IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON FOG RADIO ACCESS NETWORKS (F-RANS) FOR 5G: RECENT ADVANCES AND FUTURE TRENDS

Received February 26, 2019, accepted March 24, 2019, date of publication March 29, 2019, date of current version April 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2908263

Mobility-Aware Task Offloading and Migration Schemes in Fog Computing Networks

DONGYU WANG, ZHAOLIN LIU, XIAOXIANG WANG, AND YANWEN LAN

Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Dongyu Wang (dy_wang@bupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61701038, and in part by the Fundamental Research Funds for the Central Universities.

ABSTRACT Fog computing is an extension of cloud computing, which emphasizes distributed computing and provides computing service closer to user equipments (UEs). However, due to the limited service coverage of fog computing nodes (FCNs), the moving users may be out of the coverage, which would cause the radio handover and execution results migration when the tasks are off-loaded to FCNs. Furthermore, extra cost, including energy consumption and latency, is generated and affects the revenue of UEs. Previous works rarely consider the mobility of UEs in fog computing networks. In this paper, a generic three-layer fog computing networks architecture is considered, and the mobility of UEs is characterized by the sojourn time in each coverage of FCNs, which follows the exponential distribution. To maximize the revenue of UEs, the off-loading decisions and computation resource allocation are jointly optimized to reduce the probability of migration. The problem is modeled as a mixed integer nonlinear programming (MINLP) problem, which is NP-hard. The problem is divided into two parts: tasks off-loading and resource allocation. A Gini coefficient-based FCNs selection algorithm (GCFSA) is proposed to get a sub-optimal off-loading strategy, and a distributed resource optimization algorithm based on genetic algorithm (ROAGA) is implemented to solve the computation resource allocation problem. The proposed algorithms can handle the scenario of UEs' mobility in fog computing networks by significantly reducing the probability of migration. Simulations demonstrate that the proposed algorithms can achieve quasi-optimal revenue performance compared with other baseline algorithms.

INDEX TERMS Fog computing, mobility, Gini coefficient, resource optimization, utility function.

I. INTRODUCTION

The rapid development of mobile communication technology has motivated a plenty of novel service coming true (e.g., augmented reality (AR), Internet of Things (IoT), Internet of Vehicles (IoV), intelligent camera, etc). Most of the services exhaust the capabilities of current wireless systems, including the throughput, fronthaul/backhaul capacity, bandwidth capacity and computation capacity, which arouses an increasing number of scholars to promote the development of the fifth generation (5G) wireless systems. User equipments (UEs) themselves cannot satisfy their requirements because of the limited computation resource and battery power. With the arise of the cloud radio access networks (C-RANs), strong computing capability are available such that the terminal nodes can compute their tasks locally or remotely [1]. The idea of moving the local load such as computing and caching from UEs to the cloud servers is proposed to improve the spectral efficiency and energy efficiency performance of the system in a centralized manner. Nevertheless, due to the remote location, it should be noted that the constrained bandwidth with limited capacity and long time transmission make it difficult for the traditional central cloud to support some services like the latency-sensitive ones.

In order to solve these problems, the concept of fog radio access networks (F-RANs) is proposed [2] as a middle computing layer between the UEs and C-RANs. The key idea behind fog computing is the distributed computing among the UEs. Since fog computing nodes (FCNs) are located close to UEs, the offloading process for execution can significantly reduce the communication delay, save backhaul bandwidth between FCNs and cloud servers (CSs). So the structure of FCNs network can improve the network capacity [3].

So far, many researchers focus on fog computing to promote the performance in terms of energy efficiency and time delay of F-RAN. Some papers focus on the tasks offloading

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

and resource allocation optimization in the scenario of fog computing [4]–[11]. The works in [4]–[7] focus on the offloading strategies, of which papers [7] and [8] mainly lay emphasis on the partial offloading strategy in fog computing. The authors of [9]–[11] jointly optimize the offloading decisions and resource allocation. Cooperation and matching problems in three-layer fog computing architecture networks are studied in [12]–[19].

However, the papers mentioned above mostly ignore the mobility of UEs. Due to the mobility, UEs will cause the radio handover and service migration when moving out of the coverage of a FCN. The process of migration may cause additional cost of energy consumption and time delay, and the migration failure may occur due to late migration or early migration or migration to FCN with much higher latency. Besides, the target FCN may not have adequate available capacities to support new UEs, so the new migrated tasks should wait in line for execution, which increases the time delay and occupies the storage space.

A few papers in fog computing investigate the mobility of UEs [20]–[25]. Besides, the mobility-aware computation offloading in vehicular networks has been widely investigated in many papers [26]–[28]. Especially, the authors in [28] leverage a follow-me edge concept for enabling lightweight live migration which means services should follow the user mobility. To ensure the service continuity after migration, they proposed three mechanisms to improve the end user experience which consider the moving path of mobile users like vehicles, UAVs. But as we know, the movement of mobile users is generally irregular and unpredictable in most of the time, especially in scenes with complex bewilderment of passages. Thanks to the development of machine learning, the user's preference stay time in a certain area is available, which provided a basis for task offloading and task migration.

Although there are some ways to reduce the cost of migration, such as live migration [28], [36], the process of migration still causes the extra cost more or less. Unlike the previous papers, this paper explores the ways on how to avoid the extra migration, which means completing the execution of the task in the current FCN before the UE leaving the coverage, by integrating the mobility and migration. Therefore, motivated by the previous works in [20]–[28], we propose a mobility-aware task offloading and migration scheme in a generic three-layer computing networks architecture in this paper. We intend to minimize the probability of migration in order to maximize the revenue of UEs by optimizing the offloading decision, FCNs selection and computation resource allocation. The problem is modeled as a Mixed Integer Nonlinear Programming (MINLP) problem, which is NP-hard. The problem is divided into two parts and two algorithms are proposed to solve the sub-problems separately. The distinct contributions of this paper are as follows:

- A mobility-aware task offloading and migration scheme is proposed based on a generic multilayer fog system [29]. Since leaving the coverage of FCNs without finishing the offloaded tasks will lead to the migration of the execution results. The offloading and computation resource allocation are jointly optimized based on the UEs' mobility to reduce the probability of migration.
- A fog computing model with mobility is proposed based on the sojourn time in each coverage of FCNs, which follows the exponential distribution. The mobility model is motivated by [30] and [31], which only considered the current state of the number of UEs. The mobility model of UEs is used to predict the sojourn time in order to reduce the probability of migrations.
- A utility function considering the energy consumption and computation delay is derived when executing the computation tasks. The revenue that a UE can realize by offloading its task to the FCN is defined by the utility function. Therefore, the problem is formulated as a revenue maximization problem for each UE. We define that the revenue gain can be realized from computation offloading compared with local computing.
- The problem is formulated as a MINLP problem, which is also an NP-hard problem. The problem is divided into two sub-problems. A Gini Coefficient based FCN selection algorithm (GCFSA) is proposed to optimize the offloading decisions, and a distributed resource optimization algorithm based on genetic algorithm (ROAGA) is implemented to solve the optimization problem of computation resource allocation.
- Extensive simulations show the performance of the proposed scheme compared with the other baseline schemes, which demonstrates that the proposed scheme considering mobility can significantly reduce the probability of migration and improve the revenue of UEs.

II. RELATED WORKS

A. JOINTLY CONSIDER THE OFFLOADING AND RESOURCE ALLOCATION IN FOG COMPUTING NETWORKS

The main focus on fog computing is offloading decisions and resource allocation. A socially aware dynamic computation offloading scheme is proposed in [4] by using a game theoretic approach to minimize the social group execution cost in fog computing system with energy harvesting devices. By using lexicographic max-min fairness, delay-aware task offloading is studied in [5]. Besides, a two-step fair task offloading (FTO) scheme is proposed in [6], aiming at decreasing the energy consumption and task delay in the fog computing networks. A partial offloading approach based on edge computing is proposed in [7] with the goal of reducing fog node energy consumption, average task delay and increasing network lifetime. A suboptimal partial offloading technique is proposed in [8] with the goal of improving network lifetime and reducing energy consumption and task processing delay.

Jointly optimizing the offloading decisions and resource allocation is investigated in the following papers. To tackle a joint radio and computational resource allocation problem, student project allocation (SPA) game and user-oriented cooperation (UOC) are used in [9]. A joint computation

offloading and resource allocation method is proposed in [10] under the strict required latency in cloud based two-tier wireless heterogeneous network (HetNet). The offloading decisions and the allocation of computation resource, transmit power and radio bandwidth are optimized jointly in [11].

B. COOPERATION AND MATCHING PROBLEMS IN THREE-LAYER FOG COMPUTING ARCHITECTURE NETWORKS

A generic fog computing architecture includes three layers, UEs, fog nodes and cloud servers. Cooperation among fog nodes and cloud servers draw a lot of attention. In order to reduce delay in a mixed fog and cloud computing system, a low-complexity iterative suboptimal algorithm called FAJORA is proposed in [12], jointly optimizing offloading decision making and resource allocation. A delay-minimizing collaboration and offloading policy for fog-capable devices is proposed in [13]. And a near-optimal resource allocation mechanism is proposed in [14] to improve the quality of experience for users. Reference [15] proposes a maximal energy-efficient task scheduling (MEETS) algorithm utilizing the effective collaborations among neighboring fog nodes via cognitive spectrum access techniques. The interaction between the cloud/fog providers and the miners in a proof of work-based block chain network is studied in [16] using a game theoretic approach.

In the fog computing networks, fog nodes are commonly distributed among the UEs. So the matching problem between UEs and fog nodes is also an issue to be solved. A multi-objective optimization algorithm is proposed in [17] with the consideration of queuing theory in fog computing system. Reference [18] introduces a deferred acceptance algorithm with matching game, which lessens the worst total completion time, mean waiting and mean total completion time per task. And [19] uses Stackelberg game and many-to-many matching game to jointly optimize the utility of all fog nodes.

C. MOBILITY-AWARE IN FOG COMPUTING NETWORKS

Mobility is a significant issue of fog computing networks. Some papers study the problems of UEs' mobility in vehicular networks and D2D networks. Some papers study the mobility prediction and matching problems. A service popularity-based smart resources partitioning (SPSRP) scheme and a mobility and heterogeneity-aware partitioning algorithm are proposed in [20]. By integrating bandwidth and the mobility of robots with the offloading decisions, meanwhile, using genetic algorithm, an offloading method is proposed in [21] to improve QoS and minimum consumption of resources. A hybrid mobile task offloading method is proposed in [22], including local execution, D2D and cloud execution, to minimize the total task execution cost. The authors in [23] investigate the problem of mobility-assisted opportunistic computation offloading by exploiting the contact patterns regulated by these devices' mobility to determine the amounts of computation to be offloaded to other devices. The authors in [24] propose a mobile access prediction algorithm based on tail matching subsequence. Based on the predicted result, a mobility-aware offloading decision method is proposed considering the job completion time, energy consumption and offloading success rate. The works in [25] investigate the task offloading problem in a mobile edge network in order to reduce the latency with the consideration of the task properties, the user mobility and network constraints.

Besides, in vehicular networks, the authors in [26] conceive the idea of utilizing vehicles as the infrastructures and proposed a Vehicular Fog Computing (VFC) architecture. The authors discussed four scenarios of utilizing moving and parked vehicles as communication and computational infrastructures and carry on a quantitative analysis of the capacities of VFC. The works of [27] propose a solution for latency and quality optimized task allocation in VFC. A joint optimization problem is formulated and solved with a trade-off the service latency and quality loss.

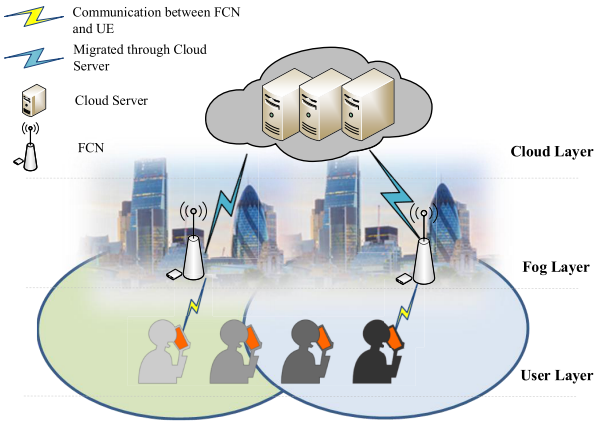

FIGURE 1. An illustration of the UEs mobility in the three-layer fog computing networks.

III. SYSTEM MODEL

A generic three-layer fog computing network architecture is illustrated in Fig. 1, including the cloud layer, fog layer and the user layer. A large number of UEs are in the user layer, which have the mobility and require large amount of computation resource. Fog layer also consists of a large scale of fog computing nodes (FCNs), which are widely deployed among the users. So they can provide the computation resource to users in close proximity. But due to the limited resource and service coverage, only a few UEs can be served. Cloud servers (CSs) lie in the cloud layer, which are in charge of the FCNs. As shown in Fig. 1, the mobility of UEs is considered, and due to the limited service coverage of FCNs, a moving UE has the chance to leave coverage of the serving FCN. If the tasks have already been offloaded to the FCN but still unfinished when the UE left the coverage of FCN, the execution results will be migrated to another FCN through the CS, which will cause the extra cost, including the energy consumption and time delay.

In this paper, the migration process refers to migrating the results of tasks from the previous FCN to the FCN which

TABLE 1. Parameter notations.

Symbol	Definition
U	The number of UEs
F	The number of FCNs
N_u	The set of UEs, $N_u = \{1, ..., U\}$
N_F	The set of FCNs, $N_F = \{1, ..., F\}$
S	The set of offloading decision of UEs
A	The set of FCN selection of UEs
K	Number of sub-channels
U_f	The set of UEs offloading to FCN f
D_u	The data size of the task for UE u
f_u	The required computation resource for UE u
T_u^{max}	The maximum latency for UE u
λ_u^E	The weight coefficient of energy consumption
λ_u^T	The weight coefficient of time delay
P_u	The transmission power for UE u
g_{uf}	The sub-channel gain from UE u to FCN f
σ^2	The background noise variance
r_{uf}	The transmission rate for UE u to FCN f
c_u^{local}	local computation capacity of UE u
c^F	The total computing capacity on FCN
c_{uf}^F	The computation resource allocated to UE u on FCN f
C_f	The computation resource allocation strategy for FCN f
τ_{uf}	The average sojourn time of UE u for FCN f
ψ_{uf}	The income of UE u for FCN f defined in Definition 2
η_{uf}	The weight factor of UE u's QoE defined in Definition 2

can connect with the UEs. The computation results will be transmitted from the previous FCN to the cloud server and then transmitted to the destination FCN. The process is in charged by the cloud server. And due to the small size of the results, the migration process won't overload the relay node.

A. FOG COMPUTING ARCHITECTURE

In the proposed system, UEs are widely distributed around the FCNs, which follow the random distribution. Set $N_F = \{1, \ldots, F\}$ to represent the set of FCNs in networks. And the set of UEs can be expressed as $N_u = \{1, \ldots, U\}$. Each UE is assumed to have only one task to operate at the same time. Each computation task Z_u is characterized by a tuple of three parameters, $\{D_u, f_u, T_u^{\max}\}, \forall u \in N_u$, in which, D_u is the data size of the task necessary to transmit for the program execution (including the input parameters, program codes and system settings) from UEs to FCNs, and $f_u = \varepsilon D_u$ specifies the required computation resource to complete the task. Furthermore, $T_u^{\max}$ represents the maximum latency allowed for Z_u to be completed. In this paper, the computation resource is characterized by the number of CPU cycles [32]. Each task of UEs can be executed locally or offloaded to FCNs. Denote $S = \{s_u, u \in N_u\}$ to be the set of the offloading decisions, in which $s_u = \{0, 1\}$. $s_u = 1$ means that the task Z_u will be offloaded to one of the FCN to execute and $s_u = 0$ specifies the task being operated locally at UE. In this paper, we consider the system with OFDMA as the multiple access scheme in the uplink [39]. Assume that one FCN can connect with multi UEs simultaneously as long as they are in the service coverage of the FCN. Each UE can only connect with one FCN at the same time even if it is in the overlap service coverage. Therefore, set $a_{uf} = 1$ to represent that UE u choose to offload its task to FCN f. Define the ground set A that contains all the FCN selection variables as $A = \{a_{uf} | u \in N_u, f \in N_F\}$. A feasible selection policy must satisfy the constraints below

$$\sum\nolimits_{u \in N_u} a_{uf} \leq K \tag{1}$$

$$\sum\nolimits_{f \in N_F} a_{uf} \leq 1 \tag{2}$$

where K is the maximum number of sub-channels of each FCN. Therefore, the number of UEs that each FCN can serve is at most K at the same time. Besides, (2) constrains that each UE can only connect with one FCN. Since the OFDMA is considered in the system. There are several uplink sub-channels accessed to the FCN, and they are orthogonality to each other. Each UE can only be allocated with one sub-channel to avoid the interference. Additionally, let $U_f = \{u \in N_u | a_{uf} = 1\}$ be the set of UEs offloading their tasks to FCN f to execute. Furthermore, set $U_{off} = \cup_{f \in N_F} U_f$ as the set of UEs that offload their tasks.

Consider the revenue of UEs when executing the computation tasks on FCNs. The revenue of UEs can be realized by offloading the tasks to the FCNs. We define that the revenue gain can be realized from computation offloading compared with local execution, where the cost of executing a task consists of energy consumption and computation delay. Next, the local computation model and FCN computation model will be discussed separately.

B. LOCAL COMPUTATION MODEL

Once UE u decides to perform its task locally on its own CPU, i.e., $s_u = 0$, the energy consumption E_u^{local} and the executing time T_u^{local} can be expressed as follows. Set c_u^{local} as the local computing capacity of UE u, i.e., CPU cycles. So far, the local executing time T_u^{local} can be given by

$$T_u^{local} = \frac{f_u}{c_u^{local}}, \quad \forall u \in N_u \tag{3}$$

To obtain the energy consumption of a UE when its task is performed locally, a widely adopted calculation model of the energy consumption per CPU cycle is used as $E = \kappa c^2$ [33], [34], where κ is the effective switched capacitance depending on the chip architecture [35] and c is the required computation resource, which is the chip frequency. Then, based on the model, the energy consumption E_u^{local} of the task Z_u in such case is calculated as

$$E_u^{local} = \kappa \left(c_u^{local}\right)^2 f_u, \quad \forall u \in N_u \tag{4}$$

After that the corresponding overhead of local execution can be derived as

$$q_u^{local} = \lambda_u^E E_u^{local} + \lambda_u^T T_u^{local}, \quad \forall u \in N_u \tag{5}$$

where λ_u^E and λ_u^T represent the constant weight coefficient of energy consumption and time delay for the task Z_u of UE u to make offloading decisions, respectively. Assume that $\lambda_u^E + \lambda_u^T = 1$ and $\lambda_u^E, \lambda_u^T \in [0, 1]$. These coefficients depend on the UEs and the type of tasks. If $\lambda_u^E < \lambda_u^T$, it means that the task Z_u of UE u is more sensitive to the executing delay. Such tasks

includes video streams, communication online, etc. On the other hand, if the battery of UE u is at a very low state, λ_u^E is supposed to be larger than λ_u^T, even $\lambda_u^E = 1$ under the extreme case.

C. FOG COMPUTATION MODEL

In this case, when UE u offloads its task Z_u to one of the FCNs, the total processing delay T_{uf}^F consists of: (i) the transmission time t_{uf}^{up} from UE u to the FCN f on the uplink, (ii) the executing time t_{uf}^{exe} to deal with the task Z_u at FCN f, and (iii) the transmission time from the FCN back to the UE on the downlink. In this paper, the size of the execution result is assumed to be generally much smaller than the input, and the channel condition of downlink is much better than that of the uplink. Therefore, the delay of transmitting the execution result back to the UE can be omitted, which are also considered in [33]. Specially, the proposed algorithm in this paper can still be appropriate for the cases when the delay of the transmission on the downlink can't be ignored, only if the downlink channel conditions and the size of execution results are given.

Each FCN can serve multiple UEs at the same time. Each UE can have access to multiple FCNs only if the SINR of transmission from UE u to FCN f satisfies the following constraint

$$SINR_{uf} \geq SINR^{th} \tag{6}$$

where $SINR^{th}$ is the threshold to guarantee the revenue of UEs. Therefore, the set of UEs which can be accessed to FCN f is denoted as $U_f^c = \left\{u \in N_u | SINR_{uf} \geq SINR^{th}\right\}$. Suppose that UEs transmit their tasks to FCNs with the maximum transmission power, i.e., $P = \{P_u | u \in N_u\}$. The sub-channel gain on the uplink from UE u to FCN f is g_{uf}, and the set G can be defined as the set of all sub-channels between the UEs and FCNs, $G = \left\{g_{uf} | \forall u \in N_u, \forall f \in N_F\right\}$. Then, the expression of $SINR_{uf}$ between UE u and FCN f is given as

$$SINR_{uf} = \frac{P_u g_{uf}}{\sigma^2 + \sum_{i \in N_F \setminus \{f\}} \sum_{j \in U_i} a_{ji} P_j g_{ji}} \tag{7}$$

where σ^2 is the background noise variance, which is considered as the Gaussian white noise. The second term at the denominator represents the accumulated interference from all the UEs accessed to other FCNs on the same sub-channel. Therefore, the transmission rate r_{uf} can be expressed as

$$r_{uf} = w \log_2(1 + SINR_{uf}), \quad \forall u \in N_u, \ \forall f \in N_F \tag{8}$$

where w is the bandwidth of sub-channel. So far, when the task of UE u is offloaded to FCN f, the transmission time t_{uf}^{up} can be derived as

$$t_{uf}^{up} = \frac{D_u}{r_{uf}}, \quad \forall u \in N_u, \ \forall f \in N_F \tag{9}$$

Consider that the total computing capacity of each FCN is c^F, and let $C_f = \left\{c_{uf}^F | u \in U_f\right\}$ be the computation resource allocation strategy on FCN f. Furthermore, set $C_{off} = \cup_{f \in N_F} C_f$ as the set of the total allocation strategies. Therefore, the constraint of c_{uf}^F must satisfy the following expression

$$\sum\nolimits_{u \in U_f} c_{uf}^F \leq c^F, \quad \forall f \in N_F \tag{10}$$

And the executing delay t_{uf}^{up} can be given as

$$t_{uf}^{exe} = \frac{f_u}{c_{uf}^F}, \quad \forall u \in N_u, \ \forall f \in N_F \tag{11}$$

Therefore, the total processing time on FCN f is derived as

$$T_{uf}^F = t_{uf}^{up} + t_{uf}^{exe}, \quad \forall u \in N_u, \ \forall f \in N_F \tag{12}$$

Furthermore, the energy consumption on transmission can be expressed as

$$E_{uf}^{up} = P_u t_{uf}^{up}, \quad \forall u \in N_u, \ \forall f \in N_F \tag{13}$$

Thus the total cost of performing the task of UE u on FCN f, i.e., q_{uf}^F is given as

$$q_{uf}^F(c_{uf}^F) = \lambda_u^E E_{uf}^{up} + \lambda_u^T T_{uf}^F, \quad \forall u \in N_u, \ \forall f \in N_F \tag{14}$$

D. FOG COMPUTING MODEL WITH MOBILITY

Consider the limited service coverage of FCNs and the mobility of UEs, the sojourn time, which represents the mobility of UEs, in the coverage of different FCNs are limited and different. According to [30], [31], the sojourn time of UE is modeled by an exponential function. Set t_{uf}^s as the sojourn time of UE u in the coverage of FCN f. Therefore, the probability density function (PDF) of sojourn time t_{uf}^s, denoted by $f_{\tau_{uf}}(t)$, is given as

$$f_{\tau_{uf}}(t) = \frac{1}{\tau_{uf}} e^{-\frac{t}{\tau_{uf}}}, t \geq 0, \quad \forall u \in N_u, \ \forall f \in N_F \tag{15}$$

where τ_{uf} denotes the average sojourn time of UE u in the coverage of FCN f. Due to the UEs' different mobility trace and characteristics, τ_{uf} is varied among the UEs in different FCNs. In particular, the sojourn time t_{uf}^s of each UE in different FCNs are assumed to be independently and identically distributed (i.i.d). For simplicity, assume that τ_{uf} follows the Gaussian i.i.d here [31]. In reality, the reliable τ_{uf} can be acquired by gathering the information of UEs with machine learning tools.

Considering the mobility of UEs, when the tasks are offloaded to the FCNs, according to the relationship between the processing delay and the predicted sojourn time of UEs, the total cost of performing the tasks on FCNs are different in two cases, which will be discussed as follows.

Case 1: When the task Z_u is offloaded to FCN f, the sojourn time of UE u in the service coverage of FCN f is longer than the processing delay, i.e., $t_{uf}^s > T_{uf}^F$, which means that the execution result can be transmitted back directly to the UE. In this case, migration of the result won't occur. According to (15), now that the sojourn time t_{uf}^s follows the exponential

distribution, the probability of case 1 can be denoted as $P_{\tau_{uf}}(t^s_{uf} > T^F_{uf})$. And the total cost of UE u on FCN f is

$$q^{F,1}_{uf} = q^F_{uf}(c^F_{uf}), \quad \forall u \in N_u, \ \forall f \in N_F \tag{16}$$

Case 2: Due to the mobility of UEs, if the sojourn time of UEs is shorter than the transmission time, the process of offloading can't succeed. Therefore, UEs whose sojourn time is shorter than the transmission time will be sorted out to execute locally, which is analyzed in next section. As for the other UEs, if the task Z_u offloaded to FCN f and the sojourn time of UE u is shorter than the processing delay on FCN f, the probability of which is $P_{\tau_{uf}}(t^s_{uf} \le T^F_{uf})$, the execution result will firstly be migrated to the FCN which can communicate with UE u through the CS, and then transmitted back to UE. The process of migration will cause the extra cost of energy consumption and time delay, which should be paid by UEs. For the sake of simplicity, the cost of migration is assumed to be only relative to the size of the task, which can be expressed as $q^{mig}_u = \delta D_u$ [30]. In addition to the task size, the cost of migration is also associated with the type of the task the distance between the FCNs and so on in practice [36]. Therefore, the total cost of UE u in case 2 can be expressed as

$$q^{F,2}_{uf} = q^F_{uf}(c^F_{uf}) + q^{mig}_u, \quad \forall u \in N_u, \ \forall f \in N_F \tag{17}$$

From (16) and (17), the cost of performing the task Z_u in two cases is expressed as

$$q^F_{uf} = \begin{cases} q^{F,1}_{uf}, & P_{\tau_{uf}}(t^s_{uf} > T^F_{uf}) \\ q^{F,2}_{uf}, & P_{\tau_{uf}}(t^s_{uf} \le T^F_{uf}) \end{cases}, \quad \forall u \in N_u, \ \forall f \in N_F \tag{18}$$

Therefore, the expectation of the cost q^F_{uf} is denoted as the total cost of performing the task Z_u, which is derived as

$$\bar{q}^F_{uf} = P_{\tau_{uf}}(t^s_{uf} > T^F_{uf}) q^{F,1}_{uf} + P_{\tau_{uf}}(t^s_{uf} \le T^F_{uf}) q^{F,2}_{uf} \tag{19}$$

$$\begin{aligned} P_{\tau_{uf}}(t^s_{uf} \le T^F_{uf}) &= \int_0^{T^F_{uf}} \frac{1}{\tau_{uf}} e^{-\frac{t^s_{uf}}{\tau_{uf}}} dt^s_{uf} \\ &= -e^{-\frac{t^s_{uf}}{\tau_{uf}}} \Big|_0^{T^F_{uf}} = -e^{-\frac{T^F_{uf}}{\tau_{uf}}} + 1 \end{aligned} \tag{20}$$

$$P_{\tau_{uf}}(t^s_{uf} > T^F_{uf}) = 1 - P_{\tau_{uf}}(t^s_{uf} \le T^F_{uf}) \tag{21}$$

IV. PROBLEM FORMULATION AND SOLUTION

A. PROBLEM FORMULATION

As aforementioned, the revenue of UE u is related to the energy consumption and computation delay. The less energy consumption and computation delay are, the higher revenue is. Therefore, define the revenue as the amount of total cost subtraction acquired by the UE if it offloads its task to the FCN. Let Q_u be the cost reduction when the task Z_u is executed on FCN. Therefore, given the offloading decision U_f, the expression of Q_u is given as follows

$$Q_u(s_u, a_{uf}, c^F_{uf}) = \begin{cases} q^{local}_u - \sum_{f \in N_F} a_{uf} \bar{q}^F_{uf}, & s_u = 1 \\ 0, & s_u = 0 \end{cases} \tag{22}$$

To this end, based on the above analysis on the revenue of UEs, an optimization problem of maximizing UEs' revenue is formulated under the resource and delay constraints as follows

$$\begin{aligned} \text{P1}: \max_{S,A,C_{off}} & \sum_{u \in N_u} Q_u(s_u, a_{uf}, c^F_{uf}) \\ s.t. \ & \text{C1}: t^{up}_{uf} < T^F_{uf} \le T^{\max}_u, \quad \forall u \in N_u \\ & \text{C2}: \sum\nolimits_{u \in U_f} c^F_{uf} \le c^F, \forall f \in N_F \\ & \text{C3}: a_{uf} = \{0, 1\}, \quad \forall u \in N_u, \ \forall f \in N_F \\ & \text{C4}: a_{uf} = I(c^F_{uf}), \quad \forall u \in N_u, \ \forall f \in N_F \\ & \text{C5}: \sum\nolimits_{u \in N_u} a_{uf} \le K, \quad \forall f \in N_F \\ & \text{C6}: \sum\nolimits_{f \in N_F} a_{uf} \le 1, \quad \forall u \in N_u \\ & \text{C7}: s_u = \{0, 1\}, \quad \forall u \in N_u \\ & \text{C8}: s_u = \sum\nolimits_{f \in N_F} a_{uf}, \quad \forall u \in N_u \end{aligned} \tag{23}$$

In (23), constraint C1 guarantees that the processing time on FCN won't exceed the maximum latency. C2 constrains that the amount of computation resource distributed to UEs which are accessed to FCN f won't exceed the total computation resource of FCN f. C3 and C7 limit the range of the variable a_{uf} and s_u. In C4, $I(\cdot)$ is the indicator function. If "$\cdot$" > 0, let $I(\cdot) = 1$; otherwise, set $I(\cdot) = 0$. C4 guarantees that the FCN won't distribute the computation resource to the UEs which decide to perform locally. C5 indicates that the number of UEs that each FCN can serve is at most K at the same time, which is depend on the number of sub-channels. C6 constraints that one UE can only access to one FCN to offload its task. And C8 shows that if UE decides to offload its task, it must be allowed to access to one of the FCNs.

Equation (23) shows that revenue of UEs is concerned with the offloading decisions, FCN selection and computation resource allocation. Furthermore, UEs' mobility has a strong effect on the revenue of UEs. If the execution results of the offloaded tasks cannot be transmitted back to the UEs directly, the FCN needs to migrate the results to another FCN through the CS. The process of migration may cause the cost of energy consumption and time delay, and the migration failure may occur due to late migration or early migration or migration to FCN with much higher latency. All of these are related to the value of q^{mig}_u. So in some cases q^{mig}_u can be very large, which will reduce the revenue substantially. Therefore, in order to maximum the revenue of UEs, the impact of the migration cost should be reduced, which can be realized by improving the probability of case 1 occurring, i.e., $P_{\tau_{uf}}(t^s_{uf} > T^{FCN}_{uf})$.

By making the offloading decisions properly, selecting the suitable FCNs and allocating the specific resource according to UEs' different sojourn time, the probability of migration occurring will be reduced significantly. Then the cost of migration will be saved and the revenue will be improved. Therefore, in the next part, a Gini Coefficient based FCN selection algorithm (GCFSA) is proposed to jointly optimize

the offloading decisions and FCN selections, and a distributed resource optimization algorithm based on Genetic Algorithm (ROAGA) is implemented to solve the optimization problem of computation resource allocation. As a result, the probability of migration can be reduced and meanwhile the revenue of UEs can be improved.

B. GINI COEFFICIENT BASED FCN SELECTION ALGORITHM

Acceding to [37], the concept of Gini Coefficient is applied to solve the offloading decision and FCN selection problem. Inspired by the unique function of gini coefficient, we proposed a FCN selection function based on the Gini Coefficient. In this paper, the income of each UE is defined by the selection function and different UEs have different contribution to the total income. We use the selection function to obtain the set $U_f = \{u \in N_u | a_{uf} = 1\}$, where the selected UEs can contribute to the majority of revenue, including the energy, time and migration. In general, the process of GCFSA can be divided into three steps, which are described as follows

- ***Step 1: Pre-offloading.*** According to the relationship between the sojourn time and the transmission time, if the average sojourn time of UEs is shorter than the transmission time, the tasks should be executed locally.
- ***Step 2: Gini Coefficient calculation.*** Based on the results of ***Step 1***,in this step, a selection function is derived to calculate the income of each UE in each FCN. Sort the UEs in each FCN according to the value of income. The Gini Coefficient of each UE is calculated to help the FCNs to decide the maximum number of UEs which can be offloaded.
- ***Step 3: Matching Selection.*** Considering that each UE can be accessed to multi FCNs, sort out the best FCN which maximizes the revenue of the system for each UE according to the income of UEs when connecting to different FCNs, under the constraints C5 and C6 of (23), and the maximum number of UEs which can be offloaded to each FCN obtained from ***Step 2***. And finally get the set $U_f = \{u \in N_u | a_{uf} = 1\}$ for each FCN.

1) PRE-OFFLOADING

If the task is decided to be offloaded to FCN, the Basic Offloading Requirement is defined as follows

Definition 1: (Basic Offloading Requirement, BOR) If the task Z_u is offloaded to the FCN to execute, the requirements should satisfy the following two constraints, $t_{uf}^{up} < \tau_{uf}$ and $q_{uf}^{FCN} < q_u^{local}$.

The UEs which satisfy the BOR, can be offloaded to FCN to execute. If $t_{uf}^{up} > \tau_{uf}$, the task will be terminated during the process of transmission with a high probability because the sojourn time in the coverage of FCN f is so short. If $q_{uf}^{FCN} < q_u^{local}$, it means that even the entire computation resource are allocated to the offloaded task, the cost cannot be saved. So this kind of task should be executed locally and let the FCN serve the more valuable UEs. Therefore, parts of the UEs are sorted out to execute locally in advance, which reduces the problem scale of (23).

Algorithm 1 The Gini Coefficient Based FCN Selection Algorithm

Input: U_f^c, N_u, N_F, Z_u.
Output: A.

```
for f = 1 : |N_F| do
  Step 1 Pre-offloading:
  Initialize: B_f = ∅
  for u = 1 : |U_f^c| do
    if t_uf^up < τ_uf & q_uf^FCN < q_u^local then
      B_f = B_f ∪ {u}
    end if
  end for
  Step 2 Gini Coefficient calculation:
  Calculated the income ψ_uf, u ∈ U_f^c
  B_f --sorted in ascending order--> B_f^s
  Calculate the Gini Coefficient G_f and the number of
    selection I_f by equation (26)–(28)
  Get the offloading space B_f^{s,o} from (29)
end for
Step 3 Matching Selection:
Initialize: a_uf
while Σ_{f∈N_F} a_uf > 1, ∃u ∈ N_u do
  for u = 1 : |N_u| do
    if Σ_{f∈N_F} a_uf > 1 then
      f_max = arg max_{f∈{f|a_uf=1}} {ψ_uf}
      for f ∈ {f|a_uf = 1}\f_max do
        a_uf = 0
        Add the first unselected UE from the set
          {B_{f,i}^s | i = |B_f^s| − I_f, ..., 1}
      end for
    end if
  end for
end while
```

2) GINI COEFFICIENT CALCULATION

First, define the selection function to calculate the income of UEs as follows

Definition 2: (Selection Function)

$$\psi_{uf} = \eta_{uf}\left[q_u^{local} - \bar{q}_{uf}^F(c^F)\right]^+, \quad \forall u \in N_u, \ \forall f \in N_F \tag{24}$$

where $[x]^+ = \max\{x, 0\}$, and η_{uf} is the weight factor of UE u's revenue, which is calculated as

$$\eta_{uf} = \left(\frac{\sum_{u \in B_f} \varepsilon_{uf}^F}{|B_f|\,\varepsilon_{uf}^F}\right)\left(\frac{\tau_{uf}}{t_{uf}^{up} + t_{uf}^{exe}}\right), \quad \forall u \in N_u, \ f \in N_F \tag{25}$$

where B_f is the set of candidate UEs of FCN f, which can be obtained from step 1, and $\varepsilon_{uf}^F = f_u \Big/ \left(\tau_{uf} - t_{uf}^{up}\right)$. $\left[q_u^{local} - \bar{q}_{uf}^F(c^F)\right]^+$ represents the revenue of UE u obtained

from FCN f when the entire computation resource is distributed to the UE. The weight factor is proposed in order to take the required computation resource and sojourn time into consideration. As for the weight factor, the ratio $\frac{\sum_{u\in B_f}\varepsilon_{uf}^F}{|B_f|\varepsilon_{uf}^F}$ is used to denote the efficiency of computation resource in revenue of UE u. Mobility is an significant factor in affecting the revenue. Considering the sojourn time related to the computational resource, the time ratio $\frac{\tau_{uf}}{t_{uf}^{up}+t_{uf}^{exe}}$ is introduced as a multiplier in the weight factor. The higher η_{uf} you get, the higher revenue of UEs can be obtained when being offloaded. Therefore, the weight factor of revenue is designed as (25).

Calculated the income of UEs in B_f for each FCN according to the Selection Function. All the UEs in B_f are sorted in ascending order of the value of income ψ_f^s: $\psi_f^{s1} \leq \psi_f^{s2} \leq \ldots \leq \psi_f^{s|B_f|}$. The sorted B_f is defined as B_f^s. Define the sum income belonging to FCN f as $Y_f = \sum_{u=1}^{|B_f|}\psi_{uf}, \forall f \in N_F$, and define the cumulative income ratio as $y_{if} = \frac{1}{Y_f}\sum_{u=1}^{i}\psi_{uf}, i = 1, 2, \ldots, |B_f|$. So far, the Gini Coefficient of FCN f, i.e., G_f, can be obtained as

$$G_f = 1 - \frac{1}{|B_f|}\left(1 + 2\sum_{i=1}^{|B_f|-1} y_{if}\right), \quad \forall f \in N_F \tag{26}$$

and define the number of selection of FCN f as

$$I_f = \min\left\{\left\lceil\frac{1}{G_f}\right\rceil + \left\lceil\frac{L_f}{|B_f|}(|B_f| - \left\lceil\frac{1}{G_f}\right\rceil)\right\rceil, |B_f|\right\} \tag{27}$$

where

$$L_f = \min\left\{\left\lfloor\frac{c^F}{\varepsilon_{f,mid}^F}\right\rfloor, |B_f^s|, K\right\}$$
$$\varepsilon_{f,mid}^F = \left\{\varepsilon_{uf}^{s,F} | u = \left\lfloor\frac{|B_f|}{2}\right\rfloor\right\} \tag{28}$$

where $\varepsilon_{uf}^{s,F}$ belongs to the set where ε_{uf}^F is sorted in ascending order.

The value of $\left\lceil\frac{1}{G_f}\right\rceil$ denotes the number of UEs contributing to the majority of total income []. $\left\lceil\frac{L_f}{|B_f|}(|B_f| - \left\lceil\frac{1}{G_f}\right\rceil)\right\rceil$ is the correction factor of $\left\lceil\frac{1}{G_f}\right\rceil$. L_f is the load capacity of resources of FCN f. And $\frac{L_f}{|B_f|}$ is the weight factor of the difference between $|B_f|$ and $\left\lceil\frac{1}{G_f}\right\rceil$. Therefore, the problem scale of (23) can be further reduced and the offloading space of UEs for FCN f, denoted as $B_f^{s,o}$, can be obtained from the sorted UEs B_f^s as

$$B_f^{s,o} = \{B_{f,|B_f^s|}^s, B_{f,|B_f^s|-1}^s, \ldots, B_{f,|B_f^s|-I_f+1}^s\}, \quad \forall f \in N_F \tag{29}$$

3) MATCHING SELECTION

The constraints C5 and C6 of (23) limit the number of FCNs that each UE can connect. But the UEs selected in Step 2 can possibly connect with more than one FCN at the same time. Therefore, the set $B_f^{s,o}$ should be further scaled to satisfy the constraints C5 and C6 of (23).

If UE u is selected by more than one FCN, compare the income in different FCNs, and choose the FCN which has the largest income. Then eliminate the UE u in the other sets of $B_f^{s,o}$ and add the new UE from the set $\left\{B_{f,i}^s | i = \left|B_f^s\right| - I_f, \ldots, 1\right\}$. Do the loop operation until all the UEs satisfy the constraint C6 in (23). Finally the set $U_f = \left\{u \in N_u | a_{uf} = 1\right\}$ for each FCN can be obtained. And the UEs which are not selected by any FCNs will be operated locally.

The complexity of GCFSA algorithm is polynomial complexity, which is analyzed in the following. Firstly, the step 1, will do the iteration for $|N_F||U_f^c|$ times to classify the inappropriate UEs to execute locally. Secondly, as for the step 2, the complexity of calculation is $O(|B_f|)$ and the complexity of sorting process is $O(|B_f^2||N_F|)$. Finally, the step of step 3, has $|N_F||U_f^c|$ iterations to get the set U_f for each FCN. Therefore, the computational complexity of GCFSA can be given as $O(|N_F||U_f^c| + |B_f| + |B_f|^2|N_F| + |N_F||U_f^c|)$. Due to $|B_f| \leq |U_f^c| \leq K$,the complexity can be further expressed as $O(K^2F)$.

C. RESOURCE OPTIMIZATION ALGORITHM BASED ON GENETIC ALGORITHM

Once the offloading decision and FCN selection are decided. A resource optimization algorithm based on genetic algorithm (ROAGA) is applied to allocate the computation resource to the offloaded tasks because of its better global search property.

Since the FCNs are independent with each other, the resource allocation in each FCN can be performed independently. Specially, the revenue is regarded as the fitness function to evaluate the goodness of the individuals. The constraints of the problem will be ensured within the initialization and selection. Since the optimization problem requires high precision, the real coded strings are chosen as the chromosomes. Each one of the chromosome is a solution of problem (23), which is represented by

$$J_i^f = [J_1, \ldots J_u \ldots, J_{|U_f|}], \quad f \in N_F \tag{30}$$

where $J_i = C_{uf}^F$ is the set of the variables of UE u. Algorithm 2 shows the detailed process.

V. SIMULATION

In this section, the revenue performance of different algorithms with different parameters are numerically showed in simulations. The simulation parameter settings are listed in TABLE 2 [14], [38]. The coverage radius of the whole fog computing networks is 500 m and the coverage of each FCN is 100 m. FCNs and UEs are randomly distributed. As for mobility of UEs, the distribution of average sojourn time of UEs follows the Gaussian distribution $CN(\mu_t, \sigma_t^2)$, where $\mu_t = 30\,seconds$ and $\sigma_t = 10$. The parameters of GA are set

Algorithm 2 The Resource Optimization Algorithm Based on Genetic Algorithm

Input: $|U_f|, K, P_c, P_m, T$.

Output: J^f_{best}, Q_{best}.

1: **Initialize:** Set K individuals for the population in a random way under the constraints of (14). Calculate the fitness value of each individual and sort out the biggest one as Q_{best}. Set the best individual as J^f_{best}.
2: **for** $t = 1$ to T **do**
3: Randomly choose two individuals and do the crossover operation with the probability P_c. The crossover operation applies the recombination method for the set C_f;
4: Select the individuals from the parents and offsprings for mutation operation with the probability P_m;
5: Calculate the fitness value of each new individual and divide them into the feasible ones and infeasible ones;
6: Do the random tournament selection operation and make sure the best individual is sorted out. Compare the best individual of iteration t, denoted by $J^{f,t}_{best}$, with the historical best individual J^f_{best}. If $J^{f,t}_{best}$ is better than J^f_{best}, then let $J^f_{best} = J^{f,t}_{best}$ and renew Q_{best};
7: **end for**

TABLE 2. Simulation parameters.

Parameter	Value	Parameter	Value
K	10	P_u	$[0.08, 0.1]W$
D_u	$[10, 15]MB$	ε	$[0.2, 0.3]Gcycles/MB$
T_u^{max}	$50s$	λ_u^E	0.1, 0.5(default), 0.9
λ_u^T	0.1, 0.5(default), 0.9	w	$10MHz$
c_u^{local}	$[0.5, 0.8]GHz$	c^F	$4MHz$
κ	10^{-11}	σ^2	$-100dBm$

as follows, $K = 32$, $P_c = 0.6$ and $P_m = 0.1$. The performance of the proposed Algorithm (PA) is compared with the other four algorithms. The first one is the near-optimal resource allocation mechanism (NORAM) from [14]. The computation resource are distributed uniformly and the mechanism didn't consider the mobility. Allocation algorithm regardless of the mobility (AARM) is similar with GAAA, except that it doesn't consider the mobility. Randomly offloading algorithm (ROA) and all offloading algorithm (AOA) both uniformly allocate the computation resource to UEs. But ROA randomly offloads the tasks with the probability 0.5 and AOA offloads all the tasks.

The revenue of UEs in different algorithms varying with the number of FCNs is shown in Fig. 2. The UEs are randomly distributed around the FCNs. FCNs selectively offload the tasks of appropriate UEs to maximum their total revenue. Due to the limited computation resource in each FCN, only a limited number of UEs can be served. So with more FCNs being deployed around the UEs, more UEs can be served and more tasks can be offloaded to improve the revenue of UEs. Considering the mobility can help the FCNs choose the appropriate UEs to offload by taking the channel condition, required computation resource and the average sojourn time

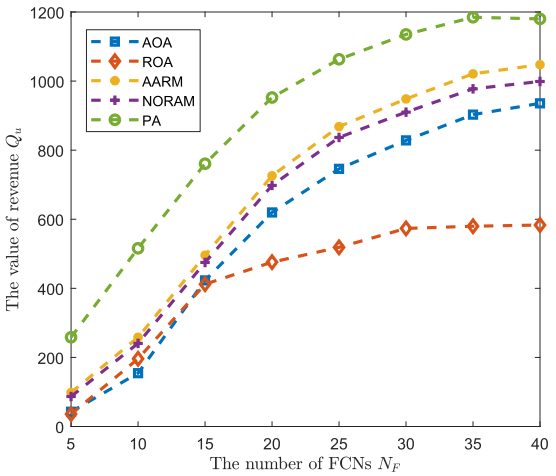

FIGURE 2. The impact of the number of FCNs F from 5 to 40, where $N = 70$, $\delta = 0.05$ and $c^F = 4$ GHz.

into account. Since the sojourn time follows the exponential distribution, knowing the average sojourn time of each UE helps FCNs to calculate the probability of migration if a certain number of computation resource is allocated. Therefore, by optimizing the offloading and resource allocation strategies, the probability of migration can be reduced as much as possible, which eventually improves the revenue of UEs. It can be observed that the performances in algorithms are essentially unchanged when the number of FCNs becomes larger. Because the number of served UEs is fixed. When the number of FCNs is large enough to serve the total UEs, the increasing number of FCNs won't make much difference any longer. Note that some UEs cannot be offloaded even when the number of FCNs becomes much large because of their worse channel condition and shorter sojourn time. That is the reason why AOA performs worst and PA performs the best. Besides, due to the uniform distribution of the computation resource, the revenue in NORAM is a little bit lower than AARM.

Fig. 3(a) shows the radio of not migrated tasks, which is the number of not migrated tasks to the number of total offloaded tasks, with different algorithms versus the number of UEs. Fig. 3(b) illustrates the total revenue of UEs with different number of UEs. When number of UEs is small, the proposed algorithm can guarantee that around 94% of offloaded tasks can be finished before UEs leaving the coverage of FCNs. And with the increasing of UEs, the radio of not migrated tasks in AOA, AARM and NORAM fall faster than PA and ROA. Because AARM and NORAM don't consider the mobility of UEs, which could cause the migration. To maximum the revenue of UEs, so many tasks are offloaded in AARM and NORAM that about half of them cannot be finished before leaving. Besides, the inappropriate selection of FCNs could also lower the radio of not migrated tasks. Note that the performance in NORAM is worse than AARM, that is because NORAM distributes the resource uniformly to the UEs. The reason of AOA is similar with AARM. But since it offloads all the tasks, the radio is smaller than

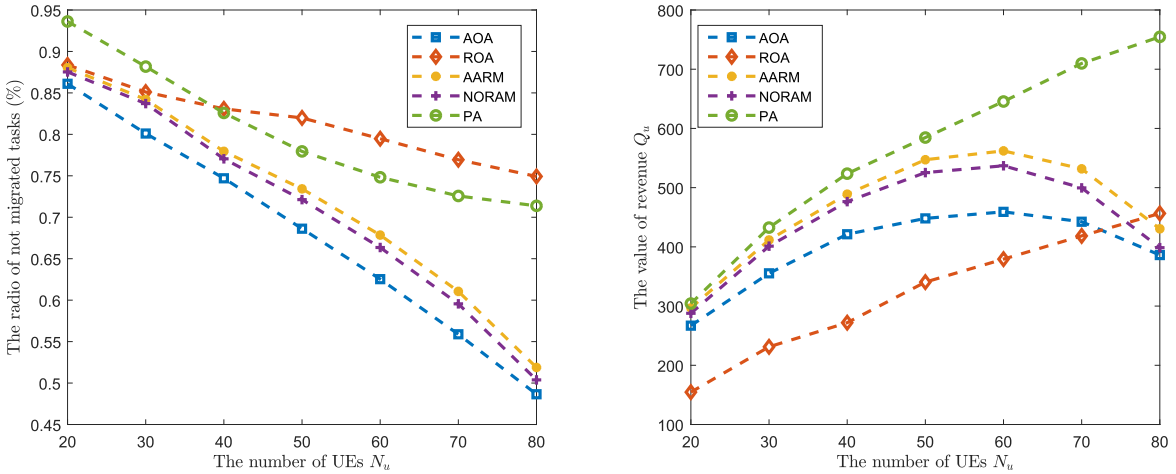

FIGURE 3. The impact of number of UEs, where total computing capacity of each FCN $c^F = 4$ *GHz*, migration cost $\delta = 0.05$ and number of FCNs $F = 20$. (a) The impact to the radio of not migrated tasks. (b) The impact to the total revenue of UEs.

the radio of AARM and NORAM. The radio in PA also decreases but at a low speed. Because the priority target of PA is to maximum the total revenue of UEs. Although offloading more tasks would cause more tasks migrated, the total revenue can be improved, as illustrated in Fig. 3(b). It is believed that if the cost of migration is much high, the less tasks would be offloaded in PA. Note that the radio in ROA decreasing slowest. Because random offloading tasks makes the number of offloaded tasks increasing slowest. Therefore, although the radio of not migrated tasks is high, even higher than PA, the revenue in ROA is much small, which is shown in Fig. 3(b).

In Fig. 3(b), note that with the increment of the number of UEs, the performances of AOA, AARM and NORAM are different from PA and ROA. The revenue of all the algorithms increase with the number of UEs at first. However, the increasing speeds of AOA, AARM and NORAM are getting slow, and when the number comes to 60, the revenue begins to decrease. Because the finite number of FCNs and their limited computing capacity can only serve a limited number of UEs. At first, the number of UEs hasn't reached the upper limit, so the total revenue still increase but the speed is getting slower. Later, together with the number of offloaded tasks increasing, the migrated tasks also increase. The increment of migration cost will be bigger than the increment of gains from offloading. Then the total revenue begin to fall. But since AOA offloads all the tasks, its revenue is worse than AARM. In addition, the revenue in NORAM is getting worse with the number of UEs increasing, because the uniform distribution would let more tasks be migrated. PA considers the mobility and the probability of migration, so when the number of UEs comes to the bottleneck of FCNs, the number of offloaded tasks stops increasing and the revenue begins to keep stable. The reason of ROA is similar with AOA, but due to the less number of offloaded tasks, it hasn't reached the upper limit yet.

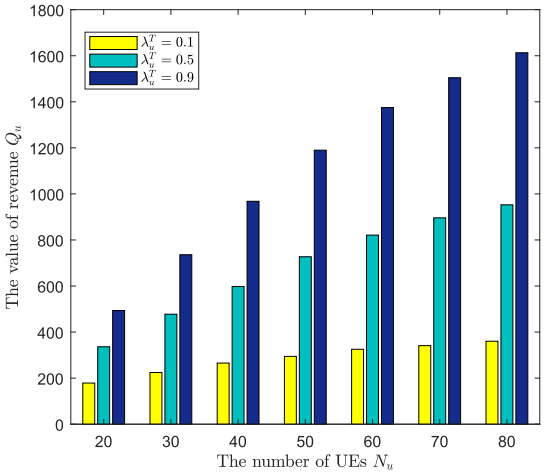

FIGURE 4. The impact of the number of UEs and different weights $\lambda_u^T = 0.1, 0.5, 0.9$, $\delta = 0.05$, $F = 20$, $c^F = 4$ *GHz*.

Fig. 4 compares the different weight coefficients of time delay $\lambda_u^T = 0.1, 0.5, 0.9$ with the number of UEs increasing from 20 to 80, where $\lambda_u^T = 0.1$ represents the case of low battery power in UEs and $\lambda_u^T = 0.9$ represents the delay sensitive applications. All of them increase with the growth of number of UEs. But the values of revenue are quite different. Therefore, to make the different simulation results among the algorithms more apparent, $\lambda_u^T = 0.5$ and $\lambda_u^E = 0.5$ are chose as simulation parameters to balance the cost of energy consumption and time delay.

The total revenue of UEs with different FCN computing capacity is simulated in Fig. 5. It can be seen that the revenue of all the algorithms get higher as the computing capacity increasing in each FCN. When the computing capacity is small at the beginning, the value of revenue in PA is still better than the others, which illustrates that the proposed algorithm can make good use of computation resource of FCNs. However, with the increasing of computing capacity,

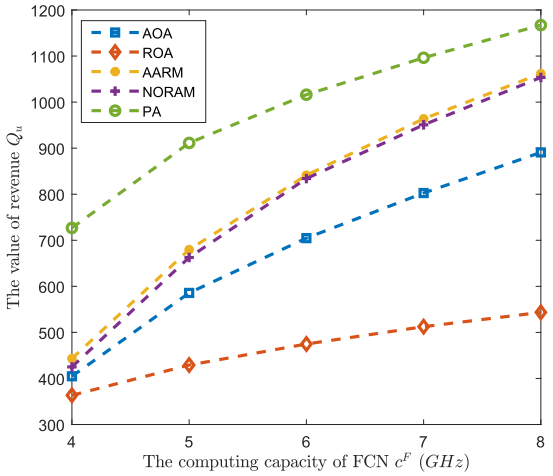

FIGURE 5. The impact of the computing capacity of each FCN c^F from 8 *GHz* to 16 *GHz*, where $N = 70$, $F = 20$, and $\delta = 0.05$.

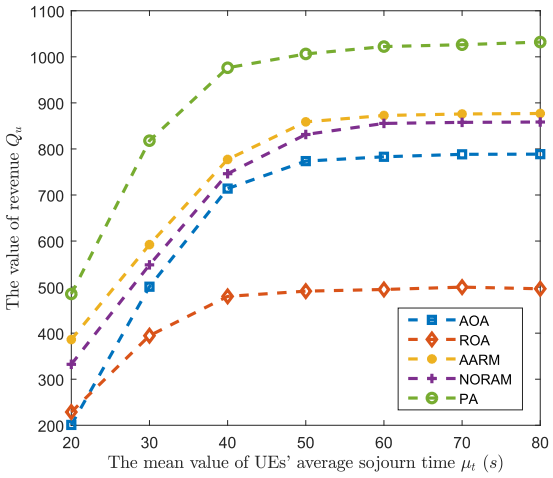

FIGURE 6. The revenue with different mean value of the distribution of UEs' average sojourn time μ_t from 20*s* to 80*s*, where $N = 60$, $F = 20$, $\delta = 0.08$, and $c^F = 4$ *GHz*.

the difference among PA and the other algorithms are getting smaller. Because PA offloads the tasks selectively in order to reduce the probability of migration. With the growth of computing capacity, the offloaded tasks will be guaranteed to be finished before UEs leaving the coverage of current FCN with high probability first. So the number of offloaded tasks won't change quickly, and the total revenue grow slowly. However, in AARM and NORAM, larger computing capacity let less tasks be migrated. Since the number of offloaded tasks and migrated tasks are large, the revenue increase faster than PA. Besides, since NORAM uniformly distributes the resource, when the computing capacity become larger, the difference between AARM and NORAM gets smaller. The number of offloaded tasks and migrated tasks are also larger in AOA, but due to the strategy of all tasks being offloaded, some UEs in worse conditions are also offloaded, which affects the total revenue. Furthermore, uniformly allocating the resource won't efficiently leverage the computing capacity. The number of offloaded tasks in ROA is small and essentially unchanged. Next, the average sojourn time of UEs are studied in Fig. 6, which indirectly demonstrates that the algorithms ignoring the mobility would make too many tasks offloaded and increase the number of migrated tasks.

In the settings, the distribution of average sojourn time of UEs follows the Gaussian distribution $CN(\mu_t, \sigma_t^2)$, where $\mu_t = 30\,seconds$ and $\sigma_t = 10$. In Fig. 6, we investigate the variation of revenue with different μ_t which characterizes the average sojourn time of all the UEs in this area. It can be observed from the figure that the revenue in PA increases faster than the other four algorithms. Because as the μ_t increasing, the number of UEs which have long simulated average sojourn time is larger than before. So considering the sojourn time of each UE can help the FCNs select more UEs which stay in the coverage for a long time. Besides, there is no need to allocate too much computation resource to reduce the probability of migration, so the resource are saved to serve more UEs. Note that when the mean value μ_t becomes much larger, the revenue of each algorithm stop to increase. Because the sojourn time is long enough and almost all the offloaded tasks can be finished and transmitted back to UEs without migration. As for the difference between the algorithms, although the μ_t is large, the simulated average sojourn time of each UE can still be small, and ignoring the mobility would result in selecting the UEs with short average sojourn time. Therefore, the increasing of the average sojourn time of all the UEs μ_t has small impact on narrowing the gap among PA and other algorithms.

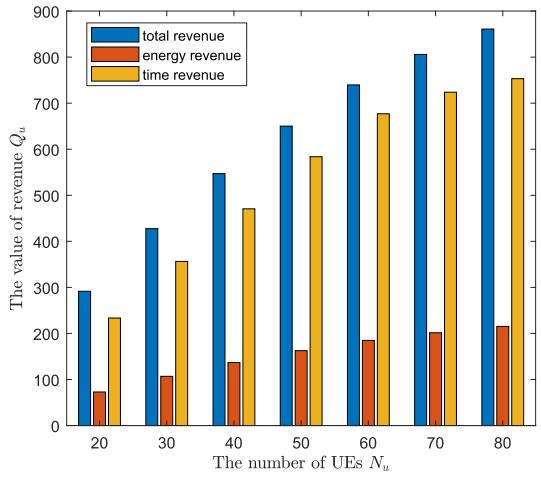

FIGURE 7. The revenue of time and energy with the increasement of UEs, where $\delta = 0.05$, $F = 20$, and $c^F = 4$ *GHz*.

In Fig. 7, both the revenue of time and energy increase with the increasement of UEs. It can be noted that the revenue of energy is smaller than the time revenue and the increasing speed of energy revenue is slower than the time revenue. Because reducing the probability of migration can effectively lower the time of completion meanwhile. However, the energy saving won't change too much because it's only related to the local energy consumption and

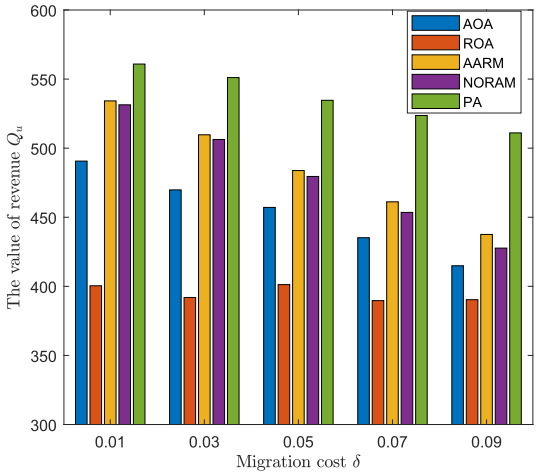

FIGURE 8. The impact of the migration cost δ from 0.01 to 0.09, where $N = 40$, $F = 10$, and $c^F = 4$ *GHz*.

transmission power. Besides, in order to reduce the probability of migration, less tasks will be offloaded and the local energy consumption can't be saved.

In Fig. 8, the total revenue of all the algorithms decrease with the growth of migration cost. The revenue in PA declines slowly, which proves that the proposed scheme in this work can efficiently lower the probability of migration. The number of migrated tasks becomes so small that the growth of migration cost has little effect on revenue. In AARM,NORAM and AOA, omitting the mobility of UEs would have more tasks offloaded. Since the computation resource is limited, the probability of migration becomes high. Therefore, the total revenue of UEs could be influenced by the increasing of migration cost strongly. Specially, it can be observed that the revenue of AOA decreases slower than AARM and NORAM. Because AOA offloads all the tasks, although it is not a good strategy, it can balance effect of the UEs with short sojourn time and long sojourn time. So the influence of different migration cost has less impact on AOA than AARM and NORAM. Furthermore, because of the random offloading in ROA, much less tasks are offloaded. So the computation resource is relatively enough and the probability of migration is small. The changing of migration cost has less influence on ROA and the curve is smooth.

VI. CONCLUSION

In this paper, the mobility of UEs is represented by the sojourn time, which follows the exponential distribution. To reduce the probability of migration so as to maximize the total revenue of UEs, a mobility aware offloading and computation allocation scheme is proposed in fog computing networks. The proposed algorithms considering mobility can effectively deal with the scenario of UEs' mobility in fog computing networks to maximum the total revenue of UEs. Simulations demonstrate that the proposed algorithms can achieve quasi-optimal revenue performance compared with other baseline algorithms, in which our proposed scheme can significantly reduce the migration times and improve the revenue of UEs. How to reduce the cost of migration for the migrated tasks will be considered in the future work.

REFERENCES

[1] H. F. Atlam, A. Alenezi, A. Alharthi, R. J. Walters, and G. B. Wills, "Integration of cloud computing with Internet of Things: Challenges and open issues," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, Jun. 2017, pp. 670–675.

[2] M. Peng, S. Yan, K. Zhang, and C. Wang, "Fog-computing-based radio access networks: Issues and challenges," *IEEE Netw.*, vol. 30, no. 4, pp. 46–53, Jul./Aug. 2016.

[3] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 416–464, 1st Quart., 2018.

[4] L. Liu, Z. Chang, and X. Guo, "Socially aware dynamic computation offloading scheme for fog computing system with energy harvesting devices," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1869–1879, Jun. 2018.

[5] Y. Jiang and D. H. K. Tsang, "Delay-aware task offloading in shared fog networks," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4945–4956, Dec. 2018.

[6] G. Zhang, F. Shen, Y. Yang, H. Qian, and W. Yao, "Fair task offloading among fog nodes in fog computing networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–6.

[7] A. Bozorgchenani, D. Tarchi, and G. E. Corazza, "Centralized and distributed architectures for energy and delay efficient fog network-based edge computing services," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 1, pp. 250–263, Mar. 2019.

[8] A. Bozorgchenani, D. Tarchi, and G. E. Corazza, "An energy and delay-efficient partial offloading technique for fog computing architectures," in *Proc. IEEE Global Commun. Conf.*, Singapore, Dec. 2017, pp. 1–6.

[9] Y. Gu, Z. Chang, M. Pan, L. Song, and Z. Han, "Joint radio and computational resource allocation in IoT fog computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7475–7484, Aug. 2018.

[10] T. T. Nguyen and B. L. Long, "Joint computation offloading and resource allocation in cloud based wireless HetNets," in *Proc. IEEE Global Commun. Conf.*, Singapore, Dec. 2017, pp. 1–6.

[11] 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.

[12] 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.

[13] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, "On reducing IoT service delay via fog offloading," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 998–1010, Apr. 2018.

[14] 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.

[15] Y. Yang, K. Wang, G. Zhang, X. Chen, X. Luo, and M. Zhou, "MEETS: Maximal energy efficient task scheduling in homogeneous fog networks," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 4076–4087, Oct. 2018.

[16] Z. Xiong, S. Feng, W. Wang, D. Niyato, P. Wang, and Z. Han, "Cloud/fog computing resource management and pricing for blockchain networks," *IEEE Internet Things J.*, to be published.

[17] L. Liu, Z. Chang, X. 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.

[18] F. Chiti, R. Fantacci, and B. Picano, "A matching theory framework for tasks offloading in fog computing for IoT systems," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5089–5096, Dec. 2018.

[19] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F. R. Yu, and Z. Han, "Computing resource allocation in three-tier IoT fog networks: A joint optimization approach combining Stackelberg game and matching," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1204–1215, Oct. 2017.

[20] G. Li, J. Wu, J. Li, K. Wang, and T. Ye, "Service popularity-based smart resources partitioning for fog computing-enabled industrial Internet of things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4702–4711, Oct. 2018.

[21] A. Rahman, J. Jin, A. Cricenti, A. Rahman, and M. Panda, "Motion and connectivity aware offloading in cloud robotics via genetic algorithm," in *Proc. IEEE Global Commun. Conf.*, Singapore, Dec. 2017, pp. 1–6.
[22] X. Chen and J. Zhang, "When D2D meets cloud: Hybrid mobile task offloadings in fog computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017, pp. 1–6.
[23] C. Wang, Y. Li, and D. Jin, "Mobility-assisted opportunistic computation offloading," *IEEE Commun. Lett.*, vol. 18, no. 10, pp. 1779–1782, Oct. 2014.
[24] Y. Shi, S. Chen, and X. Xu, "MAGA: A mobility-aware computation offloading decision for distributed mobile cloud computing," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 164–174, Feb. 2018.
[25] Z. Wang, Z. Zhao, G. Min, X. Huang, Q. Ni, and R. Wang, "User mobility aware task assignment for mobile edge computing," *Future Gener. Comput. Syst.*, vol. 85, pp. 1–8, Aug. 2018.
[26] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.
[27] C. Zhu *et al.*, "Folo: Latency and quality optimized task allocation in vehicular fog computing," *IEEE Internet Things J.*, to be published.
[28] R. A. Addad, D. L. C. Dutra, M. Bagaa, T. Taleb, and H. Flinck, "Towards a fast service migration in 5G," in *Proc. IEEE Conf. Standards Commun. Netw. (CSCN)*, Paris, France, Oct. 2018, pp. 1–6.
[29] H. Zhang, Y. Xiao, S. Bu, D. Niyato, R. Yu, and Z. Han, "Fog computing in multi-tier data center networks: A hierarchical game approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6.
[30] W. Nasrin and J. Xie, "SharedMEC: Sharing clouds to support user mobility in mobile edge computing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–6.
[31] X. Liu, J. Zhang, X. Zhang, and W. Wang, "Mobility-aware coded probabilistic caching scheme for MEC-enabled small cell networks," *IEEE Access*, vol. 5, pp. 17824–17833, 2017.
[32] A. Al-Shuwaili and O. Simeone, "Energy-efficient resource allocation for mobile edge computing-based augmented reality applications," *IEEE Wireless Commun. Lett.*, vol. 6, no. 3, pp. 398–401, Jun. 2017.
[33] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.
[34] Y. Wen, W. Zhang, and H. Luo, "Energy-optimal mobile application execution: Taming resource-poor mobile devices with cloud clones," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Orlando, FL, USA, Mar. 2012, pp. 2716–2720.
[35] W. Hao and S. Yang, "Small cell cluster-based resource allocation for wireless backhaul in two-tier heterogeneous networks with massive MIMO," *IEEE Trans. Veh. Technol.*, vol. 67, no. 1, pp. 509–523, Jan. 2018.
[36] A. Machen, S. Wang, K. K. Leung, B. J. Ko, and T. Salonidis, "Live service migration in mobile edge clouds," *IEEE Wireless Commun.*, vol. 25, no. 1, pp. 140–147, Feb. 2018.
[37] P. Zhao, H. Tian, C. Qin, and G. Nie, "Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing," *IEEE Access*, vol. 5, pp. 11255–11268, 2017.
[38] M. Liu, F. R. Yu, Y. Teng, V. C. M. Leung, and M. Song, "Distributed resource allocation in blockchain-based video streaming systems with mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 695–708, Jan. 2019.
[39] E. Dahlman, S. Parkvall, and J. Skold, *4G: LTE/LTE-Advanced for Mobile Broadband*. New York, NY, USA: Academic, 2013.

DONGYU WANG received the B.S. and M.S. degrees from Tianjin Polytechnic University, China, in 2008 and 2011, respectively, and the Ph.D. degree from the Beijing University of Posts and Telecommunications, China, in 2014. From 2014 to 2016, he held a postdoctoral position with the Department of Biomedical Engineering, Chinese PLA General Hospital, Beijing. In 2016, he joined the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications. His research interests include device-to-device communication, multimedia broadcast/multicast service systems, resource allocation, theory and signal processing, with specific interests in cooperative communications, fog computing, and mobile edge computing.

ZHAOLIN LIU received the B.S. degree in communications engineering from the University of Electronic Science and Technology of China (UESTC), in 2017. He is currently pursuing the M.E. degree with the Beijing University of Posts and Telecommunications (BUPT). His research interest includes edge computing, including mobile edge computing and fog computing.

XIAOXIANG WANG received the B.S. degree in physics from Qufu Normal University, Qufu, China, in 1991, the M.S. degree in information engineering from East China Normal University, Shanghai, China, in 1994, and the Ph.D. degree in electronic engineering from the Beijing Institute of Technology, Beijing, China, in 1998. In 1998, she joined the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications. From 2010 to 2011, she was a Visiting Fellow with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh. Her research interests include communications theory and signal processing, with specific interests in cooperative communications, multiple-input-multiple-output systems, multimedia broadcast/multicast service systems, and resource allocation.

YANWEN LAN received the B.S. degree in communications engineering from Henan University, in 2013, and the M.E. degree in electrical and communications engineering from Hainan University, in 2016. He is currently pursuing the M.S. degree with the Beijing University of Posts and Telecommunications (BUPT). His research interests include fog computing, mobile edge computing, and cache-enabled heterogeneous networks.