IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 27, 2020, accepted August 29, 2020, date of publication September 1, 2020, date of current version September 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3020935

Joint Network Selection and Service Placement Based on Particle Swarm Optimization for Multi-Access Edge Computing

SHUYUE MA, SHUDIAN SONG, JINGMEI ZHAO, (Member, IEEE), LINBO ZHAI, AND FENG YANG

School of Information Science and Engineering, Shandong Normal University, Jinan 250014, China

Corresponding authors: Linbo Zhai (zhai@mail.sdu.edu.cn) and Feng Yang (yangfeng@sdnu.edu.cn)

This work was supported by the Key Research and Development Program of Shandong Province, China, under Grant 2017GGX10142.

ABSTRACT With the popularity of mobile devices such as smartphones and tablets, the improvement of service of quality is an important issue facing great challenges. The improvement of user service of quality is mainly reflected in reducing the energy consumption of mobile devices and the delay of task execution. Multi-access edge computing sinks computing and storage capabilities from the remote cloud to the edge network, which can effectively reduce the high latency caused by the transmission of tasks between the mobile device and the remote cloud and the high energy consumption of tasks performed locally. Most of the previous work was limited to service of quality optimization through dynamic service layout, while ignoring the critical impact of access network selection on network congestion. This article studies the task offloading model of multiple tasks and services with several MEC servers, and jointly optimizes the MEC's access network selection and service placement issues. Considering the delay and energy consumption caused by task offloading and execution, this article designs an effect function on delay and energy consumption, and aims to minimize this function to solve the MEC problem. Since this problem is NP-hard, this article designs a new optimization algorithm based on particle swarm optimization to solve this problem. Extensive simulation experiments show that the proposed optimization algorithm realize better performance than other algorithms. The algorithm has achieved good results in terms of time delay and energy consumption, which effectively reduces the system cost.

INDEX TERMS Multi-access edge computing, particle swarm optimization, quality-of-service.

I. INTRODUCTION

In recent years, with the explosive growth of smart devices, many emerging applications, such as AR(augmented reality, AR) [1], face recognition [2], interactive games [3], have attracted more and more attention. Such applications not only require intensive computing resources, but also have higher requirement for time delay. Due to physical size limitations [4], mobile devices are limited in computing power and energy. While traditional cloud computing provides centralized services for applications, the distance between the service hosting cloud and users is far, which inevitably results in large end-to-end delays. Therefore, neither local computing nor the traditional cloud computing paradigm can meet the timeliness requirements of such applications.

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

To meet this challenge, a new computing paradigm, MEC (Multi-access Edge Computing, MEC) is proposed to deploy computing and storage resources from remote cloud to network edge deployment close to users [5]. MEC is widely recognized as a promising technology that not only meets the growing demand for computing from applications, but also meets the growing demand for user QoS (quality of service, QoS). By deploying edge clouds near users to store a large number of computing resources and services [6], MEC enables cloud computing power and IT(information technology, IT) environment close to users, achieving the goal of reducing latency and saving device energy.

The main goal of multi-access edge computing is to provide satisfactory service quality and obtain high economic benefits for operators. Since the services and computing resources of the edge cloud are independent and fine-grained [7], system-wide optimization can be achieved

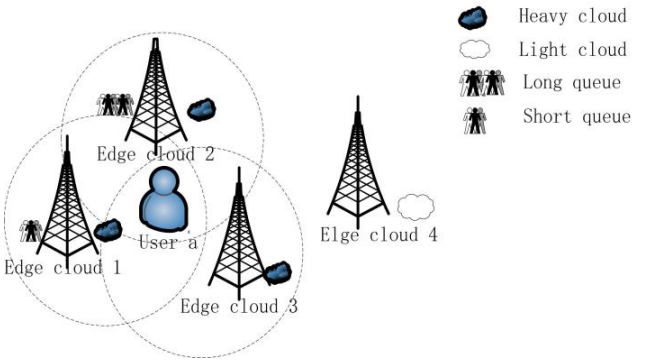


FIGURE 1. The network selection and service placement.

through appropriate offloading of tasks and services [8]. When too many tasks are connected to the same edge cloud at the same time, it will cause network congestion, increasing delay and energy consumption. Therefore, reasonable network selection and service placement are crucial. Most of the previous work is confined to the optimization of QoS through dynamic service layout and the influence of access network selection is ignored.

In this article, a MEC system with multiple edge server services is considered to jointly optimize the access network selection and service placement of MEC. Each task can be served by multiple access points, while each task can only be served by one access point in each time slot t. In order to improve QoS of MEC system, this article comprehensively studies the influence of communication, queuing, switching, computing and transmission delay. In particular, during the offloading process, this article not only selects the nearest edge cloud with users, but also considers the current edge cloud network status, queuing situation, etc. By integrating the above factors, this article chooses the most appropriate access point and service placement point for users. The specific process is shown in Figure 1. Figure 1 consists of user a and four edge clouds, where each edge cloud consists of an access point and a cloud. The optional base station of user a is edge cloud 1, 2, 3, and it is out of the service range of edge cloud 4. From the perspective of physical distance, edge cloud 2 is the best network access point closest to the user, while edge cloud 2 has a large number of tasks waiting to be queued. Compared with edge cloud 2, edge cloud 1 is farther from the user and has fewer tasks waiting to be processed, while edge cloud 3 is farthest from the user, with no tasks to process. Besides, the edge clouds 1, 2, and 3 places a large number of services, causing the cloud's heavy. The edge cloud 4 beyond the coverage area has fewer placement services, while choosing the edge cloud 4 will inevitably cause a lot of communication delays. Therefore, this article has designed an optimization algorithm based on PSO (particle swarm optimization, PSO) to optimize the user's network selection and service placement. The main contributions of this article are as follows:

1) This article studies the multi-task and multi-service task offloading model with multiple MEC servers, and jointly optimizes the MEC access network selection and service placement problems.

2) Most of the current research work focuses on minimizing the total delay of data transmission and task execution under energy consumption or minimizing the energy consumption of mobile terminals under the performance of satisfying the user's perceived delay. Different from this kind of works, this article considers both delay and energy consumption, designs an effect function on delay and energy consumption, and minimizes the effect function to minimize the total cost of the mobile terminal or the whole system consisting of mobile terminal and edge server.

3) Since MEC network selection and service placement are NP-hard, this article designs an algorithm based on PSO to solve this problem. As far as we know, there is no work using PSO-based algorithms to solve this problem.

4) In the optimization process, this article designs a transition probability for selecting access points and service placement base stations.

5) Extensive simulations show that the algorithm proposed in this article is superior to other algorithms.

The rest of this article is organized as follows. The section II introduces the related work. The section III illustrates the system model and problem statement. Next, this article introduces the algorithm description and algorithm design respectively in section IV and V. The section VI evaluates the performance of the proposed algorithm through simulation experiments. The conclusion is given in Section VII.

II. RELATED WORK

Compared with cloud computing [9], MEC pushes computing and storage resources from the remote cloud to the network edge close to users, so as to realize local business localization and reduce delay and energy consumption. Service placement is a key issue in cloud computing and edge computing, leading to a lot of work in this area.

In [10], a virtual machine placement and migration method is proposed to minimize the consumption of data transmission time by optimizing service placement in the cloud. In [11], authors design and implement a data transmission application layer transmission protocol that uses cloud computing to manage mobile applications, reducing communication delay and realizing network balance. Authors of [12] study the latest trends in the field of multi-access edge computing combined with SDN(software-defined networking, SDN). In [13], in order to realize the efficient computing offload of mobile cloud computing, a game theory method is proposed. In [14], authors take a single user as an example and design a genetic algorithm for optimizing computational partitioning to determine whether to unload. A general guideline is proposed to determine unloading decisions in order to minimize the energy consumption of unloading [15]. Considering the multi-cell and multi-user MEC scenario, an adaptive offloading game method, which adjusts the number of offloading users, is proposed to avoid unexpected queuing delays [16]. Authors of [17] solve the problem of handling the

operation of autonomous MEC servers in a fully distributed IoT (Internet of Thing, IoT) network while improving the QoS satisfaction of mobile devices. Authors of [18] study the problem of multi-user computing offloading in mobile edge cloud computing under multi-channel wireless interference environment, and use game theory to solve this problem. The work [19] and [20] model the service migration problem as a MDP (Markov decision process, MDP). By using markov chains to predict user traffic to decide whether to migrate services. However, in some cases, Markov hypothesis is invalid [21]. In addition, authors of [7] formulate an online service placement strategy by considering various types of costs to minimize costs. However, this type of work only consider the costs related to services such as service exchange and service configuration costs, while ignoring the cost of users accessing the network. When a user randomly selects an access point to access the network without any optimization, there may be service-agnostic costs such as queuing delay. In particular, when too many users are simultaneously accessing the same access point [22], network congestion will occur, which is part of the main overhead of service communication.

This article jointly optimizes network selection and service placement. In addition to the cost [7], this article also considers the cost of users' access to the network, queuing cost and switching cost. This article designs an effect function on delay and energy consumption, and optimizes the whole MEC system with the goal of minimizing this function. Since the problem is NP-hard, there are currently a large number of studies on the application of swarm intelligence algorithms to solve NP-hard problems. In [23], a hybrid artificial bee colony algorithm is proposed in order to solve the parallel batch distributed flow shop problem where the work deteriorated. Authors of [24] design an extended ant colony optimization algorithm to deal with multi-peak optimization problems. Authors of [25] propose a fast co-evolutionary particle swarm optimizer based on a co-evolution framework and a particle swarm optimizer with a simple mutation operator to solve the optimization problem of multiple variables. In [26], authors combine the extended social forces model and the improved artificial bee colony algorithm to propose a new path planning method for emergency evacuation simulation. Based on these studies, this article proposes an optimization algorithm based on particle swarm algorithm to solve our research problem.

III. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, the paper first describes the system model of the study in Part A. Then, in Part B, Part C, and Part D, the access point selection model, service placement model and QoS model of this article are introduced respectively. Finally, the problem is formulated in Part E.

A. SYSTEM MODEL

This article considers a multi-access edge computing system consisting of M access points and K users. An access point can be viewed as an edge cloud. This article uses $M = \{0, 1, \ldots M\}$ and $K = \{0,1, \ldots K\}$ to represent edge cloud set and user set, respectively. To simplify the problem, this article assumes that each user has only one task in each time slot t, that is, the number of tasks is equal to the number of users K. In addition, this article adopts the communication model, channel gain formula and interference model in the literature [27]. This article considers a user-managed multi-access edge computing system, which means that users only know the local information, such as real-time location and computing requirements, but cannot observe global information. To better describe the characteristics of user movement, this article assumes that the service placement decision is made in a time slot structure, and the time axis is discretized into time frames $t \in T = \{0, 1, 2, \ldots, T\}$. At the beginning of t of each time slot, the mobile user determines a suitable computing node to perform the task in the neighboring AP (access point, AP) [28]. This article assumes that the user location remains unchanged and the network environment does not change in a short period of time. Table 1 lists the key parameter symbols of this article.

TABLE 1. Parameter table of system model.

Notation	Definition
M	Set of access point/edge clouds
$\Phi(t)$	Set of available access points for task at time slot t
$x_i^k(t)$	Whether the service of user k is placed on edge cloud i(=1) or not (=0)
$y_j^k(t)$	Whether the access point j is selected for user k to access the edge cloud j (=1) or not(=0)
$S_k(t)$	The service resource demand of user k
R_i	The maximum service resource capacity of each edge cloud i
P_j	The access point capacity of edge cloud j
$r_k(t)$	The access resource requirement of user k
S_t	Total handover delay for completing tasks at time slot t
S_0	The delay to complete a switch
Q_t	Total queuing delay to complete tasks at time slot t
q_0	The waiting time when there is a queued task in front
C_t	Total communication delay to complete the task at time slot t
$l_{ij}(t)$	The communication delay from access point j to edge cloud i.
P_t	Total Computing delay to complete the task at time slot t
D_t	Total transmission delay for completing tasks at time slot t
$Z(k)$	Set of task of user k

B. ACCESS POINT SELECTION MODEL

At each time slot, the system makes an AP selection decision for each task. The task can be executed on the user's local device or offloaded to other external nodes (i.e., edge server or remote cloud) for execution. Here, this article designs a binary vector $y_j^k(t)$ to represent the dynamic access point decision. If $y_j^k(t) = 1$, the task of user k selects access point j to access the network in time slot t; otherwise, $y_j^k(t) = 0$. Note that in a given time slot, each task is served by only one

AP. This article has the following AP constraints:

$$\sum_{j\in M} y_j^k(t) = 1, \quad \forall t \tag{1}$$

$$y_j^k(t) \in \{0, 1\}, \quad \forall j, t \tag{2}$$

The system cannot exceed the resource limit of the access point:

$$\sum_{k\in K} r_k(t) y_j^k(t) \le P_j, \quad \forall j, t \tag{3}$$

where $r_k(t)$ represents the access resource requirement of user k in time slot t, P_j represents the access point capacity of edge cloud j.

C. SERVICE PLACEMENT MODEL

As described above, the operator selects a suitable AP j for each task to access the edge cloud, and then proposes a service on the edge cloud i to provide service requirements for the task. In particular, for a single task, there is no necessary correlation between AP selection and service placement decisions. More specifically, the service of the task can be placed on any edge cloud $i \in M$, but the access point can only choose $j \in \Phi(t)$. This is because users have limited communication distance in the MEC system. Similar to the access point selection model, this article indicates that the service placement model is as follows:

$$\sum_{i\in M} x_i^k(t) = 1, \quad \forall t \tag{4}$$

$$\sum_{k\in K} s_k(t) x_i^k(t) \le R_i, \quad \forall i, t \tag{5}$$

$$x_i^k(t) \in \{0, 1\}, \quad \forall i, t \tag{6}$$

where $x_i^k(t)$ is the service placement decision variable. If $x_i^k(t) = 1$, it means that the service of user k is placed on the edge cloud i; otherwise, $x_i^k(t) = 0$. $s_k(t)$ represents the service resource demand of user k. R_i represents the maximum service resource capacity of each edge cloud i. Constraint (4) indicates that each task can only be assigned to one edge cloud. Equation (5) guarantees that the total number of services placed in each cloud cannot exceed the resource limit. Equation (6) indicates whether the task of use k is placed on the edge cloud i.

D. QUALITY OF SERVICE MODEL

1) SWITCHING DELAY

Due to the mobility of users, it may lead to the need to switch to other access points to obtain good user perception. In return, there will be a certain switching delay [29]. Assuming that S_0 is the delay caused by a switching, the overall switching cost of the task is:

$$S_t = \sum_{k\in K}\sum_{i\in M} S_0 \left[x_i^k(t) - x_i^k(t-1)\right]_+, \quad \forall i, t \tag{7}$$

2) QUEUING DELAY

For the number of users accessed by each access point changes over time, when the access point is preferentially selected according to the user's location, there may be too many access points to access the user, resulting in queuing problems. In order to better analyze the delay performance, this article takes the queuing delay into the model of this article. Given the queuing delay for tasks performed at time slot t:

$$Q_t = \sum_{k\in K}\sum_{j\in M} q_0 \left[y_j^k(t) - y_j^k(t-1)\right]_+, \quad \forall j, t \tag{8}$$

where q_0 is the waiting time when there is a queued task in front.

3) COMMUNICATION DELAY

In the model of this article, service placement and AP selection may not be in the same cloud. Obviously, this can reduce the pressure on some hotspot clouds, but at the same time, accessing services through the edge cloud will generate additional communication delays. Therefore, when considering the service placement decision $x_i^k(t)$ and the access point selection decision $y_j^k(t)$, the total communication delay of the system in time slot t can be expressed as:

$$C_t = \sum_{k\in K}\sum_{i\in M}\sum_{j\in\varphi(t)} x_i^k(t) y_j^k(t) l_{ij}(t), \quad \forall i, j, t \tag{9}$$

where $l_{ij}(t)$ represents the communication delay from access point j to edge cloud i.

4) COMPUTING DELAY

For each task, it can be performed on the user's local device or offloaded to other computing nodes. Because the calculation demand of the task is affected by many factors, such as the location of the user and the status of the network. Therefore, this article represents the calculated demand of user k as $\lambda_k(t)$ that changes with time, and $C_j(t)$ represents the computing power (i.e., the CPU cycle per second) that can perform the task on the compute node i at time slot t. Therefore, the computational delay can be expressed as:

$$P_t = \sum_{k\in K} [\lambda_k(t) / \sum_{j\in M} C_j(t) y_j^k(t)], \quad \forall j, t \tag{10}$$

5) TRANSMISSION DELAY

The user's tasks are uploaded from the local device to the access point through the wireless channel. This article uses $h_{k,i}^t$ to denote the channel gain between user k and access point i in time slot t. Because this article divides the time T into small time slots, assuming that the user has only one task and the user hardly moves in each time slot t, then $h_{k,i}^t$ can be regarded as a constant. Define the user's transmission power as p_{tan}, then the transmission rate from user k to access point i in time slot t can be expressed as:

$$v_{k,i}^t = W \log_2\left(\frac{1 + p_{\tan} h_{k,i}^t}{\sigma^2 + I_{k,i}^t}\right), \quad \forall k, t \tag{11}$$

where W is the channel bandwidth, σ^2 is the noise power, and $I^t_{k,i}$ is the inter-cell interference power of the user k access to the edge cloud i in the time slot t.

Then in time slot t, the total transmission delay from the user to the access point can be expressed as:

$$D_t = \sum_{k \in K} \frac{\lambda_k(t)'}{v^t_{k,i}}, \quad \forall t \tag{12}$$

where $\lambda_k(t)$' represents the task length of user k in time slot t. As compared with the task length, the length of the task execution result can be ignored. Therefore, this article only calculates the time delay and energy consumption of the task upload to the access point, and ignores the time delay and energy consumption of the task return.

The transmission energy consumption of the task in time slot t can be expressed as the transmission power p_{tan} of the task multiplied by the transmission delay D_t:

$$e_t = D_t \times p_{tan}, \quad \forall t \tag{13}$$

6) ENERGY CONSUMPTION

In the time slot t, the energy consumption of the task execution includes the energy consumption of the user transmitting data to the MEC server and the task execution in the edge cloud. For ease of expression, this article uses $O_j(t)$ to represent the energy consumption of the task performed on the computing node j. Hence, the energy consumption at time slot t can be expressed as:

$$E_t = \sum_{k \in K} \sum_{j \in M} y^k_j(t) O_j(t) + e_t, \quad \forall j, t \tag{14}$$

E. PROBLEM FORMULATION

At present, most researches focus on minimizing the total delay of data transmission and task execution under the energy consumption or minimizing the energy consumption of the mobile terminal under the performance that satisfies the user's perceived delay. In order to better balance the time delay and energy consumption, this article considers the weighted summation of time delay and energy consumption to obtain the minimum value, so that the total cost of the overall system composed of the mobile terminals and the edge server is the smallest. Therefore, in a given time frame T, the problem can be expressed as follows:

$$min \sum_{t=1}^{T} \lambda^t_1 (S_t + Q_t + C_t + P_t + D_t) + \lambda^t_2 E_t, \quad \forall i, j, t, \tag{15}$$

$$\lambda^t_1 + \lambda^t_2 = 1$$

$$\text{Subject to } (1) - (6), (16) \tag{16}$$

where $\lambda^t_1, \lambda^t_2 \in [0, 1]$ represent the weighting coefficients of the calculation time and energy of the service.

IV. ALGORITHM DESCRIPTION

Because the problem of network selection and service placement is NP-hard [30], this article uses PSO algorithm to solve it. This algorithm is good at NP-hard problem optimization [31]. This part first introduces the PSO algorithm. Then a joint network selection and service placement algorithm based on PSO algorithm is proposed.

A. PARTICLE SWARM OPTIMIZATION ALGORITHM

In the PSO algorithm [32], each particle flies through the search space at a certain speed. The velocity and position of the particles vary with the flight experience of the individual and companions. By this mechanism, each particle is updated towards the region of good solutions. Let V_{id} be the velocity of the particle and X_{id} be the position of the particle. The update formula of particle motion is:

$$V_{id} = w \times V_{id} + c_1 \times r_1 \times (P_{id} - X_{id}) + c_2 \times r_2 \times (P_{gd} - X_{id}) \tag{17}$$

$$X_{id} = X_{id} + V_{id} \tag{18}$$

where w is the inertia weight, P_{id} is the best historical position of particles, P_{gd} is the best global position of particles, c_1 and c_2 are learning factors, also known as the acceleration constant, r_1 and r_2 are uniform random numbers within the range of [0,1]. The three parts in formula (17) respectively represent that the particle has a tendency to maintain its previous velocity, a tendency to approach its own historical best position and a tendency to approach the best group position. This article sets $c_1 = c_2$, that is, this article believes that the local optimal and global optimal have the same impact on particle renewal. Formula (18) updates the positions of particles in the population.

B. JOINT NETWORK SELECTION AND SERVICE PLACEMENT ALGORITHM BASED ON PARTICLE SWARM OPTIMIZATION

This article designs a network selection and service placement algorithm based on particle swarm optimization. According to the PSO algorithm, the network selection and service placement of each particle serve as a solution to this problem. It can be represented by a matrix as follows:

This article assumes there are K users, M base stations and G services in the system. Each particle is defined as a $(K + G)^*M$ matrix X, where first K rows represent user network location choice, and last G rows indicate the user service place. In the matrix X, if $X[a][b] = 1$ ($0 < a \leq K$), user a select edge cloud b as access point selection. If X[a][b] $= 1$ ($K < a \leq K+G$), service a placed in the edge of cloud b, X[a][b]$= 0$ indicates tasks or services are not placed on this edge cloud.

The PSO-JNSSP algorithm (joint network selection and service placement algorithm based on PSO, PSO-JNSSP) is described as follows:

Step 1: Initialize n particles. Each particle contains the network selection of K users and service placement location of G services.

Step 2: To calculate the fitness value of each particle, this article chooses the objective function (15) in this

article as the fitness function. The particle with the highest fitness value is found, and the network selection and service placement location of the particle are considered as the global optimal solution P_{gd}.

Step 3: For each particle, this article gets a network selection and a service placement for each user in step 1, and each base station has its own coordinate. This article updates the particle velocity (the step size of particle renewal) use equation (17), and then calculates a new coordinate with equations (18). Obviously, the new coordinate are not necessarily the location coordinates of a base station. This article designs the transition probability formula (19) based on the distance d_i and the queue length $path_i$, where d_i is the distance between the coordinates calculated and the coordinates of the base station, and $path_i$ is the queue length of each base station. This article uses the probability to select the network selection and service placement locations for this iteration. Details are in the section 5.2.

Step 4: To judge whether the termination condition is reached, if the number of iterations m reaches the given maximum value, or the optimal solution does not change for a period of time, then go to step 5. Otherwise go to step 2.

Step 5: Returns the optimal network selection and service placement.

The algorithm first generates randomly the network selection and service placement of n groups of users (i.e., n particles), finds a particle with high fitness, and then uses the particle to transform the other particles in a favorable direction. In this process, each particle flies in a known direction, selects a good transformation in its neighborhood, maintains a good solution, and optimizes network selection and service placement.

V. ALGORITHM DESIGN

In this section, this article introduces the PSO-JNSSP algorithm design. Firstly, this article introduces the particle initialization Part A. Secondly, in Part B, this article describes the particle optimization process in detail. Finally, this article analyzes the complexity of PSO-JNSSP algorithm.

A. PARTICLE INITIALIZATION

The first important question to solve is how the algorithm produces the particles in the first place. This article produces the following random particles.

For a mobile user, their location should be considered first. As we all know, each base station has its own service range, beyond the service range, the service cannot be provided. When users choose network selection base station, it must choose from the base station which can affect it. When the distance between the base station and the user is less than the defined value, the base station i can be selected as the network access point of the mobile user. If there are multiple optional base stations at the same time, a user chooses one of them randomly. Then, the corresponding service is placed on a base station j, which can be the same as i or other base stations. Besides, the user's network selection and service placement should satisfy the constraints (3) and (5) respectively. This is the process of selecting network selection and service placement for all mobile users. Then generates a $(K+G)^*M$ matrix X corresponding to this particle, where M is the number of base stations in the user range, G is the number of services and K is the number of mobile users. The first K rows of matrix X represent the user's network selection location, and the last G rows represent the user's service placement location.

Randomly initialize n particles, and each particle represents the network selection and service placement of all k mobile users.

B. PARTICLE OPTIMIZATION PROCESS

Formula (15) was selected as the fitness function of this study. The fitness function mainly involves the task delay and energy consumption constraints. In order to eliminate the dimensional influence between delay and energy consumption, this article uses min-max standardization to carry out linear transformation on the original data, so that all values are processed to between [0, 1], and then carry out weighted summation. The min-max standardization indicates as $X^*= (X\text{-}min)/(max\text{-}min)$. Max and min are the maximum and minimum values of the original sample data, respectively. X is the current value that needs to change, X^* is the changed value. λ_1^t and λ_2^t is the corresponding normalized weighted function. In a real algorithm, different values can be set according to the needs of the user.

According to the current matrix, the fitness function of the particle is calculated, and the historical optimal location P_{id} of the particle and the global optimal location P_{gd} can be obtained. For the base stations corresponding to the global optimal and local optimal solutions, each base station has a three-dimensional coordinate. This article substitutes these coordinates into formulas (17) and (18) to obtain the optimized network selection and service placement coordinates of each particle. Obviously, the optimized coordinates this article calculates and the coordinates of the base station in the real scene are not necessarily the same. To solve this problem, this article designs a transition probability as a criterion for further selection of network selection and service placement. This article considers two properties to calculate the probability, one is the distance d_i between each base station i and the optimized coordinates this article calculated, another is the number of tasks queuing on the base station i expressed as $path_i$, and select the appropriate network selection and service placement point from base station M according to the probability:

$$Pi = \frac{(\frac{1}{di \times pathi})}{\sum_{i \in n'} d_i \times path_i} \tag{19}$$

Algorithm 1 The Overall Flow of PSO-JNSSP Algorithm

Input:
Number of tasks k;
Base station number m;
Task position (X1,Y1,Z1);
Base station location (X2,Y2,Z2);
Output: Optimal network selection and service placement location
Initialization:
Randomly generate each particle
Optimization:
repeat
for each particle
Optimize network selection and service placement
Update the P_{id};
Update the P_{gd};
end for
until stopping criterion is satisfied

Algorithm 2 The Initialization Process of Each Particle

for each task
Calculate the distance between the task and the edge cloud d_i
if the distance d_i <10
add the base station to the optional base station matrix Available
end if
end for
for each task
Initialize network selection for the task
Randomly select a base station from the available base stations as the network selection for the task
end for
for each task
Initialize service placement for the task
Randomly select a base station from the available base stations as the service placement for the task
end for

when distance d_i is smaller, the probability of selecting this base station is greater; when the queue length is shorter, the more probability to select this base station. If the distance d_i is greater than the defined value, the previous network selection or service placement point is maintained. If the distance d_i is shorter than the value, this study adds the base station to n'. Hence, n' is all solution that meet the requirement.

In addition, this study uses Dijkstra algorithm to calculate the optimal path from network selection point a to service placement point b. Dijkstra algorithm [33] is a typical single-source shortest path algorithm. Network selection and service placement may be on the same base station, or through n ($n >= 0$) base stations switching to other base stations to obtain services. If the distance directly from a to b is the shortest, there is no need to switch to other base stations in the middle, otherwise, there may be 1 or n base stations in the middle. Based on this, the switching delay from the network selection point to the service placement point can be calculated. Since the transmission time of data sent to the transmission medium is negligible compared with the time of data transmitted to the transmission medium by electromagnetic or optical signals, only the transmission delay is considered in this study.

The proposed algorithm for network selection and service placement based on particle swarm optimization is described in algorithm 1, 2 and 3 respectively.

Algorithm 3 Particle Network Selection and Service Placement Optimization Process

for each particle
for each task
Calculate the delay-energy fitness according to formula (15)
end for
end for
Update individual history optimal location P_{id};
Update the global best location P_{gd};
Updates the optimal network selection and service placement for each particle in this iteration according to formulas (17) and (18)
for each particle
for each task
Calculate the distance between each task and the base station
Assign the best network selection and service placement point for each task according to the probability in equation(19)
end for
end for

C. ANALYSIS OF COMPUTATIONAL COMPLEXITY

The computational complexity analysis of the PSO-JNSSP algorithm is as follows:

The computational complexity of the initialization process is $O(K \times M^2 \times n)$, where K is the number of mobile users, M is the number of base stations, and n is the number of particles.

The main computational complexity of the algorithm proposed in this article lies in the optimization process of Algorithm 3. In each particle, a base station that satisfies constraints (1) - (6) needs to be selected as an optional network selection and service placement base station set, and then the most suitable network selection and service placement base station is selected for each user to perform the task Uninstall. Therefore, the computational complexity of each particle in the optimization process is $O(K \times M^2)$, and the computational complexity of all particles is $O(K \times M^2 \times n)$.

TABLE 2. The simulation parameters.

Simulation parameters	values
Computation capacity of each edge server	2.5GHZ
Service resources capacity of each edge server	2.8GHZ
The AP resource demand of each tasks	0-10 bit
The service resource demand of each tasks	0-15bit
The transmission speed from mobile device to edge server	180kHZ
The transmission speed between edge servers	180KHZ
The CPU cycles required for each task calculation	500-1000 cycles
Energy consumption of edge servers	1J/G cycles

The computational complexity of the particle updating its velocity and position is $O(K \times n)$. Therefore, the computational complexity of the entire algorithm is $O(K \times M^2 \times n)$.

VI. SIMULATIONS

In this section, the paper will show the performance of the algorithm through experimental data. First, the paper describes the simulation design of the experiment in Part A. Then perform performance analysis and convergence analysis in Part B and Part C. Part D shows the computational complexity analysis.

A. SIMULATION DESIGN

This article uses the three-dimensional area with size parameter 3km*3km*3km. There are multiple users and multiple edge servers in this space. The specific number is described in detail in the following section. The number of tasks that can be accessed by each edge server cannot exceed its maximum accessible capacity of 2.5GHZ, and the number of users that each edge server can serve cannot exceed the maximum service resource capacity of the edge server of 2.8GHZ. The simulation experiment in this article simulates a delay-sensitive system with λ_1^t set to 0.8 and λ_2^t set to 0.2. During the simulation, unless otherwise stated, the parameter settings are shown in Table 2.

In order to evaluate the performance of the algorithm, this article compares it with other benchmark algorithms: PSOCA(single-objective calculation algorithm, PSOCA), RCA(random calculation algorithm, RCA), LCA(local calculation algorithm, LCA). PSOCA is calculated based on particle swarm algorithm, while its goal only considers one aspect, the energy consumption. In RCA, users randomly make decisions about access point and service placement selection, and edge servers also randomly allocate computing resources to users. In LCA, all users choose local computing. This article evaluates four cases based on the distribution of edge severs and tasks.

B. PERFORMANCE ANALYSIS

This section mainly shows the comparison between the PSO-JNSSP algorithm and the four kinds of algorithms mentioned above, and show you through the different settings of the base stations and tasks, and verify the effectiveness of the algorithm.

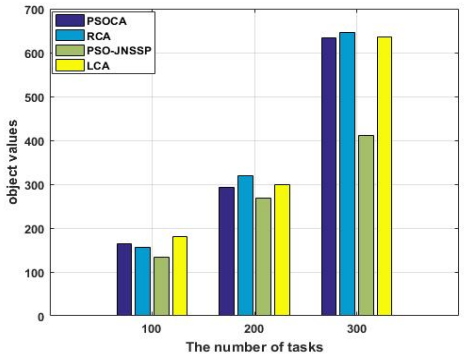


FIGURE 2. The cost with the change of task number when edge servers evenly distributed.

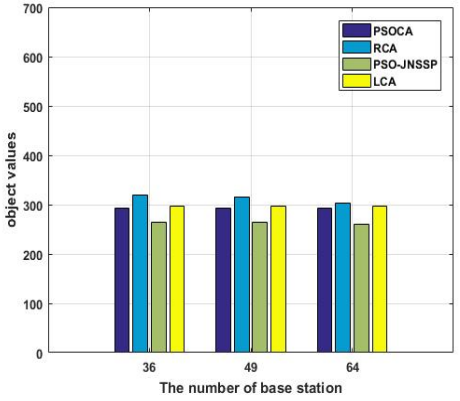


FIGURE 3. The cost with the change of task number when edge servers unevenly distributed.

Figure 2 shows the performance of four kinds of algorithms, under the circumstances of 49 edge servers evenly distributed, and the number of tasks changes from 100 to 300. It can be seen that when the number of users increases, the sum of its delay and energy consumption also increases. There are two reasons. On one hand, the number of tasks increases, which will lead to the growth of energy consumption. On the other hand, as the number of users increases, each edge server will be allocated more tasks, so that the calculation delay and queuing delay will increase relatively.

In Figure 3, edge servers are unevenly distributed. 30 of them in the 1.5*3*3 area and other 19 edge servers are distributed in the other part of the area. The number of tasks is 100, 200, 300, and the tasks are evenly distributed. Figure 3 describes the changes in the system cost as the number of users changes. It can be seen from the Figure 3 that although the edge servers are unevenly distributed, similar to Figure 2, the value of the objective function also increases as the task increases. As the task increases, the overall energy consumption increases. And the edge servers are not evenly distributed, there is a high possibility that the communication delay will increase significantly.

In Figure 4, edge servers and users are evenly distributed. The number of tasks is 200, while edge servers change

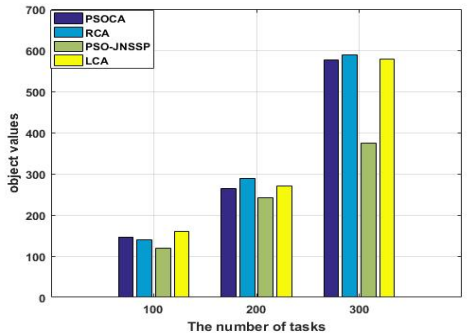


FIGURE 4. The cost with the change of edge server number when tasks evenly distributed.

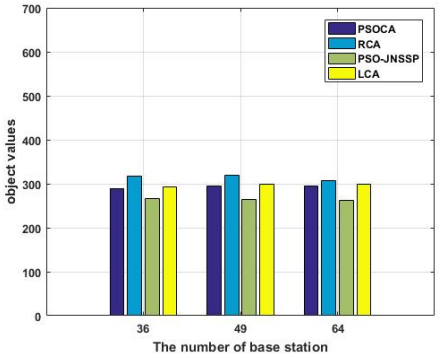


FIGURE 5. The cost with the change of edge server number when tasks unevenly distributed.

from 36 to 64. Figure 4 describes the cost when the number of base stations changes. It can be found from Figure 4 that as the number of base stations increases, the trend of the PSO-JNSSP algorithm's objective function value declines slowly. There are two reasons lead to this result. First, the number of tasks is set more, and the increase of base stations is small, which has little effect on the overall scenario. Second, the growth of the number of edge server brings more choice for access point selection and service placements. The more choices result in the better solution.

Figure 5 describes the change of system cost when the number of base station changes. Edge servers are distributed randomly, value 36, 49, 64 respectively, and 150 users are in 1.5*3*3, 50 users are in another area. It can be seen from the Figure 5 that similar to the Figure 4, although the value of the objective function drops, the decrease is not obvious, and even tends to be flat.

As shown in Figure 2, 3, 4, and 5, the performance of the PSO-JNSSP algorithm is superior to the other three algorithms. In RCA, the decision-making of the decision-making organization is random, which may cause serious interference between adjacent edge servers. In this case, the delay and energy consumption may be very large, so that the expected effect cannot be obtained. The performance is the most unsatisfactory of these algorithms. Although the PSOCA algorithm also has better performance, it only takes one of energy consumption or delay as the target, and cannot

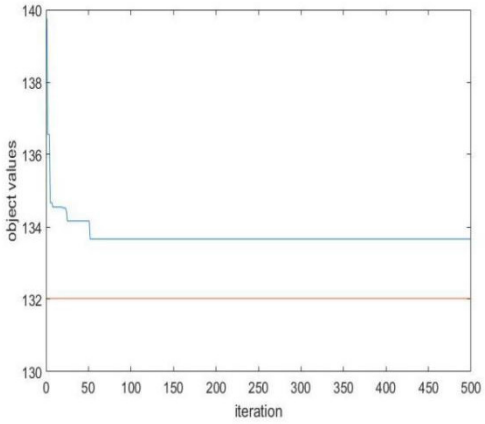


FIGURE 6. The iteration of PSO-JNSSP and exhaustive algorithm.

optimize the two together very well. The performance is slightly worse than the PSO-JNSSP algorithm. In the LCA algorithm, users all choose local computing. Because the capability of mobile devices is very different from edge servers, the computing delay and energy consumption will increase severely. Although the communication delay may be relatively reduced, it has little effect. Therefore, the performance of the PSO-JNSSP algorithm has better performance than the other three algorithms.

C. CONVERGENCE ANALYSIS

To illustrate the distance between the proposed algorithm and the optimal solution, this article performed performance simulation on the optimal solution obtained by the proposed algorithm and the exhaustive algorithm. This article randomly generates 49 edge servers and 300 tasks in the defined area, their respective parameters are shown in Table 2.

In order to prove the convergence of the PSO-JNSSP algorithm, this article compares its iterative process with the exhaustive algorithm, and found that our algorithm has good convergence and robustness. As show in Figure 6, first, the PSO-JNSSP algorithm converges. Second, although the algorithm in this article is not optimal, as the iteration progresses, the gap between the algorithm and exhaustive algorithm becomes smaller and smaller. Third, it can be seen that there has been a significant drop in the first 50 iterations. After the number of iterations shown in Figure 6, the performance can be improved and closer to the optimal result. Therefore, there is a trade-off between acceptable performance and time complexity.

D. COMPUTATIONAL COMPLEXITY ANALYSIS

This article verifies the computational complexity of the proposed PSO-JNSSP algorithm on a window server equipped with AMD A10-7300 Radeon R6,10 Compute Core 4G+6G 1.90GZ processors and 4GB RAM. This article evaluates the computational complexity of the proposed algorithm by measuring the time for the objective function to reach the convergence value. Table 3 shows that when the tasks and BSs

TABLE 3. Convergence time of PSO-JNSSP.

Number of Tasks \ Number of BSs (Convergence Time (s))	16	25	36
20	0.183	0.551	0.802
40	0.336	0.777	1.192
60	0.559	1.056	1.563

(base stations, BSs) are uniformly distributed, the number of tasks is 20, 40, 60 and the number of base stations is 16, 25, 36, and the time it takes for the objective function to reach convergence. It can be seen from Table 3 that the proposed algorithm meets the very limited running time requirements and reflects the efficiency of the algorithm.

VII. CONCLUSION

This article studies the problem of access point selection and service placement based on particle swarm optimization. This scheme jointly optimizes both delay and energy consumption, considering the influence of computing delay, communication delay, queuing delay, and overall energy consumption. First, a system model of multi-access edge computing is described. It uses energy consumption and delay as optimization goals and chooses user access points and service placement. Secondly, this article uses particle swarm optimization to design a PSO-JNSSP algorithm to solve this problem. Then it analyzes the performance and convergence of the PSO-JNSSP problem. Finally, simulation results show that the algorithm has better performance than other baseline algorithms.

ACKNOWLEDGMENT

(Shuyue Ma and Shudian Song contributed equally to this work.)

REFERENCES

[1] F. Liu, P. Shu, H. Jin, L. Ding, J. Yu, D. Niu, and B. Li, "Gearing resource-poor mobile devices with powerful clouds: Architectures, challenges, and applications," *IEEE Wireless Commun.*, vol. 20, no. 3, pp. 14–22, Jun. 2013.

[2] L. Best-Rowden and A. K. Jain, "Longitudinal study of automatic face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 148–162, Jan. 2018, doi: 10.1109/TPAMI.2017.2652466.

[3] S. Sun, C. Luo, and J. Chen, "A review of natural language processing techniques for opinion mining systems," *Inf. Fusion*, vol. 36, pp. 10–25, Jul. 2017, doi: 10.1016/j.inffus.2016.10.004.

[4] C. Wang, Y. He, F. R. Yu, Q. Chen, and L. Tang, "Integration of networking, caching, and computing in wireless systems: A survey, some research issues, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 7–38, 1st Quart., 2018.

[5] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[6] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5G networks: New paradigms, scenarios, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 54–61, Apr. 2017.

[7] L. Wang, L. Jiao, T. He, J. Li, and M. Muhlhauser, "Service entity placement for social virtual reality applications in edge computing," in *Proc. IEEE INFOCOM*, Apr. 2018, pp. 468–476.

[8] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 3rd Quart., 2017.

[9] H. Li, G. Shou, Y. Hu, and Z. Guo, "Mobile edge computing: Progress and challenges," in *Proc. 4th IEEE Int. Conf. Mobile Cloud Comput., Services, Eng. (MobileCloud)*, Mar. 2016, pp. 83–84.

[10] J. T. Piao and J. Yan, "A network-aware virtual machine placement and migration approach in cloud computing," in *Proc. 9th Int. Conf. Grid Cloud Comput.*, Nov. 2010, pp. 87–92.

[11] F. Liu, P. Shu, and J. C. S. Lui, "AppATP: An energy conserving adaptive mobile-cloud transmission protocol," *IEEE Trans. Comput.*, vol. 64, no. 11, pp. 3051–3063, Nov. 2015.

[12] G. Mitsis, P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, "Intelligent dynamic data offloading in a competitive mobile edge computing market," *Future Internet*, vol. 11, no. 5, p. 118, May 2019.

[13] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015, doi: 10.1109/TPDS.2014.2316834.

[14] L. Yang, J. Cao, S. Tang, T. Li, and A. T. S. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing," in *Proc. IEEE 5th Int. Conf. Cloud Comput.*, Jun. 2012, pp. 794–802.

[15] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, Apr. 2010.

[16] M. Deng, H. Tian, and X. Lyu, "Adaptive sequential offloading game for multi-cell mobile edge computing," in *Proc. 23rd Int. Conf. Telecommun. (ICT)*, May 2016, pp. 1–5.

[17] A. Pavlos, E. E. Tsiropoulou, and S. Papavassiliou, "Game-theoretic Learning-based QoS Satisfaction in Autonomous Mobile Edge Computing," in *Proc. Global Inf. Infrastruct. Netw. Symp. (GIIS)*, 2018, pp. 1–5, doi: 10.1109/GIIS.2018.8635770.

[18] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016, doi: 10.1109/TNET.2015.2487344.

[19] A. Ksentini, T. Taleb, and M. Chen, "A Markov decision process-based service migration procedure for follow me cloud," in *Proc. IEEE ICC*, Jun. 2014, pp. 1350–1354.

[20] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic service migration in mobile edge-clouds," in *Proc. IFIP/IEEE Netw. Conf.*, May 2015, pp. 1–9.

[21] M. Srivatsa, R. Ganti, J. Wang, and V. Kolar, "Map matching: Facts and myths," in *Proc. 21st ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, 2013, pp. 484–487.

[22] A. J. Nicholson, Y. Chawathe, M. Y. Chen, B. D. Noble, and D. Wetherall, "Improved access point selection," in *Proc. 4th Int. Conf. Mobile Syst., Appl. Services*, 2006, pp. 233–245.

[23] J.-Q. Li, M.-X. Song, L. Wang, P.-Y. Duan, Y.-Y. Han, H.-Y. Sang, and Q.-K. Pan, "Hybrid artificial bee colony algorithm for a parallel batching distributed flow-shop problem with deteriorating jobs," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2425–2439, Jun. 2020, doi: 10.1109/TCYB.2019.2943606.

[24] Q. Yang, W.-N. Chen, Z. Yu, T. Gu, Y. Li, H. Zhang, and J. Zhang, "Adaptive multimodal continuous ant colony optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 191–205, Apr. 2017, doi: 10.1109/TEVC.2016.2591064.

[25] X. Zheng and H. Liu, "A scalable coevolutionary multi-objective particle swarm optimizer," *Int. J. Comput. Intell. Syst.*, vol. 3, no. 5, p. 590, 2010, doi: 10.1080/18756891.2010.9727725.

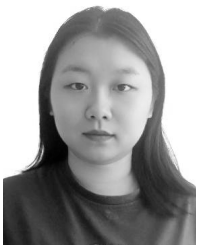
[26] H. Liu, B. Xu, D. Lu, and G. Zhang, "A path planning approach for crowd evacuation in buildings based on improved artificial bee colony algorithm," *Appl. Soft Comput.*, vol. 68, pp. 360–376, Jul. 2018, doi: 10.1016/j.asoc.2018.04.015.

[27] C. Singhal and S. De, *Resource Allocation in Next-Generation Broadband Wireless Access Networks*. Hershey, PA, USA: IGI Global, 2017, doi: 10.4018/978-1-5225-2023-8.

[28] Y. ThomasHou, Y. Shi, and H. D. Sherali, "Optimal base station selection for anycast routing in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 55, no. 3, pp. 813–821, May 2006.

[29] J. Wu, E. W. M. Wong, Y.-C. Chan, and M. Zukerman, "Energy efficiency-QoS tradeoff in cellular networks with base-station sleeping," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–7.
[30] B. Gao, Z. Zhou, F. Liu, and F. Xu, "Winning at the starting line: Joint network selection and service placement for mobile edge computing," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Paris, France, Apr. 2019, pp. 1459–1467, doi: 10.1109/INFOCOM.2019.8737543.
[31] J. Kennedy and R. Eberhart, "Particle swarm optimizatio," in *Proc. IEEE Int. Conf. Neural Netw.*, Perth, WA, Australia, Jun. 1995, pp. 1942–1948.
[32] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Hum. Sci. (MHS)*, Nagoya, Japan, 1995, pp. 39–43.
[33] D. James, "A Dijkstra's algorithm shortest path assignment using the Google maps API: Poster session," *J. Comput. Sci. Coll.*, vol. 25, pp. 253–255, Jun. 2010.

SHUYUE MA is currently pursuing the master's degree with the School of Information Science and Engineering, Shandong Normal University. Her current research interests include offloading algorithm, edge computing, and the Internet of Things (IoT).



SHUDIAN SONG is currently pursuing the master's degree with the School of Information Science and Engineering, Shandong Normal University. Her current research interests include offloading algorithm, edge computing, and the Internet of Things (IoT).

JINGMEI ZHAO (Member, IEEE) received the B.S. degree in electronic information science and technology from Liaocheng University, in 2007, the M.S. degree from Liaoning Technical University, in 2010, and the Ph.D. degree in electronic science and technology from the Beijing University of Posts and Telecommunications, in 2017. She worked as a Researcher, from 2010 to 2013. She is currently an Instructor with the School of Information Science and Engineering, Shandong Normal University. Her current research interests include power amplifier linearization, microwave wireless communications, and distributed network optimization.

LINBO ZHAI received the B.S. and M.S. degrees from the School of Information Science and Engineering, Shandong University, in 2004 and 2007, respectively, and the Ph.D. degree from the School of Electronic Engineering, Beijing University of Posts and Telecommunication, in 2010. He has been a Teacher with Shandong Normal University. His current research interests include cognitive radio, crowdsourcing, and distributed network optimization.

FENG YANG received the B.S. and M.S. degrees in radio electronics from Shandong University, in 1985 and 1988, respectively. He is currently a Professor with the School of Information Science and Engineering, Shandong Normal University, and the Director of the Department of Communication Engineering. He has published more than 30 articles, edited textbooks and edited five books, obtained seven national invention patents, four utility model patents, and participated in one national fund project. He has presided over five provincial and university-level education reform projects. He was a recipient of the one Provincial-Level Teaching Achievement Award, the one School-Level Teaching Achievement Award, and three provincial awards for scientific and technological progress.