number of grids to the number of edge servers decreases, that is, increasing the number of edge servers in the region which has constant number of grids, the offloading time in OONS is significantly reduced. The result can be interpreted that OONS can select the offloading node more accurately as the number of states and actions increases in MDP. Furthermore, we can observe that the offloading time in MBS is less than that in MDS in Fig. 3 (c). The reason is that when the density of the edge server increases, the influence of the distance on the offloading time is reduced, but selecting the offloading nodes based on the maximum available network bandwidth can further optimize the offloading time by reducing transmission time. Hence, to sum up, OONS remains best in minimizing the offloading time when the uploading data size increases.

B. Number of Edge Servers

In this part, we investigate the effect of changes in the number of edge servers on the offloading time where the edge

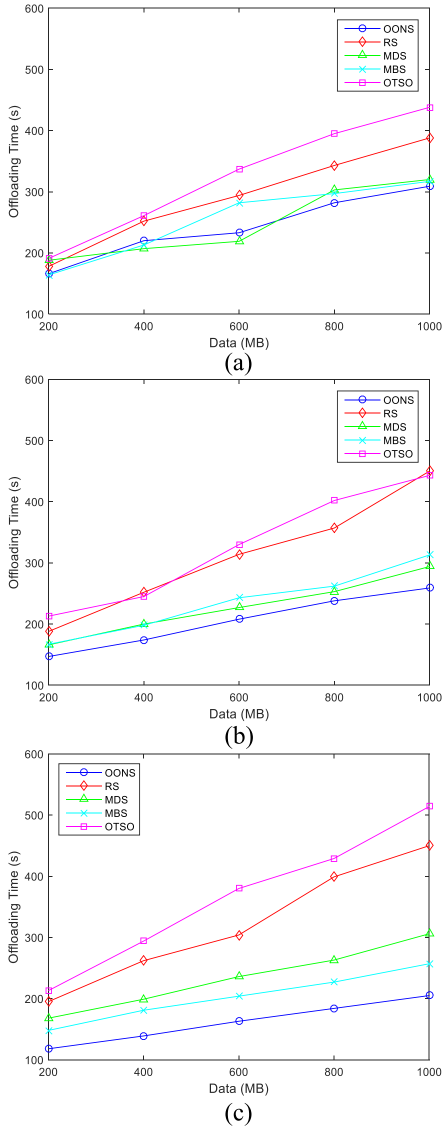

Fig. 3. Impact of uploading data size on offloading time. (a) Ratio = 2:1. (b) Ratio = 1:1. (c) Ratio = 1:2.

servers are distributed in the given region randomly and the size of uploading data is set to 500 MB.

Fig. 4 shows the variety of offloading time in the five strategies as the number of edge servers changes from 8 to 56. In the evaluations, simulations are conducted on four scenarios designed as (a) $Rw < Rc$; $Bw = Bc$, (b) $Rw > Rc$; $Bw = Bc$, (c) $Rw = Rc$; $Bw > Bc$, and (d) $Rw = Rc$; $Bw < Bc$, where R_w and R_c denote the radio radii of the edge server deployed at femto cell BS and WiFi AP, respectively. Bw and Bc represent the available network bandwidth of the edge server deployed at femto cell BS and WiFi AP, respectively.

It can be seen that OONS achieves the best results on minimizing the offloading time in comparison to MDS, MBS, RS, and OTSO. Specifically, the offloading time decreases as the number of edge servers increases in OONS. In addition, MDS and MBS have a slight increment in reducing the offloading time when the number of edge servers increases, whereas, the change of number of edge servers does not have much impact on offloading time in RS and OTSO. One possible reason is

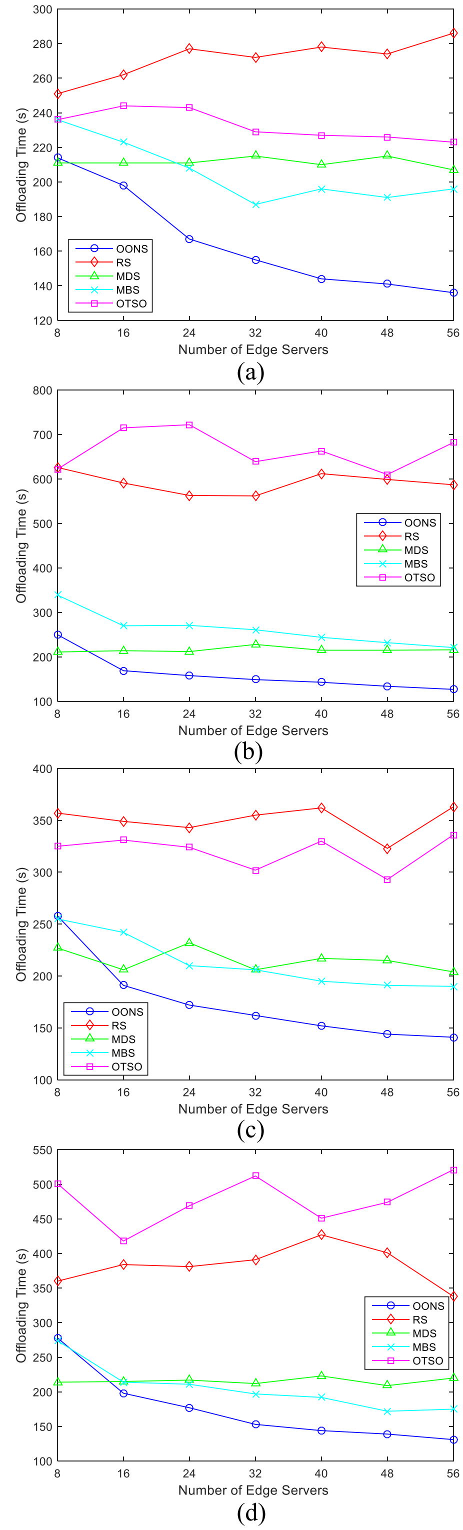

Fig. 4. Influence of number of edge servers on offloading time. (a) $Rw < Rc$; $Bw = Bc$. (b) $Rw > Rc$; $Bw = Bc$. (c) $Rw = Rc$; $Bw > Bc$. (d) $Rw = Rc$; $Bw < Bc$.

that the mobile device can be located within the radio radii of more edge servers as the number of edge servers increase, which will result in increasing the number of edge servers that can communicate with the mobile device. Moreover, as a whole, increasing the number of edge servers will reduce the distance between edge servers and mobile devices, which can reduce the transmission time of the computing tasks, and further reduce the offloading time. These are the reasons why the offloading time shows a decrease trend when the number of edge servers increases in OONS. It is worth noting that the performance of offloading time in MDS outperforms that in OONS when the number of edge servers is set to 8. This is because, the mobile device cannot communicate with edge servers frequently due to the limited radio radii of the edge servers and the movement of the mobile device.

MDS selects the nearest offloading node to offload computing tasks, which guarantees the wireless network connection when the mobile device uploads computing tasks.

Besides, it can be seen from Fig. 4(a) and (c) that the offloading time in OTSO is less than that in RS as the number of edge servers increase, while Fig. 4(b) and (d) show the opposite trend.

This phenomenon can be explained that the mobile devices prefer to select the edge servers deployed at WiFi AP to offload computing tasks so the offloading time highly depends on WiFi network in OTSO.

Therefore, OTSO has higher possibility to select offloading nodes deployed at WiFi AP, which have strong transmission capability to minimize transmission time as the parameter value of *Rw* is set to less than *Rc* and *Bw* is set to a value bigger than *Bc*.

C. Heterogeneity of Edge Servers

To further verify the influence of the heterogeneity of edge servers on the offloading time, the uploading data size is set to 500 MB and the number of edge servers is set to 24.

As can be seen from the figure, the performance of OONS is superior to OTSO, RS, MBS, and MDS in terms of optimal offloading time, respectively. The simulation results can be explained by that OONS formulates an MDP model which is beneficial to select an optimal offloading node in dynamic system, that is, the location of the mobile device and the available network bandwidth of the edge server may vary with time. Comparing *A*, *B*, and *C*, it can be seen that the change of the offloading time in MBS is greater than that in MDS. This is because the mobile device selects the offloading node based only on the available network bandwidth of the edge server in MBS. OTSO selects edge server deployed at WiFi AP to offload computing tasks, but it ignores the difference in the transmission data of the available network bandwidth. This is why the offloading time in OTSO changes significantly in the three scenarios which represent the heterogeneity of the edge servers in available network bandwidth.

Fig. 5(a)–(c) shows the impact of the radio radii on the offloading time. When increasing the radio radii of edge servers deployed at WiFi AP until it is larger than the radio radii of the edge server deployed at femto cell BS, the performance gap of offloading time is more obvious. The reason is that during the movement of the mobile device, the radio radii of the edge server determines whether the mobile device communicates with the edge server, which will affect the selection of the optimal offloading node. Specifically, the Euclidean distance between the mobile device and edge servers changes as the mobile devices move. From (10) and (11), it is known that the ratio of the Euclidean distance to the radio radii of the edge server affects the ability of the edge server to transmit computing tasks, further affecting the calculation of the offloading time. For instance, given an available network bandwidth, the closer the device is to the edge server, the smaller the offloading time.

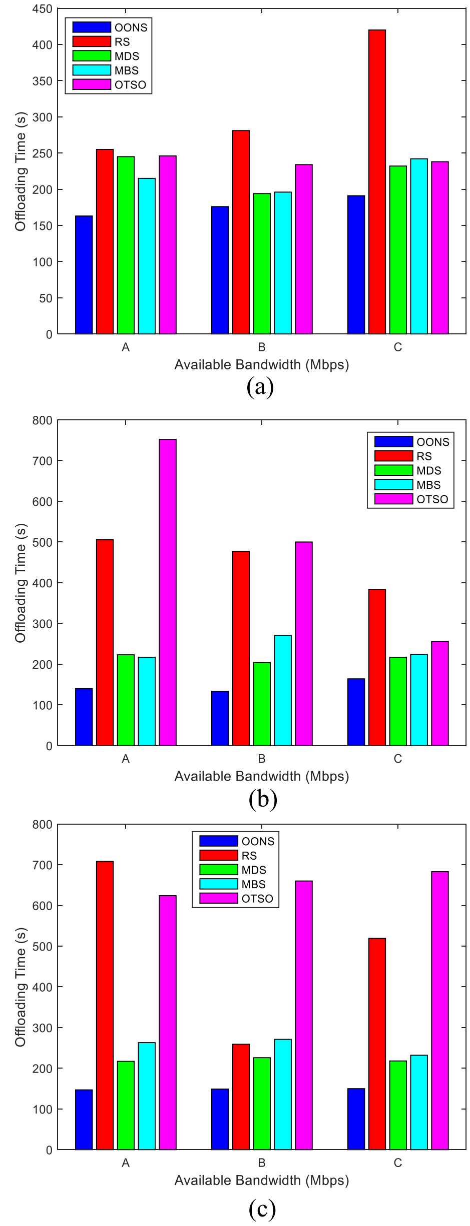

Fig. 5. Influence of the heterogeneity of edge servers on offloading time. (a) $Rw < Rc$. (b) $Rw = Rc$. (c) $Rw > Rc$.

VI. Conclusion

In this article, we have studied the offloading time optimization on computation offloading in MEC, and presented an OONS strategy. To obtain the strategy, an MDP model is formulated by considering the mobility of the mobile device and the heterogeneity of edge servers jointly. Then, VIA is employed to solve MDP and obtain the optimal offloading time. In addition, four comparison strategies were used to validate the flexibility and effectiveness of the proposed strategy in terms of offloading time optimization.

The innovation of this article is exploiting the available network bandwidth of the heterogeneous edge severs and the location prediction of mobile devices to select optimal offloading node. One direction of our future work is to combine the blockchain technology with MEC to improve the security of computation offloading. First, we take the computation offloading as transaction, and discuss how to ensure secure transaction from two aspects. On the one hand, a smart contract, which allows transactions to be conducted in anonymous between edge servers and mobile devices without the need for a third-party authority, can be designed. The smart contract includes the steps of account registration, release of offloading request, response record, computing task execution, computing task verification and payment. On the other hand, we consider designing a permissioned Byzantine consensus protocol based on subjective logic. Specifically, we can calculate the reputation value of each edge server through a multiweight reputation fusion calculation method. Then an edge server with a highest reputation value will be selected in advance as the consensus node who is responsible for auditing, verifying transaction records, and generating blocks. In summary, by combining the above two blockchain technologies, a secure computation offloading solution will be achieved.

REFERENCES

[1] V. Friedman. (Aug. 2018). *On The Edge: Solving The Challenges Of Edge Computing In The Era Of IoT*. [Online]. Available: https://data-economy.com/on-the-edge-solving-the-challenges-of-edge-computing-in-the-era-of-iot/

[2] S. Meng, Y. Wang, Z. Miao, and K. Sun, "Joint optimization of wireless bandwidth and computing resource in cloudlet-based mobile cloud computing environment," *Peer Peer Netw. Appl.*, vol. 11, no. 3, pp. 462–472, May 2018, doi: 10.1007/s12083-017-0544-x.

[3] H. T. Dinh *et al.*, "A survey of mobile cloud computing: Architecture, applications, and approaches," *Wireless Commun. Mobile Comput.*, vol. 13, no. 18, pp. 1587–1611, Dec. 2013, doi: 10.1002/wcm.1203.

[4] X. Wang, Y. Han, C. Wang, Q. Zhao, X. Chen, and M. Chen, "In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning," *IEEE Netw.*, vol. 33, no. 5, pp. 156–165, Sep./Oct. 2019, doi: 10.1109/MNET.2019.1800286.

[5] D. Xu *et al.*, "A survey of opportunistic offloading," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2198–2236, 3rd Quart., 2018, doi: 10.1109/COMST.2018.2808242.

[6] M. B. Threfe, H. Lee, N. Heo, G. C. Fox, and S. Oh, "Energy-efficient multisite offloading policy using Markov decision process for mobile cloud computing," *Pervasive Mobile Comput.*, vol. 27, pp. 75–89, Apr. 2016, doi: 10.1016/j.pmcj.2015.10.008.

[7] Y. Wu, K. Ni, C. Zhang, L. P. Qian, and D. H. K. Tsang, "NOMA-assisted multi-access mobile edge computing: A joint optimization of computation offloading and time allocation," *IEEE Trans. Veh. Technol.*, vol. 67, no. 12, pp. 12244–12258, Dec. 2018, doi: 10.1109/TVT.2018.2875337.

[8] C. S. M. Babou, D. Fall, S. Kashihara, I. Niang, and Y. Kadobayashi, "Home edge computing (HEC): Design of a new edge computing technology for achieving ultra-low latency," in *Proc. Int. Conf. Edge Comput. (EDGE)*, Seattle, WA, USA, 2018, pp. 13–27.

[9] W. Labidi, M. Sarkiss, and M. Kamoun, "Energy-optimal resource scheduling and computation offloading in small cell networks," *presented at the 22nd Int. Conf. Telecommun. (ICT)*, Sydney, NSW, Australia, Apr. 2015, pp. 313–318.

[10] H. Ko, J. Lee, and S. Pack, "Spatial and temporal computation offloading decision algorithm in edge cloud-enabled heterogeneous networks," *IEEE Access*, vol. 11, pp. 18920–18932, 2018, doi: 10.1109/access.2018.2818111.

[11] C. Zhu, Y.-H. Chiang, A. Mehrabi, Y. Xiao, A. Ylä-Jääski, and Y. Ji, "Chameleon: Latency and resolution aware task offloading for visual-based assisted driving," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 9038–9048, Sep. 2019, doi: 10.1109/TVT.2019.2924911.

[12] X. W. Yang, X. Wang, Y. Wu, L. P. Qian, W. Lu, and H. Zhou, "Small-cell assisted secure traffic offloading for narrowband Internet of Thing (NB-IoT) systems," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1516–1526, Jun. 2018, doi: 10.1109/JIOT.2017.2779820.

[13] A. Mamane, M. El Ghazi, S. Mazer, M. Bekkali, M. Fattah, and M. Mahfoudi, "The impact of scheduling algorithms for real-time traffic in the 5G femto-cells network," in *Proc. IEEE 9th Int. Symp. Signal Image Video Commun. (ISIVC)*, Rabat, Morocco, Nov. 2018, pp. 147–151, doi: 10.1109/ISIVC.2018.8709175.

[14] Z. Y. Jiang *et al.*, "Dependency-aware task scheduling in vehicular edge computing," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4961–4971, Jun. 2020, doi: 10.1109/JIOT.2020.2972041.

[15] L. Q. Liu, C. Zheng, X. J. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 283–294, Feb. 2018, doi: 10.1109/JIOT.2017.2780236.

[16] Q. Tang, R. Xie, F. R. Yu, T. Huang, and Y. Liu, "Decentralized computation offloading in IoT fog computing system with energy harvesting: A dec-POMDP approach," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 4898–4911, Jun. 2020, doi: 10.1109/JIOT.2020.2971323.

[17] M. G. R. Alam, M. Mehedi, H. M. Z. Uddin, A. Almogren, and G. Fortino, "Autonomic computation offloading in mobile edge for IoT applications," *Future Gener. Comput. Syst.*, vol. 90, pp. 149–157, Jan. 2019, doi: 10.1016/j.future.2018.07.050.

[18] Y. Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Area Commun.*, vol. 34, no. 12, pp. 3590–3605, Dec. 2016, doi: 10.1109/JSAC.2016.2611964.

[19] K. Q. Li, "Computation offloading strategy optimization with multiple heterogeneous servers in mobile edge computing," *IEEE Trans. Sustain. Comput.*, early access, Mar. 12, 2019, doi: 10.1109/TSUSC.2019.2904680.

[20] D. V. Le and C.-K. Tham, "A Deep Reinforcement Learning based Offloading Scheme in Ad-hoc Mobile Clouds," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Honolulu, HI, USA, 2018, pp. 760–765.

[21] Y. Zhang, D. Niyato, and P. Wang, "Offloading in mobile cloudlet systems with intermittent connectivity," *IEEE Trans. mobile Comput.*, vol. 14, no. 12, pp. 2516–2529, Dec. 2015, doi: 10.1109/TMC.2015.2405539.

[22] J. Zhang *et al.*, "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2633–2645, Aug. 2018, doi: 10.1109/JIOT.2017.2786343.

[23] K. Zhang *et al.,* "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE Access*, vol. 4, pp. 5896–5907, 2016, doi: 10.1109/ACCESS.2016.2597169.

[24] D. Li, Y. Han, C. Wang, G. Shi, X. Wang, X. Li, and V. C. M. Leung, "Deep reinforcement learning for cooperative edge caching in future mobile networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Marrakesh, MA, USA, Apr. 2019, pp. 1–6.

[25] H. Zhang, M. Feng, K. Long, G. K. Karagiannidis, V. C. M. Leung, and H. V. Poor, "Energy efficient resource management in SWIPT enabled heterogeneous networks with NOMA," *IEEE T Wirel Commun*, vol. 19, no. 2, pp. 835–845, Feb. 2020, doi: 10.1109/TWC.2019.2948874.

[26] Z. Kuang, L. Li, J. Gao, L. Zhao, and A. Liu, "Partial offloading scheduling and power allocation for mobile edge computing systems," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6774–6785, Aug. 2019.

[27] S. Zhou, Y. Sun, Z. Jiang, and N. Niu, "Exploiting moving intelligence: Delay-optimized computation offloading in vehicular fog networks," *IEEE Commun. Mag.*, vol. 57, no. 5, pp. 49–55, May 2019, doi: 10.1109/MCOM.2019.1800230.

[28] D. Q. Liu, L. Khoukhi, and A. Hafid, "Prediction-based mobile data offloading in mobile cloud computing," *IEEE Trans. Wireless Commun.*, vol. 17, no. 7, pp. 4660–4673, Jul. 2018, doi: 10.1109/TWC.2018.2829513.

[29] Y. Zhang, H. Liu, L. Jiao, and X. Fu, "To offload or not to offload: An efficient code partition algorithm for mobile cloud computing," in *Proc. IEEE 1st Int. Conf. Cloud Netw. (CLOUDNET)*, Paris, France, Nov. 2012. pp. 80–86.

[30] V. A. Siris and D. Kalyvas, "Enhancing mobile data offloading with mobility prediction and prefetching," in *Proc. 7th ACM Int. Workshop Mobility Evol. Internet Archit. (MobiArch)*, Istanbul, Turkey, Aug. 2012, pp. 17–22.
[31] J. Yao, S. S. Kanhere, and M. Hassan, "Geo-intelligent traffic scheduling for multi-homed on-board networks," in *Proc. 4th ACM Int. Workshop Mobility Evol. Internet Archit. (MobiArch)*, Kraków, Poland, Jun. 2009, pp. 1–6.
[32] C. Liu, T. Scott, K. Wu, and D. Hoffman, "Range-free sensor localisation with ring overlapping based on comparison of received signal strength indicator," *Int. J. Sens. Netw.*, vol. 2, nos. 5–6, pp. 399–405, 2007, doi: 10.1504/ijsnet.2007.014364.
[33] A. Sheth, S. Nedevschi, R. Patra, S. Surana, E. Brewer, and L. Subramanian, "Packet loss characterization in WiFi-based long distance networks," in *Proc. IEEE INFOCOM Int. Conf. Comput. Commun.*, Barcelona, Spain, May 2007, pp. 312–320.
[34] M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY, USA: Wiley, 1994.
[35] M. L. Littman, T. L. Dean, and L. P. Kaelbling, "On the complexity of solving markov decision problems," in *Proc. 11th Int. Comf. Uncertainty Artif. Intell.*, San Francisco, CA, USA, 1995, pp. 394–402.
[36] K. Lee, J. Lee, Y. Yi, I. Rhee, and S. Chong, "Mobile data offloading: How much can WiFi deliver?" in *Proc. 6th Int. Conf.*, New York, NY, USA, 2010, pp. 1–12.

Guisong Yang (Member, IEEE) received the Ph.D. degree in control theory and control engineering from Tongji University, Shanghai, China, in 2013.

He worked as a Research Scholar with Michigan State University, East Lansing, MI, USA, from 2009 to 2011. He is currently an Associate Professor with the Department of Computer Science and Engineering, University of Shanghai for Science and Technology, Shanghai. His research interests include Internet of Things and pervasive computing, delay-tolerant and opportunistic networks, and mobile crowdsensing.

Dr. Yang is a Member of CCF.

Ling Hou received the B.S. degree in computer science and technology from Hunan University of Arts and Science, Changde, China, in 2017. She is currently pursuing the master's degree in computer technology with the University of Shanghai for Science and Technology, Shanghai, China.

Her current research interest is mobile edge computing.

Xingyu He received the Ph.D. degree in control theory and control engineering from Tongji University, Shanghai, China, in 2017.

She is an Assistant Professor with the College of Communication and Art Design, University of Shanghai for Science and Technology, Shanghai. Her research interests include wireless sensor networks and pervasive computing, delay tolerant networks, incentive scheme, swarm intelligence, and mobile crowdsensing.

Dr. He is a Member of CCF.

Daojing He (Member, IEEE) received the B.Eng. and M.Eng. degrees in computer science from Harbin Institute of Technology, Harbin, China, in 2007 and 2009, respectively, and the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2012.

He is currently a Professor with the School of Computer Science and Software Engineering, East China Normal University, Shanghai, China. His research interests include network and systems security.

Prof. He is on the Editorial Board of some international journals, such as *IEEE Communications Magazine*.

Sammy Chan (Member, IEEE) received the B.E. and M.Eng.Sc. degrees in electrical engineering from the University of Melbourne, Melbourne, VIC, Australia, in 1988 and 1990, respectively, and the Ph.D. degree in communication engineering from the Royal Melbourne Institute of Technology, Melbourne, in 1995.

Since December 1994, he has been with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong, where he is currently an Associate Professor.

Mohsen Guizani (Fellow, IEEE) received the B.S. (with Distinction) and M.S. degrees in electrical engineering and the M.S. and Ph.D. degrees in computer engineering from Syracuse University, Syracuse, NY, USA, in 1984, 1986, 1987, and 1990, respectively.

He is currently a Professor with the CSE Department, Qatar University, Doha, Qatar.

Prof. Guizani is a Senior Member of ACM.