

Research Article

An Optimization Method for Mobile Edge Service Migration in Cyberphysical Power System

Qian Cao ¹, **Qilin Wu** ¹, **Bo Liu** ¹, **Shaowei Zhang** ², and **Yiwen Zhang** ³

¹School of Information Engineering, Chaohu University, Chaohu 238000, China

²School of Computer Science and Technology, Anhui Wenda University of Information Engineering, Hefei 230000, China

³School of Computer Science and Technology, Anhui University, Hefei 230000, China

Correspondence should be addressed to Qilin Wu; lingqiw@126.com

Received 1 December 2020; Revised 14 January 2021; Accepted 31 January 2021; Published 15 February 2021

Academic Editor: Mohammad R. Khosravi

Copyright © 2021 Qian Cao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To relieve the pressure of processing computation-intensive applications on mobile devices and avoid high latency during data transmission, edge computing is proposed to solve this problem. Mobile edge computing (MEC) allows the deployment of MEC servers at the edge of the network to interact with users on the premise of low transmission delay, thereby improving the quality of service (QoS) for users. However, due to the high mobility of users, with the continuous change of geographical location, when users exceed the signal range of the MEC server, the services they request on the MEC server will also be migrated to other MEC servers. The handoff process may involve high response delays caused by service forwarding, thereby greatly degrading QoS. For the above problems, in this paper, a service migration optimization method based on transmission power control is proposed. First, according to the transmission power of the MEC server, the user's activity range is divided into multiple subregions based on a Voronoi diagram. Therefore, there is one MEC server in each subregion, and the size of each subregion is adjusted by controlling the transmission power of the MEC server to minimize the number of wireless handoffs and the energy consumption of the MEC server. Then, the particle swarm optimization (PSO) is adopted to solve the above multiobjective optimization problem. Finally, the effectiveness of the proposed method is verified through extensive experiments.

1. Introduction

Nowadays, with the rapid development of mobile devices, mobile applications are becoming more and more complex, and mobile devices with limited resources usually cannot meet the needs of most applications. Therefore, the industry began to consider offloading such computation-intensive applications to the cloud [1]. However, the remote offloading in traditional cloud computing may involve high latency and cannot meet the low-latency requirements [2] of some latency-sensitive applications, including augmented reality (AR) and remote game control [3]. Meanwhile, the exponential growth of information caused by a large number of devices and applications has brought tremendous pressure to remote information transmission. To solve the above problems, mobile edge computing (MEC) has been proposed, and a large number of servers are placed at the network edge [4, 5]. MEC is regarded as a supplement to

mobile devices with relatively limited computational and storage capacity, which can enable computation offloading and provide services to users. In MEC, a new computing device called an MEC server, which is deployed on the base station to provide services and computing resources for users, is deployed at the network edge to act as a small cloud data center, giving the network edge the ability to process data [6]. Clearly, MEC servers can provide users with cloud services closer to the end-users so that users can request services with low latency.

Service providers can deploy related services on the MEC server to improve user experience, expand the user market, and earn more benefits. The reason is that the use of the MEC server helps users' mobile devices meet the performance requirements of some applications, greatly reduces information transmission delays, improves users' QoS, and reduces the traffic between users and the core network, thereby reducing operating cost. However, in the mobile edge

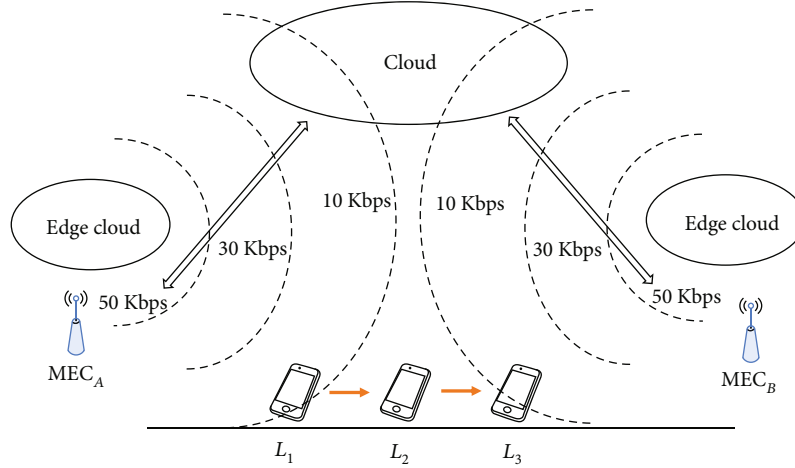


FIGURE 1: Example of edge service migration in mobile network.

computing environment, mobile devices often have high mobility, and service migration operations may be involved due to the time-varying location of their users. As shown in Figure 1, the edge computing system structure includes two MEC servers, MEC_A and MEC_B , and the signal ranges corresponding to different powers are shown in the dotted arc in the figure. While the user moves from L_1 via L_2 to L_3 , when the user walks out of the signal range of the MEC_A server and enters the signal range of the MEC_B server, the service requested by the user will be migrated from MEC_A to MEC_B . Before the end of the handoff, the user can only use the services of the MEC_A . When the service request is forwarded from MEC_A to MEC_B through a limited capacity backhaul, the response time of the service will be significantly increased [7], resulting in a significant decrease in the quality of user experience.

Due to the limited coverage of the MEC server and the high time-varying location of users, users may switch servers frequently in the process of using mobile devices. So, the forwarding time and downtime of the request involved in this process will degrade QoS. At present, some studies propose to reduce the number and probability of service migration by deploying service copies in advance. However, at the same time point, users can only request services from one MEC server. The MEC servers that are not accessed by users but have deployed service copies will be occupied with unnecessary storage resources, resulting in a waste of resources.

In this paper, we focus on the problem of how to improve user QoS and effectively reduce server energy consumption when edge users have high mobility. Since the number of wireless handoffs and the energy consumption of the MEC are related to the coverage of the MEC server, it is necessary to reasonably control the coverage of the MEC server. However, the coverage of the MEC server is closely related to the transmission power of the MEC server. Therefore, by using transmission power control for service migration optimization, we can minimize the number of wireless handoffs and the energy consumption of an MEC server through service migration optimization. That is our motivation.

Compared with the existing methods, our main contributions can be summarized as follows:

- (1) A service migration optimization method is proposed based on transmission power control. This method adjusts the size of each subarea according to the transmission power of the MEC server, so as to achieve the goal of minimizing the number of wireless handoffs and energy consumption of the MEC server
- (2) The experimental scene is modeled by using the Voronoi diagram, and the multiobjective optimization problem is transformed into a single-objective optimization problem by using the weight coefficient transformation method. Furthermore, the PSO algorithm is used to solve the optimization problem, so as to achieve the goal of minimizing the user wireless handoff times and minimizing the energy consumption of the MEC server
- (3) A large number of simulation experiments were carried out using a real base station data set, the Telecom Dataset [8–10], under the assumption that the user's mobile path is known, which verified the effectiveness and efficiency of the algorithm in this study

The remainder of this paper is organized as follows. Section 2 discusses and summarizes related work. In Section 3, we introduce the system model. After that, we introduce the PSO optimization method for minimizing the number of wireless handoffs of user equipment and minimizing the energy consumption of the MEC server in Section 4. Then, we give the experimental results and analysis in Section 5. Finally, the paper is concluded in Section 6.

2. Related Work

With the rapid development of the mobile Internet and the Internet of Things, a large number of delay-sensitive and computation-intensive applications have emerged. To meet

the low-latency and high-performance requirements, edge computing was proposed to provide solutions. Since the coverage of the MEC server is limited, as the user moves, the edge nodes that the user can connect to also change. If the user's service is always located on the MEC server where the user initially connected, the user's service request should be forwarded from the MEC server to the original MEC server through the backhaul link, which will increase the service delivery delay. Therefore, in order to maintain the user QoS, the edge service should be dynamically migrated between multiple MEC servers along with the user's movement.

At present, many studies have contributed to reducing service migration time. Taleb and Ksentini [11] proposed a Follow-Me Cloud (FMC) analysis model. In this work, the Markovian mobility model was proposed to analyze the performance of MEC when users move, but they did not consider whether or not services are migrated and where to migrate. In the scenario of a one-dimensional mobile model, Ksentini et al. [12] used the Markov decision process (MDP) to decide whether or not to carry out service migration by weighing the cost of service migration and the improved experience quality of users. However, their solution can be very time-consuming when MDP has a large number of states. Wang et al. [13] modeled the service migration problem as MDP and proved that when users follow a one-dimensional asymmetric random walk model, the best strategy for solving service migration is the threshold strategy, and they proposed an algorithm to find the best threshold. The algorithm proposed by Wang et al. is more effective than the standard solution of MDP. In addition, in [14], Chen et al. studied the service migration problem when users follow the two-dimensional mobile model. Afterwards, Wang et al. [15] proposed a layered migration architecture (base layer, application layer, and instance layer), which can effectively reduce transmission time. Machen et al. [16] proposed a service migration method based on container handoff, which uses a hierarchical storage system to reduce synchronization overhead of the file system, thereby reducing time cost for service migration. Furthermore, Ud Din et al. [17] studied the performance optimization of edge service under the constraint of long-term service migration overhead. To solve the problem of unpredictable user movement behavior, Ouyang et al. adopted the Lyapunov optimization to decompose the long-term optimization problem into a series of real-time optimization problems.

There are also some studies dedicated to reducing the probability and frequency of service migration to achieve the objective of optimizing user QoS. To reduce the delay caused by service migration, Ma et al. [18] proposed a Cloud-Spider architecture combining placement of a virtual machine (VM) replica and VM scheduling to reduce high migration delay caused by VM image transmission through low-bandwidth wide-area network (WAN) links. They used deduplication technology to compensate for the additional storage requirements caused by the placement of replicas, and studied the VM replica placement algorithm. Besides, Ouyang et al. [19] deployed service replicas on MEC servers near users in advance, aiming to minimize the probability of

service migration and the number of service replicas. Yatao et al. [20] proposed an analysis model to compare the costs of service migration and service replica deployment. The model analyzes the impact of user movement mode and service duration on migration and replication costs, respectively. The above several studies deploy service replicas on MEC servers around users in advance to reduce the probability of service migration. However, the user can only request the resources of one MEC server at the same time, and the MEC server that is not accessed by the user but has service replicas will be occupied with storage resources. Since the resources on the MEC server are limited, the backup of useless resources will cause the resource waste of the MEC server.

In terms of reducing transmission delay, controlling transmission power is an effective method because transmission power is closely related to signal quality, interference, and channel capacity. Therefore, Bose et al. [21] proposed a cloud-aware power control method to maximize the result delivery rate under the condition of satisfying the delay requirement. Since the transmission power is also related to signal coverage, for the scenario of two MEC servers in [22], the coverage of MEC servers is controlled by transmission function, and VM migration is used to achieve load balancing, so the average service delay of MEC servers is reduced. Afterwards, Zhang et al. [23] extended the previous work in [24]. In the case of multiple MEC servers, they considered the mobility of users and studied how to maximize the cost-effectiveness, that is, to minimize the number of activated MEC servers, under the condition of meeting the service delay.

In summary, based on the transmission power control technology, this paper is aimed at solving the problem of user QoS degradation caused by user mobility in a mobile edge computing environment, and uses the weight coefficient transformation method and PSO algorithm to solve the multiobjective optimization problem, which can reduce the number of wireless handoffs and service migrations of user equipment and minimize the server energy consumption.

3. Problems and Models

3.1. Problem Description. Here, we consider n MEC servers and m users, which form two sets $E = \{e_1, e_2, \dots, e_n\}$ and $U = \{u_1, u_2, \dots, u_m\}$, respectively. The MEC server creates an isolated virtual machine environment for users. In order to meet the service delay requirements, we assume that the virtual machine is always placed on the MEC server connected by the user wirelessly. With the movement of users, radio network resources are also changing dynamically. So, service migration is always accompanied by wireless handoff. As a result, frequent wireless handoff and service migration will have a great impact on user experience.

Suppose there are three users u_1 , u_2 , and u_3 , and their moving paths are shown in Figure 2. The coverage radius of the MEC server e_1 is configured as r_1 and r_3 , and the coverage radius of e_2 , it is configured as r_2 or r_4 . Then, according to Figure 1, there will be four cases as follows:

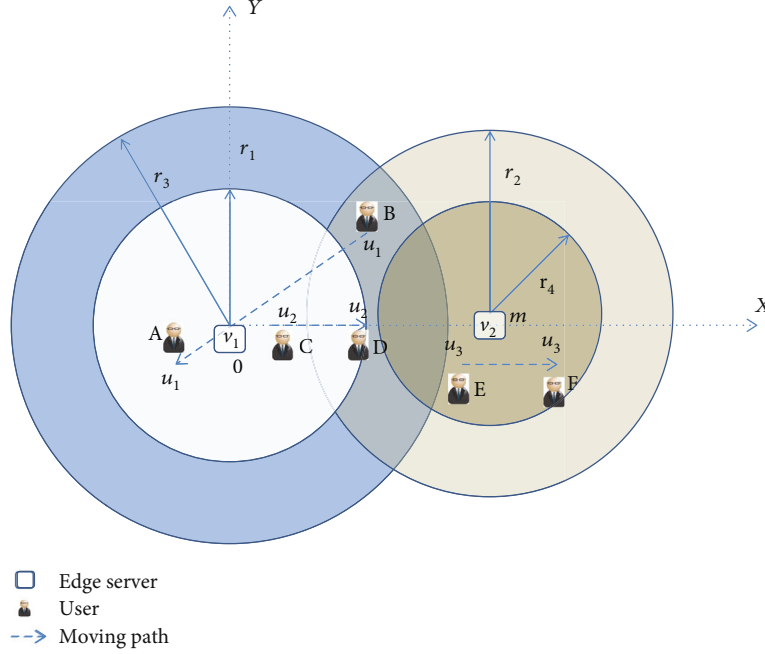


FIGURE 2: The impact of transmission power on service migration.

Case 1. If $r_{e_1} = r_1, r_{e_2} = r_2$, user u_1 leaves the coverage area of e_1 and wireless handoff and service migration need to be performed.

Case 2. If $r_{e_1} = r_1, r_{e_2} = r_4$, user u_1 cannot connect to any MEC server at point B.

Case 3. If $r_{e_1} = r_3, r_{e_2} = r_2$, u_1, u_2 , and u_3 do not require wireless handoff and service migration.

Case 4. If $r_{e_1} = r_3, r_{e_2} = r_4$, as in case 3, all users do not need wireless handoff.

In summary, when the coverage radius is small, users will make frequent wireless handoffs, and even the connection will be interrupted (for case 2). When the coverage area is large, it will cause unnecessary high energy consumption and waste of resources (for cases 3 and 4, handoff can be avoided, and a smaller coverage area can meet the demand). The coverage of the MEC server is related to its transmission power [25]. Therefore, this section will study how to set the transmission power of the MEC server, so as to minimize the number of user wireless handoffs and the energy consumption of the MEC server.

3.2. System Model. The definitions of related concepts are given as follows:

Definition 1 (MEC server). The MEC server can be defined as a two-tuple $e = (p, tp)$, where

- (1) p is the location of the MEC server
- (2) tp is the transmission power of the MEC server

Definition 2 (movement path). The user's movement path can be modeled as a triple $mp = (\text{time}, \text{location}, M)$, where

- (1) *Time* is the length of time that the user moves, which is composed of a series of discrete moments
- (2) *Location* is the location of the user at the above moment
- (3) M is a mapping relationship from time to location:
 $M : \text{time} \rightarrow \text{location}$

In this section, it is assumed that the user is always connected to the MEC server that provides the maximum received signal strength (RSS). Therefore, in this section, the user activity area is divided into an RSS Voronoi diagram. The RSS Voronoi diagram is defined as follows:

Definition 3 (RSS Voronoi diagram). Assuming that there is a group of MEC servers $E = \{e_1, e_2, \dots, e_n\}$ in area A , the RSS Voronoi diagram divides area A into multiple V polygons, each V polygon has an MEC server, and the points in the V polygon are defined as follows:

$$V(e_i) = \{u_k : \text{RSS}_{e_i}^{u_k} > \text{RSS}_{e_j}^{u_k} \forall j \neq i\}, \quad (1)$$

where $\text{RSS}_{e_i}^{u_k}$ is the received signal strength, user u_k is the receiver, and MEC server e_i is the sender.

The basic features of the Voronoi diagram are as follows:

- (1) There is a generator in each subregion

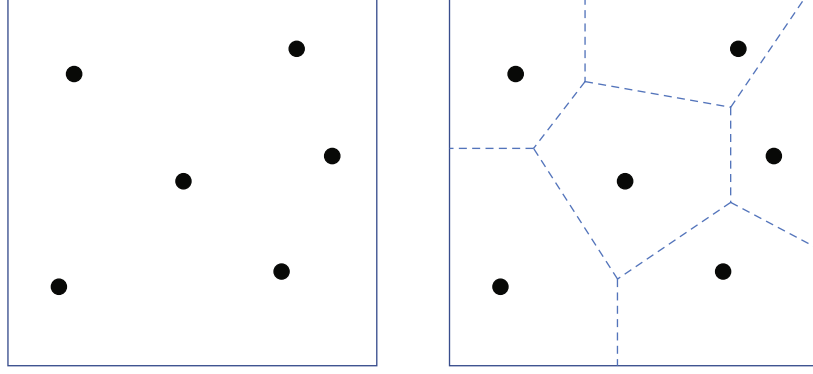


FIGURE 3: Examples of the Voronoi diagram.

- (2) The distance from the point in the subregion to the generator is less than the distances to other generators

Different from the traditional Voronoi diagram, the standard of dividing the subregions is the received signal strength rather than Euclidean distance. Figure 3 is an example of the RSS Voronoi diagram, in which

- (1) There is one MEC server in each subregion
- (2) The signal strength received by the points in the subregion from the MEC server in the subregion is greater than that received from other MEC servers
- (3) The points on the boundary of the subarea receive the same signal strength from the MEC server that generated the boundary

When the user's moving path spans multiple subregions, the user needs to perform multiple wireless handoffs. The division of subregions is related to the user's received signal strength, and the user's received signal strength is related to the transmission power of the MEC server. Therefore, the number of user wireless handoffs and service migrations can be reduced by controlling the transmission power of the MEC server. The relationship between the received signal strength of the user and the transmission power of the MEC server [26, 27] is as follows:

$$\text{RSS}_{e_j}^{u_i} = \text{tp}_{e_j} + G_{e_j} + G_{u_i} + H - L_{e_j}^{u_i}, \quad (2)$$

where tp_{e_j} is the transmission power of the MEC server e_j in decibels (dbm), G_{e_j} and G_{u_i} are the antenna gains of the sender and receiver, respectively, H is the Rayleigh power fading coefficient, and $L_{e_j}^{u_i}$ is the path loss between the sender and the receiver, which is calculated as follows [28–30]:

$$L_{e_j}^{u_i} = L_1 + 10n_1 \log_{10}(d(e_j, u_i)) + 10(n_2 - n_1) \left(1 + \frac{d(e_j, u_i)}{d_b} \right), \quad (3)$$

TABLE 1: The symbols commonly used in this section.

Symbols	Meaning
E	MEC server set
e_i	The i th MEC server in E
RSS	Received signal strength
Tp	Transmission power
G	Antenna gain
H	Rayleigh power fading coefficient
L	Path loss
n_1	Short-distance path loss index
n_2	Long-distance path loss index
d_b	Boundary value of long distance and short distance
EN_j	Energy consumption of MEC server j
Ht_i	Number of handoffs of user u_i

where L_1 is the path loss when the distance between the receiver and the sender is 1 meter; n_1 and n_2 are the long-distance and short-distance path loss indexes, respectively; $d(e_j, u_i)$ is the Euclidean distance between the MEC server e_j and the user u_i , d_b is the boundary value dividing long distance and short distance. The symbols commonly used in this section are shown in Table 1.

Through the mapping $M : \text{time} \rightarrow \text{location}$, the location of the user at any point in time can be derived, and the MEC server that the user is connected to can be obtained. According to the user's movement trajectory, the final number of user's handoffs can be obtained. Here, the user's mobile information can be obtained from many channels; for example, the user may apply map services (such as navigation). In addition, since the user's daily itinerary will not change much, the user's movement path can also be inferred through the user's past behavior. There are many studies on predicting user mobile behavior [31–33], which is beyond the scope of this section. In this section, it is assumed that the user's moving path is known.

As users move, MEC servers that can provide services to users are constantly changing. When the transmission power of the MEC server is low, the coverage area of the MEC server is small, causing frequent user wireless handoffs, and even

signal interruption may occur. If the transmission power is too large, the energy consumption of the MEC server will also increase. Moreover, when the MEC server has a large overlapping coverage area, the interference received by the MEC server will be too large, which in turn increases the service delay. Based on such fact, in this section, the service migration optimization problem is modeled as a multiobjective optimization problem, which is aimed at minimizing the number of user wireless handoffs and minimize the energy consumption of the MEC server.

The energy consumption of the MEC server can be expressed as follows:

$$EN_j = \alpha * tp_j. \quad (4)$$

In formula (4), EN_j is the energy consumption of MEC server e_j , which is proportional to the transmission power tp_j of e_j , and α is an adjustable parameter.

Let $C_i(t_k)$ represent the MEC server that user u_i is connected to at time t_k , and the user is always connected to the MEC server that provides the maximum received signal strength, so $C_i(t_k)$ can be expressed as follows:

$$C_i(t_k) = \left\{ j \mid RSS_{e_j}^{u_i} > RSS_{e_l}^{u_i}, \forall e_l \in E \right\}. \quad (5)$$

Next, the following variables are defined:

$$I_{ij}(t_k) = \begin{cases} 1, & C_i(t_k) = j, \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

$$I_i(t_k) = \begin{cases} 1, & C_i(t_k) \neq C_i(t_{k+1}), \\ 0, & \text{otherwise.} \end{cases}$$

Since users can only connect to the same MEC server at a time, we can have

$$\sum_{e_j \in E} I_{ij}(t_k) = 1. \quad (7)$$

Let Ht_i represent the number of device handoffs during the user's movement, then Ht_i can be expressed as follows:

$$Ht_i = \sum_k I_i(t_k). \quad (8)$$

Therefore, in this paper, the optimization objective can be expressed as follows:

$$\begin{aligned} \mathbf{P} : \quad & \min \sum_{u_i \in U} Ht_i \\ & \min \sum_{e_j \in E} EN_j. \end{aligned} \quad (9)$$

4. The Optimization Method

In this section, two optimization objectives are considered: minimizing the number of wireless handoffs of user equipment and minimizing the energy consumption of the MEC server. This is a multiobjective optimization problem. There are many ways to solve the multiobjective optimization problem. As in reference [34], this paper uses the weight coefficient transformation method to solve the problems raised in this section. Therefore, problem P can be transformed into problem $P1$:

$$P1 : \quad \min \quad w_1 \sum_{u_i \in U} Ht_i + w_2 \sum_{e_j \in E} EN_j, \quad (10)$$

where $w_1 + w_2 = 1$. In this study, the PSO algorithm [34] was used to solve problem $P1$. Because of its simplicity and ease of implementation and the small number of parameters, the particle swarm algorithm is widely used in function optimization, neural network training, etc. In the PSO algorithm, the particle is a bird in the search space, the position of the particle is the solution to the optimization problem, and the speed of the particle determines the direction and distance of its flight. During each iteration, each particle updates its position according to its velocity. After multiple iterations, the optimal solution is finally obtained. Using PSO algorithm to solve the problem $P1$ mainly includes the following steps:

Step 1. Randomly initialize the particle swarm position and velocity matrix: $xMatrix$, $vMatrix$. The position and velocity of each particle are n -dimensional vectors, and the position is composed of the transmission power of n MEC servers. Assuming that the transmission power range of the MEC server is $[tp_{\min}, tp_{\max}]$, the position of particle i is initialized to n random numbers in $[tp_{\min}, tp_{\max}]$, that is, $x_i = (tp_{i1}, tp_{i2}, \dots, tp_{in})$, and the velocity is initialized to n random numbers in $(0, 1)$, that is, $v_i = (v_{i1}, v_{i2}, \dots, v_{in})$.

Step 2. Calculate the fitness of the particles according to formulas (4)–(8), and get the fitness matrix $fitMatrix$

Step 3. Update the historical optimal position of each particle. In the first iteration, the historical optimal position of each particle is its random initial position $p_i = x_i$

Step 4. Update the global optimal position of the group. Initially, the global optimal position of the swarm is the particle position with the smallest fitness in the particle swarm position matrix $xMatrix$ $p_g = \min_{fit}(xMatrix)$

Step 5. Update the speed and position of each particle:

$$v_i^{t+1} = w \cdot v_i^t + c_1 \cdot r_1 \cdot (p_i^t - x_i^t) + c_2 \cdot r_2 \cdot (p_g^t - x_i^t), \quad (11)$$

where w is the inertia weight, c_1 is the local learning factor, c_2 is the global learning factor, r_1, r_2 are random numbers in $[0, 1]$.

```

Input: base station location, user moving path
Output: number of wireless handoffs, energy consumption
Begin
1: //initialization
2: for each particle  $i$  do
3:   Randomly select  $n$  numbers from the interval  $[tp_{\min}, tp_{\max}]$  as the position  $x_i$  of  $i$ 
4:   Randomly select  $n$  numbers from the interval  $(0, 1)$  as the speed  $v_i$  of  $i$ 
5:   Apply formulas (4)–(8) to evaluate particle  $i$ 
6:   The historical optimal position of particle  $ip_i = x_i$ 
7: end for
8: Global optimal position  $p_g = \min_{\text{fit}}\{p_i\}$ 
9: while  $iterations < IteratorNum$  do:
10:  for  $i = 1$  to  $N$  do
11:    Apply formula (11) to update velocity of particle  $i$ 
12:    Apply formula (12) to update position of particle  $i$ 
13:    // Update  $p_i$  and  $p_g$ 
14:    if  $\text{fit}(x_i) < \text{fit}(p_i)$  do
15:       $p_i = x_i$ 
16:    if  $\text{fit}(p_i) < \text{fit}(p_g)$  do
17:       $p_g = p_i$ 
18:    end for
19: end while
End

```

ALGORITHM 1: PSO algorithm to solve problem P1.

The position of the particle is updated to:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (12)$$

Step 6. Repeat (Steps 2–5) until the end condition is met.

The algorithm is described as follows, where *iterations* represents the current number of iterations, *IteratorNum* represents the number of iterations, and *N* is the size of the particle swarm:

5. Experimental Results and Analysis

The experimental environment is PyCharm Community Edition, the programming language is Python 3.5, and the configuration of the experimental machine is 16 G memory, core i7-4790 3.60 GHz processor, Windows 7, 64-bit operating system. The Telecom Dataset is provided by Shanghai Telecom [8–10]. The data set has six parameters including month, day, start time, end time, base station location, and user ID. The data set contains a total of 7.2 million records, which are records of 9,481 mobile phones accessing the Internet through 3,233 base stations. In this experiment, 10 stations of three subways in Shanghai were selected as the user's moving routes, the latitude and longitude of these 30 sites were obtained on Baidu Maps, and 8 base stations were selected near the 30 sites to provide services to users. The experiment parameters are shown in Table 2:

5.1. *Experiment for PSO Parameter Selection.* First, the parameters in the PSO algorithm were verified, including

TABLE 2: Experimental parameters.

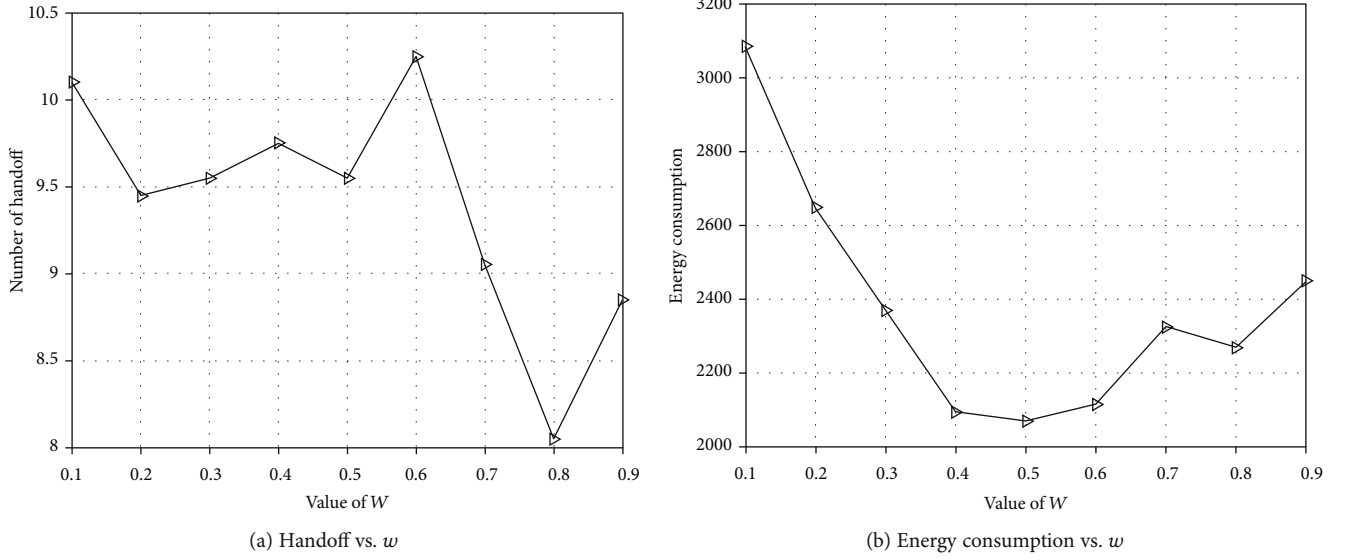
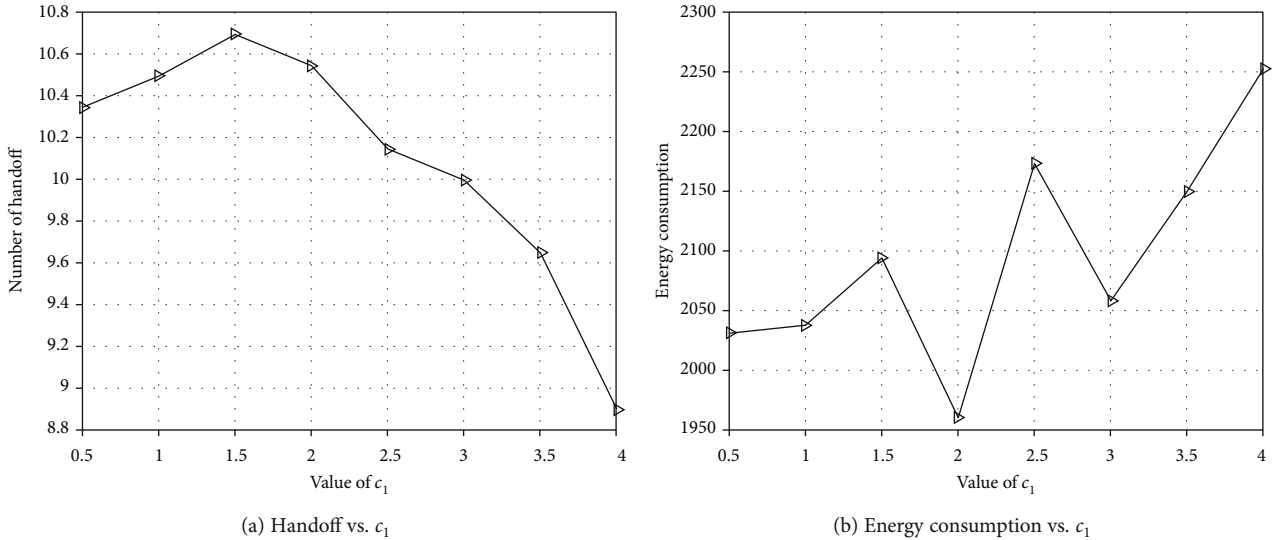
Parameters	Values
L_1	20 dBm
G_{ui}	8.35 dB
G_{ej}	24.5 dB
n_1	2
n_2	4
d_b	100 m
PSO number of particles	10
PSO number of iterations	100

TABLE 3: PSO parameters.

Parameter	w	c_1	c_2
Experiment 1	0.1 to 1	2	2
Experiment 2	0.5	0.5 to 4	2
Experiment 3	0.5	2	0.5 to 4

the impact of inertia weight w , local learning factor c_1 , global learning factor c_2 on the number of handoffs, and energy consumption. The particle swarm size was set to 10, the number of iterations was set to 100, the particle swarm algorithm was executed 20 times, and the average value was taken as the experimental result. The experimental parameter settings are shown in Table 3.

The essence of PSO is to apply formula (11) to update the velocity of particles in each iteration. Formula (11) is

FIGURE 4: The influence of parameter w on experimental results.FIGURE 5: The influence of parameter c_1 on experimental results.

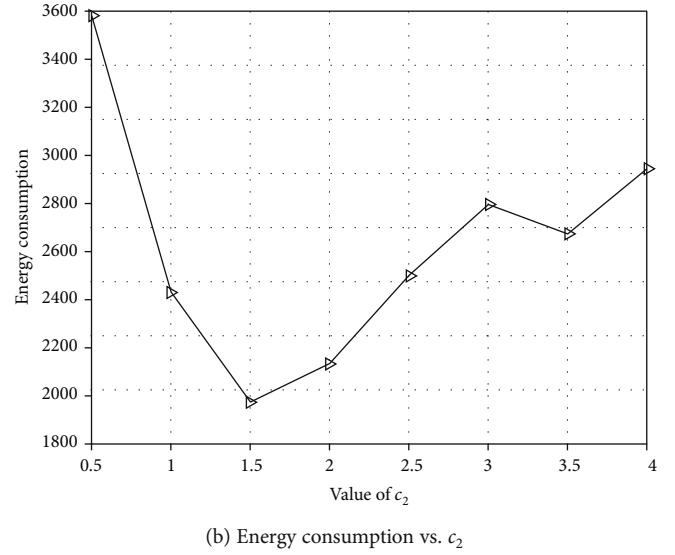
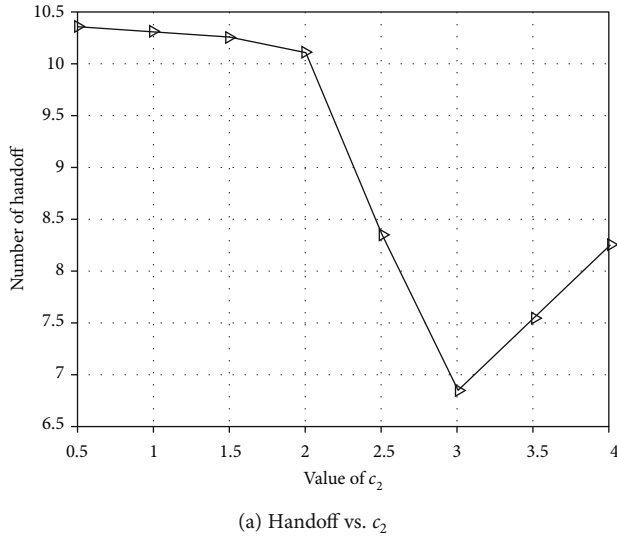
composed of three items: memory item, self-cognition item, and group recognition item. The inertia weight w determines the influence of the speed in the previous iteration on this iteration. To study the influence of the inertia weight w on the experiment, the values of other parameters are fixed: $c_1 = c_2 = 2$, w changes from 0.1 to 0.9, and the step size is 0.1. The experimental results are shown in Figure 4.

The figure above shows that, as the value of w increases, the trend of the number of handoffs is slowly increasing and then decreasing sharply. The minimum value is obtained when $w = 0.8$, and the minimum value is close to 8. The energy consumption drops first and then rises. When the energy consumption is small, the power of the MEC server is small and the coverage area is small, so the number of handoffs is relatively high. Therefore, when $w = 0.5$, the energy consumption achieved the minimum value, and the number of handoffs is 9.55, which is at a relatively high level.

In this study, 20 repeated experiments were performed and the average value was taken as the result, so the number of handoffs may be a decimal.

The local learning factor c_1 and the global learning factor c_2 are the weights of self-cognition items and group recognition items. The purpose of the self-cognition item is that the speed of particles is affected by one's own experience. The group cognition item reflects the influence of knowledge sharing between particles on finding the optimal solution. To study the influence of these two parameters on the experiment, $w = 0.5$ was set and c_1, c_2 were set to range from 0.5 to 4, and the step length was 0.5. The experimental results are shown in Figures 5 and 6.

In Figure 5, the number of user handoffs decreases as the value of c_1 increases, while the energy consumption shows a fluctuating upward trend. The influence of the global learning factor on the number of user handoffs and the energy

FIGURE 6: The influence of parameter c_2 on the experimental results.

consumption of the MEC server is shown in Figure 5. The number of user handoffs and the MEC server energy consumption have a similar trend, and both decrease first and then increase. When $c_2 = 3$, the number of user handoffs achieves the extreme value, and when $c_2 = 1.5$, the energy consumption achieves the optimal value.

In Figure 6, the experimental results show that the two evaluation indicators of number of handoffs and energy consumption cannot reach the optimal value at the same time. In order to balance these two evaluation indicators, the parameters of PSO were set as $w = 0.8$, $c_1 = 3$, and $c_2 = 1$.

5.2. Comparative Analysis of Experiment Results. This experiment was compared with the following two algorithms:

- (1) Genetic algorithm (GA) [24]: the genetic algorithm searches for the optimal solution of the problem by simulating the natural evolution process. The main steps are as follows: (1) initialize the population, (2) assess the individual fitness value, (3) select, (4) crossover, (5) mutate, and (6) repeat (2)–(5) until the end conditions are met
- (2) Simulated annealing algorithm (SA) [25]: the principle of solid annealing is the theoretical basis of the simulated annealing algorithm. A solid is heated to a sufficiently high temperature, and then slowly cooled. During cooling, the particles are gradually ordered, and finally the internal energy is the smallest at room temperature. The basic steps of the simulated annealing algorithm are as follows: (1) initialize the solution T_{old} ; (2) generate a new solution T_{new} ; (3) apply the evaluation function to evaluate T_{old} and T_{new} ; (4) if T_{new} is better than T_{old} , replace T_{old} with T_{new} , otherwise, accept T_{new} with a certain probability; (5) repeat (2)–(4) until the end condition is met. The parameters of the simulated annealing algorithm were set as number of iterations = 100, initial

temperature = 100, attenuation factor = 0.85, and probability of accepting the difference = $\exp(-\Delta t/t)$, where Δt is the difference between T_{new} and T_{old} , and t is the current temperature

Three subway routes in Shanghai were selected as the user's movement trajectory. For each subway line, 10 stations were selected, and 8 base stations were selected around these 30 stations to provide services to users. Then, the influence of the number of sites on the number of handoffs and energy consumption was studied. Figure 7 is the result of the comparison of the three algorithms.

Figure 6(a) shows that, for the three algorithms, the difference in the number of handoffs is small and the results and trends of PSO and SA are very similar. When the number of stations is 20 and 25, the number of handoffs calculated by the two are equal. When the number of stations is less than 20, PSO is slightly better than SA. GA has the worst performance in the number of handoffs, and only when the number of sites is 25 is it better than SA and PSO. The performance of the three algorithms in terms of energy consumption is shown in Figure 6(b). The energy consumption increases as the number of sites increases. Obviously, the results calculated by PSO are optimal in terms of energy consumption. Finally, these three algorithms were also measured from the running time. Although GA performs poorly in the number of handoffs and energy consumption, its execution time cost is the lowest. Although PSO has a slight advantage over SA in the number of handoffs and energy consumption, its execution time cost is much lower than SA. Therefore, PSO can effectively solve this multiobjective optimization problem.

6. Conclusion

Deploying services on edge nodes can bring computing and storage resources closer to users, thereby reducing service delays and improving user experience. However, the dynamic feature of user equipment in mobile edge computing has led to

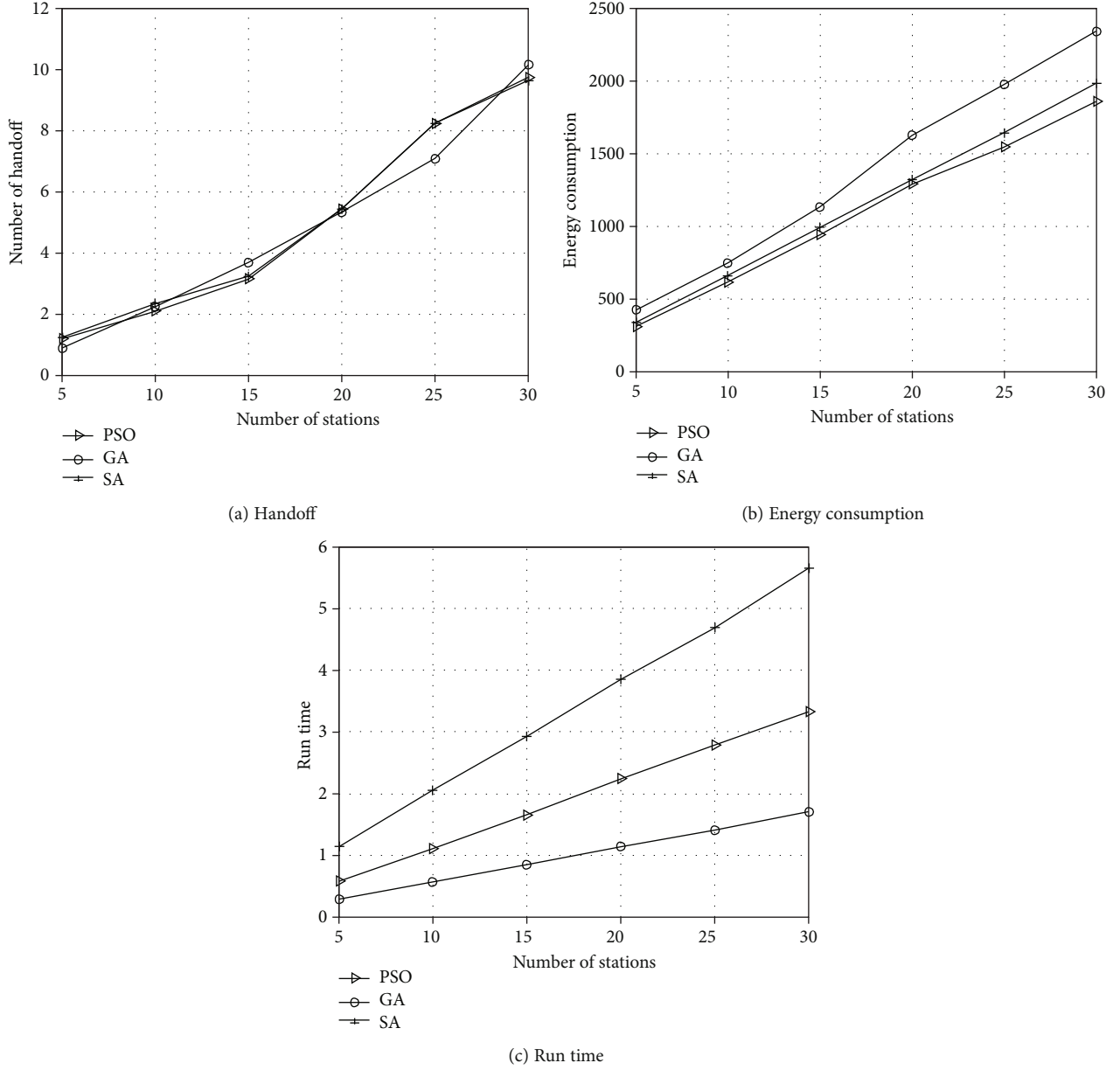


FIGURE 7: The influence of the number of stations on the experimental results.

the need to migrate services between different MEC servers. Since the edge nodes communicate through the backhaul link, the service forwarding between the edge nodes will cause a high delay, which leads to a significant reduction in the quality of user experience. Therefore, reducing service migration is a very important task.

The study considers the use of transmission power control technology to reduce the number of service migrations during user movement. Under the assumption that the user's moving route is known, the edge node of the user's wireless connection is controlled through the transmission power of the MEC server, so the number of wireless handoffs during the user's movement is reduced. Meanwhile, in order to avoid energy waste caused by excessive transmission power, minimizing the energy consumption of the MEC server is also

regarded as an optimization objective. The multiobjective optimization problem is transformed into a single-objective problem through the weight coefficient conversion method, and then the PSO algorithm is used to solve the problem. The experimental results show the effectiveness of the PSO algorithm in this multiobjective optimization problem.

Data Availability

All of the data used in this study are already available on the Internet and is easily accessible.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Talent Project of Anhui Province (Grant No. gxgnfx2019034), the Anhui Province Key Research and Development Program Project (Grant No. 201904a05020091) and its supporting projects (Grant No. PT04a05020094), the Excellent Talents Support Program of Colleges and Universities (Grant No. gxyq2020083), the key project of the Natural Science Research of Higher Education Institutions in Anhui Province (Grant No. KJ2020A0680), the 2018 Higher Education Research Project of Anhui Province (Grant No. 2018JYXM0334), the 2020 Quality Improvement Project of Chaohu College on Discipline Construction (Grant No. kj20xqyx03), and a scientific research project commissioned by the Hefei Saile Education Technology Co., Ltd. (Grant No. hxkt20210002).

References

- [1] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thin-kair: dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *2012 Proceedings IEEE INFOCOM*, pp. 945–953, Orlando, FL, USA, 2012.
- [2] S. Wang, M. Zafer, and K. K. Leung, "Online placement of multi-component applications in edge computing environments," *IEEE Access*, vol. 5, pp. 2514–2533, 2017.
- [3] B. G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clone-cloud: elastic execution between mobile device and cloud," in *Proceedings of the sixth conference on Computer systems*, pp. 301–314, New York, 2011.
- [4] S. Wang, A. Zhou, R. Bao, W. Chou, and S. S. Yau, "Towards green service composition approach in the cloud," *IEEE Transactions on Services Computing*, pp. 1–1, 2018.
- [5] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.
- [6] M. R. Bonyadi and Z. Michalewicz, "Particle swarm optimization for single objective continuous space problems: a review," *Evolutionary Computation*, vol. 25, no. 1, pp. 1–54, 2017.
- [7] Z. Becvar, J. Plachy, and P. Mach, "Path selection using hand-over in mobile networks with cloud-enabled small cells," in *2014 IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)*, pp. 1480–1485, Washington, DC, 2014.
- [8] S. Wang, Y. Zhao, L. Huang, J. Xu, and C. H. Hsu, "QoS prediction for service recommendations in mobile edge computing," *Journal of Parallel and Distributed Computing*, vol. 127, pp. 134–144, 2019.
- [9] Y. Guo, S. Wang, A. Zhou, J. Xu, J. Yuan, and C.-H. Hsu, "User allocation-aware edge cloud placement in mobile edge computing," *Software: Practice and Experience*, vol. 50, no. 5, pp. 489–502, 2020.
- [10] J. Xu, S. Wang, B. K. Bhargava, and F. Yang, "A blockchain-enabled trustless crowd-intelligence ecosystem on mobile edge computing," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3538–3547, 2019.
- [11] T. Taleb and A. Ksentini, "An analytical model for follow me cloud," in *2013 IEEE Global Communications Conference (GLOBECOM)*, pp. 1291–1296, Atlanta, GA, USA, 2013.
- [12] A. Ksentini, T. Taleb, and M. A. Chen, "Markov decision process-based service migration procedure for follow me cloud," in *2014 IEEE International Conference on Communications (ICC)*, pp. 1350–1354, Sydney, NSW, Australia, 2014.
- [13] S. Wang, R. Urgaonkar, T. He, M. Zafer, K. Chan, and K. K. Leung, "Mobility-induced service migration in mobile micro-clouds," in *2014 IEEE Military Communications Conference*, pp. 835–840, Baltimore, MD, USA, 2014.
- [14] C. Chen, Z. Liu, S. Wan, J. Luan, and Q. Pei, "Traffic flow prediction based on deep learning in internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2020.
- [15] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic service migration in mobile edge-clouds," in *2015 IFIP Networking Conference (IFIP Networking)*, pp. 1–9, Toulouse, France, 2015.
- [16] A. Machen, S. Wang, K. K. Leung, B. J. Ko, and T. Salonidis, "Live service migration in mobile edge clouds," *IEEE Wireless Communications*, vol. 25, no. 1, pp. 140–147, 2017.
- [17] F. Ud Din, A. Ahmad, H. Ullah, A. Khan, T. Umer, and S. Wan, "Efficient sizing and placement of distributed generators in cyber-physical power systems," *Journal of Systems Architecture*, vol. 97, pp. 197–207, 2019.
- [18] L. Ma, S. Yi, and Q. Li, "Efficient service migration across edge servers via docker container migration," in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, pp. 1–13, New York, NY, USA, 2017.
- [19] T. Ouyang, Z. Zhou, and X. Chen, "Follow me at the edge: mobility-aware dynamic service placement for mobile edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2333–2345, 2018.
- [20] Y. Yang, Z. Zheng, X. Niu, M. Tang, Y. Lu, and X. Liao, "A location-based factorization machine model for web service QoS prediction," *IEEE Transactions on Services Computing*, pp. 1–1, 2019.
- [21] S. K. Bose, S. Brock, R. Skeoch, and S. Rao, "CloudSpider: combining replication with scheduling for optimizing live migration of virtual machines across wide area networks," in *2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 13–22, Newport Beach, CA, USA, 2011.
- [22] I. Farris, T. Taleb, M. Bagaa, and H. Flick, "Optimizing service replication for mobile delay-sensitive applications in 5G edge network," in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, Paris, France, 2017.
- [23] Y. Zhang, G. Cui, S. Deng, F. Chen, Y. Wang, and Q. He, "Efficient query of quality correlation for service composition," *IEEE Transactions on Services Computing*, 2018.
- [24] P. A. Frangoudis and A. Ksentini, "Service migration versus service replication in multi-access edge computing," in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 124–129, Limassol, Cyprus, 2018.
- [25] P. Mach and Z. Becvar, "Cloud-aware power control for real-time application offloading in mobile edge computing," *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 5, pp. 648–661, 2016.
- [26] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, "Hybrid method for minimizing service delay in edge cloud computing through VM migration and transmission power control," *IEEE Transactions on Computers*, vol. 66, no. 5, pp. 810–819, 2017.
- [27] T. G. Rodrigues, K. Suto, H. Nishiyama, N. Kato, and K. Temma, "Cloudlets activation scheme for scalable mobile

- edge computing with transmission power control and virtual machine migration,” *IEEE Transactions on Computers*, vol. 67, no. 9, pp. 1287–1300, 2018.
- [28] S. Wan, A. Gu, and Q. Ni, “Cognitive computing and wireless communications on the edge for healthcare service robots,” *Computer Communications*, vol. 149, pp. 99–106, 2020.
 - [29] Y. Zhang, C. Yin, Q. Wu, Q. He, and H. Zhu, “Location-aware deep collaborative filtering for service recommendation,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–12, 2020.
 - [30] R. Fernandes and R. D’Souza, “A new approach to predict user mobility using semantic analysis and machine learning,” *Journal of medical systems*, vol. 41, no. 12, 2017.
 - [31] S. Roy, R. Bose, and D. Sarddar, “Fuzzy based dynamic load balancing scheme for efficient edge server selection in cloud-oriented content delivery network using Voronoi diagram,” in *2015 IEEE International Advance Computing Conference (IACC)*, pp. 828–833, Bangalore, India, 2015.
 - [32] W. Su, S. J. Lee, and M. Gerla, “Mobility prediction in wireless networks,” in *MILCOM 2000 Proceedings. 21st Century Military Communications. Architectures and Technologies for Information Superiority (Cat. No.00CH37155)*, vol. 1, pp. 491–495, Los Angeles, CA, USA, 2000.
 - [33] F. Calabrese, G. Di Lorenzo, and C. Ratti, “Human mobility prediction based on individual and collective geographical preferences,” in *13th international IEEE conference on intelligent transportation systems*, pp. 312–317, Funchal, Portugal, 2010.
 - [34] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei, “An edge traffic flow detection scheme based on deep learning in an intelligent transportation system,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2020.