

Research Article

Joint Optimization of Energy Conservation and Migration Cost for Complex Systems in Edge Computing

Xiaolong Xu ^{1,2,3} Yuan Xue,¹ Mengmeng Cui,¹ Yuan Yuan,⁴ and Lianyong Qi ⁵

¹School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, China

²Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing, China

³Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science & Technology, Nanjing, China

⁴Department of Computer Science and Engineering, Michigan State University, MI, USA

⁵School of Information Science and Engineering, Qufu Normal University, Qufu, China

Correspondence should be addressed to Lianyong Qi; lianyongqi@gmail.com

Received 5 May 2019; Revised 25 July 2019; Accepted 19 August 2019; Published 4 December 2019

Guest Editor: Xiaoqing Bai

Copyright © 2019 Xiaolong Xu 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.

By means of the complex systems, multiple renewable energy sources are integrated to provide energy supply for users. Considering that there are massive services needed to process in complex systems, the mobile services are offloaded from mobile devices to edge servers for efficient implementation. In spite of the benefits of complex systems and edge servers, massive resource requirements for implementing the increasing resource requests decrease the execution efficiency and affect the whole resource usage of edge servers. Therefore, it remains an issue to achieve dynamic scheduling of the computing resources across edge servers. With the consideration of this issue, a Balanced Resource Scheduling Method, named BRSM, for trade-offs between virtual machine (VM) migration cost and energy consumption of VM migrations for edge server management, named BRSM, is designed in this paper. Technically, we analyze the load conditions of edge servers and formulate the energy consumption of VM migrations and VM migration cost as a multi-objective optimization problem. Then, we propose a dynamic resource scheduling method for WMAN to deal with the multi-objective optimization problem. In addition, nondominated sorting genetic algorithm III (NSGA-III) is adopted to generate optimal resource scheduling strategies. Finally, we conduct experiment simulations to testify the efficiency of the proposed method BRSM.

1. Introduction

In recent years, fossil fuels are adopted in most energy systems, which generate a lot of harmful substances and pollute the environment [1, 2]. In order to save fuel and reduce pollution to the environment, renewable energy gradually replaces the fossil energy. Nevertheless, renewable energy is intermittent and difficult to control. For example, provided that the wind stops, wind power generation is difficult to continue. With the intermittent nature of renewable energy, one type of renewable energy cannot meet the balance between supply and demand of current energy systems [3]. Therefore, it is of urgency to develop complex systems that integrate multiple renewable energy sources. Complex systems are systems of intelligent, adaptive subjects with a medium number of actions based on local information, which are defined by attributes like feedback and adaptation. Considering the intermittent nature of renewable energy, a large number of mobile devices are used in complex systems. The mobile devices are arranged to collect

real-time environmental parameters to provide reliable references for strategy formulation.

Although the environmental information is valuable, in view of the huge volume of data in complex systems, the resource consumption of mobile devices rises sharply [4]. Constrained by numerous factors such as limited battery life and limited capacity, mobile devices gradually fail to meet the requirements of mobile services and application execution is of less efficiency in local mobile devices. Therefore, it is suitable to offload the computing applications or services to the cloud platform, considering that the resources in the cloud are accessed conveniently [5]. In the cloud platform, physical resources are provided in the form of multiple virtual machines (VMs) and each mobile device corresponds to a cloud clone which is deployed on a VM. Users are able to access the configurable computing resources shared pool in the cloud and use the required cloud clone to perform the offloaded mobile services [6].

Nevertheless, the transmission delay is unneglectable in terms of the long distance between mobile devices and the

remote cloud. Besides, transmitting mobile services to the cloud consumes massive energy in complex systems, violating the concept of green cloud computing [7]. Edge server, which is co-located with access point (AP), is proposed to be a substitute for the cloud. Mobile users access computing resources of edge servers through APs. Compared to the cloud, edge servers are in close proximity with mobile devices, reducing the data transmission delay [8]. Furthermore, the virtualization technology is used in edge computing as well. By means of the virtualization technology, hardware resources in edge servers are decoupled from software and the same type of VM can run in different physical devices [9]. Consequently, mobile software is enabled to execute on multiple hardware, increasing the utilization of hardware resources as well as achieving the energy conservation.

Despite the advantages, the adoption of edge servers in complex systems has several shortcomings. In fact, compared to remote cloud clusters, the computing resources of edge servers are limited. In densely populated areas, as the number of mobile devices increases dramatically, there is a high possibility that a mass of users intends to offload the computation-intensive services or applications to edge servers nearby. Hence, the edge servers need to response the resource requirements from a large number of mobile devices. As edge servers cannot provision sufficient computing resources for abundant services simultaneously, the computing resources of edge servers run out, resulting in the rejection of new service requests [10, 11]. The applications whose requests are rejected are queued in the edge server until the previous application is completed and the needed resources are available again. Especially, provided that the waiting applications are latency-sensitive, unexpected consequences take place and the quality of experience (QoE) for users greatly drops off. Thus, it is necessary to offload the queued service quests to the edge servers with unoccupied computing resources through VM migration.

Apart from the waiting time, energy consumption is a key criterion for resource management in complex systems. With the explosive growth of consumed energy for information and communication technology (ICT), numerous greenhouse gases are emitted into the atmosphere [12]. Thus, energy conservation plays a significant role in the implementation of future wireless communication. Considering that edge computing is of great use in wireless communication, the energy consumption of edge computing is a research hotspot for better communication modes [13]. In complex systems, the energy consumption of computation offloading includes two parts. The first part is the energy used for information transmission across edge servers. With the virtualization technology, the transmission is in the form of VM migrations. That is to say, the first part of energy consumption is the energy consumption of VM migrations. Moreover, the other part is the energy consumption of infrastructures, including the energy caused by data processing in edge servers and the energy caused by sending and receiving data in APs. As the configuration of each edge server is the same and, the energy consumption of data processing is the same for a service. With the consideration of time consumption and energy consumption, it is of utmost significance to develop a strategy to achieve

TABLE 1: Key terms and descriptions.

Terms	Descriptions
N	The number of edge servers
C	The set of edge servers, $C = \{c_1, c_2, \dots, c_N\}$
c_n	The n -th edge server in C
φ_n	The capacity of the n -th edge server c_n
Q	The number of APs
W	The set of APs, $W = \{w_1, w_2, \dots, w_Q\}$
w_q	The q -th AP in W
Z	The number of mobile services
S	The set of mobile services, $S = \{s_1, s_2, \dots, s_Z\}$
s_z	The n -th mobile service in S
os_z	The occupied start time of s_z
od_z	The duration time of s_z
nv_z	The amount of VMs of s_z
pv_z	The packaged VM to perform s_z

the joint optimization of migration cost and energy consumption.

The main contributions of this paper are summarized as follows:

- (i) We present the basic concepts and definitions to analyze the VM migration cost and the energy consumption of VM migrations in complex systems.
- (ii) We design a Balanced Resource Scheduling Method for trade-offs between VM migration cost and energy consumption of VM migrations for edge server management, named BRSM, to dynamically provision resource management in complex systems by live VM migration technique.
- (iii) We adopt nondominated sorting genetic algorithm III (NSGA-III) [14] to bring about the multi-objective optimization. Multi-criteria decision-making (MCDM) and simple additive weighting (SAW) are employed to select the optimal VM migration strategy.
- (iv) We conduct simulations to verify the efficiency of our proposed method BRSM.

The remainder of this paper is organized as follows. Section 2 introduces the basic concepts and the completed modeling as well as formulation. Section 3 elaborates the proposed dynamic resource management method. Section 4 shows the comparison analysis of the simulation experiments. Section 5 reviews the related work. Finally, Section 6 outlines the conclusions and future work.

2. System Model and Problem Formulation

In this section, basic concepts and definitions for complex systems in edge computing are introduced. Besides, VM migration cost and energy consumption of VM migrations are also analyzed. Key terms and the descriptions are listed in Table 1.

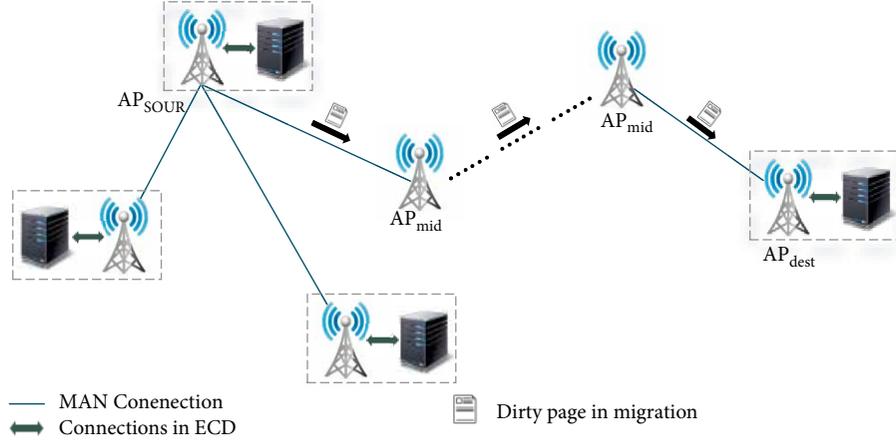


FIGURE 1: An instance of VM migration in ECD-based complex systems.

2.1. Basic Concepts and Definitions. In complex systems, the experience of mobile users are promoted continuously and edge servers provide computing services which are in close proximity to mobile terminals, increasing the QoE for mobile users. There is an AP which is located with an edge server, forming an edge computing device (ECD). Mobile users are able to access the resources in the edge server through the co-located AP. Assume that there are N edge servers in complex systems, which are denoted as $C = \{c_1, c_2, \dots, c_N\}$.

Due to the loss of generality, there is a physical machine (PM) in each edge server to provide mobile services. As the virtualized technique plays a critical role in the resource management for infrastructure, it is appropriate to adopt the virtualized technique in complex systems. The physical resources in an edge server are provided in the form of virtual machines (VMs). By means of the virtualized technique, the physical resources in the edge server are abstracted as several resource units, which are known as VM instances, to accommodate mobile services. Aiming to measure the capacity of each edge server, let φ_n be the number of VM instances in the n -th ($n = \{1, 2, \dots, N\}$) edge server c_n . Considering that there are separate access points, we assume that there are Q access points, denoted as $W = \{w_1, w_2, \dots, w_Q\}$, and the q -th ($q = \{1, 2, \dots, Q\}$) access point is denoted as w_q .

The mobile service set offloaded to VMs is denoted as $S = \{s_1, s_2, \dots, s_Z\}$ and Z represents the number of mobile services. For resource provisioning, the mobile service s_z ($z = \{1, 2, \dots, Z\}$) in S requires multiple VMs which are in the requirements of desired execution time.

Definition 1 (VM occupation requirement of s_z). A 3-tuple, which is denoted as $r_z = (nv_z, os_z, od_z)$, is used to define the VM occupation requirement of s_z , nv_z , os_z and od_z are the number of VMs of s_z , the occupied start time, the duration time respectively.

To avoid the unnecessary transmission delay, the VMs which host the same mobile service are deployed in the same edge server, decreasing the communication cost.

Definition 2 (packaged VM for s_z). Let pv_z represent the packaged VM for the mobile service s_z . The packaged VM pv_z contains all VMs to host the mobile service s_z , whose number is nv_z .

Figure 1 presents an instance of mobile service offloading in the form of VM migration. The AP which is deployed next to the source edge server is named as AP_{sour} and the AP deployed next to the destination edge server is named as AP_{dest} . There are several APs between the AP_{sour} and AP_{dest} , each of which is named as AP_{mid} . In a VM migration strategy, the VM image and the dirty pages caused by migrations are transmitted from the AP_{sour} to the AP_{dest} across AP_{mid} s.

2.2. Migration Cost Analysis of VM Migrations. Considering that VM migration takes a lot of time and it is not always possible to be done, it is necessary to calculate the cost caused by the VM migrations during the whole execution period $[T_s, T_e]$. The migration cost, which also represents VM downtime, consists of the access time of the log file and the switch time of the VMs [18]. Suppose the transmission times of the memory image is $H_z(t)$ during the migration process of pv_z from c_n to $c_{n'}$.

At the time instant t , $F_z^n(t)$ is a binary variable to judge whether pv_z is deployed on c_n , which is defined by

$$F_z^n(t) = \begin{cases} 1, & \text{if } pv_z \text{ is deployed on } c_n, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

$P_z^{n,n'}(t)$ is a binary variable to judge whether pv_z is migrated from c_n to $c_{n'}$, which is defined by

$$P_z^{n,n'}(t) = \begin{cases} 1, & \text{if } pv_z \text{ is migrated from } c_n \text{ to } c_{n'}, \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

When the dirty page transfers at $h_z(t)$ time, let $AT_z^{h_z}(t)$ be the access time of all log files which contains $T_{edge}^{h_z}(t)$ and $T_{mid}^{h_z}(t)$ [18]. When pv_z needs to be migrated between the edge server and the AP which is located next to the edge server, the time consumption is defined as

$$T_{edge}^{h_z}(t) = \sum_{z=1}^Z \sum_{n'=1}^N F_z^n(t) \cdot P_z^{n,n'}(t) \cdot \frac{S_z^{h_z}(t)}{\phi}, \quad (3)$$

where $S_z^{h_z}(t)$ is the size of dirty page produced by the migration of pv_z at $h_z(t)$ time, and ϕ is the bandwidth between the edge server and its corresponding AP.

Suppose $I_{n,n'}$ is the total number of APs between c_n and $c_{n'}$, including a AP_{sour} , a AP_{dest} and several AP_{mid} s. The AP_{sour} is located next to c_n and the AP_{dest} is located next to $c_{n'}$. Thus, the time consumption caused by the migration of pv_z from AP_{sour} to AP_{dest} is calculated by

$$T_{mid}^{h_z}(t) = \sum_{z=1}^Z \sum_{n'=1}^N F_z^n(t) \cdot P_z^{n,n'}(t) \cdot \frac{S_z^{h_z}(t)}{\lambda} \cdot (I_{n,n'} - 1), \quad (4)$$

where λ is the bandwidth between APs.

The process of VM migration is divided into multiple rounds. In the first round, the VM image would be sent to the destination PM while after that the dirty page produced in the previous round is sent to the destination PM in each subsequent round. Therefore, the size of dirty page is calculated by

$$S_z^{h_z}(t) = \begin{cases} M_z(t), & \text{if } h = 0, \\ R_z(t) \cdot (2T_{edge}^{h_z-1}(t) + T_{mid}^{h_z-1}(t)), & \text{otherwise,} \end{cases} \quad (5)$$

where $M_z(t)$ refers to the size of the image memory of pv_z and $R_z(t)$ is the producing rate of the dirty page.

The switch time of pv_z in the migration operation is calculated by

$$OT_z(t) = \sum_{z=1}^Z \sum_{n'=1}^N F_z^n(t) \cdot P_z^{n,n'}(t) \cdot 2\zeta_z(t), \quad (6)$$

where $\zeta_z(t)$ represents the switch time of the packaged VM pv_z .

Definition 3 (migration cost of a packaged VM inc_n). The migration cost represents the time consumption of a VM migration. When pv_z is migrated from c_n to $c_{n'}$, the migration cost is calculated by

$$d_n(t) = 2T_{edge}^{h_z}(t) + T_{mid}^{h_z}(t) + OT_z(t). \quad (7)$$

The average migration cost is calculated by

$$d(t) = \frac{1}{Z} \sum_{n=1}^N (2T_{edge}^{h_z}(t) + T_{mid}^{h_z}(t) + OT_z(t)). \quad (8)$$

During the whole execution period $[T_s, T_e]$, the migration cost is calculated by

$$D = \frac{1}{T_e - T_s} \int_{T_s}^{T_e} d(t) dt. \quad (9)$$

2.3. Energy Consumption Analysis of VM Migrations. Apart from the VM migration cost, we consider the energy consumption in complex systems during the whole execution period $[T_s, T_e]$. Generally, we calculate the energy consumption of edge servers and APs. With the adoption of virtualized technique, the energy consumption of edge servers mainly includes the energy consumption of VMs in edge servers. Considering that each edge server configuration is the same, the energy consumption

```

Input:  $A, t, c_n$ 
Output:  $is_{t,n}$ 
1:  $is_{t,n} = \varphi_n$ 
2: for  $m = 1$  to  $M$  do
3: if  $ac_m = c_n$  then
4: if  $t \geq as_m$  then
5:  $ft_z = as_m + ad_m$ 
6: if  $t < ft_z$  then
7:  $is_{t,n} = is_{t,n} - 1$ 
8: end if
9: end if
10: end if
11: end for
12: Return  $is_{t,n}$ 

```

ALGORITHM 1: Edge server idle space acquisition

of data processing is the same for a service whatever the offloading strategy is. Therefore, the energy consumption of data processing is not calculated in this paper.

At the time instant t , the energy consumption of running VMs in edge servers is calculated by

$$E_{active}(t) = \sum_{n=1}^N \sum_{z=1}^Z F_z^n(t) \cdot nv_z \cdot od_z \cdot \alpha_z, \quad (10)$$

where α_z is the energy consumption rate of each running VM.

The edge server keeps running until all VMs in this edge server have finished the hosted tasks. Therefore, the running time of the n -th edge server c_n is calculated by

$$t_n(t) = \max_{z=1}^Z \{F_z^n(t) \cdot od_z\}. \quad (11)$$

The energy consumption of idle VMs in edge servers is calculated by

$$E_{idle}(t) = \sum_{n=1}^N \sum_{z=1}^Z F_z^n(t) \cdot nv_z \cdot (t_n(t) - od_z) \cdot \beta_z + (\varphi_n - F_z^n(t) \cdot nv_z) \cdot t_n(t) \cdot \beta_z, \quad (12)$$

where β_z is the energy consumption rate of VMs in idle mode.

The basic energy consumption of edge servers is calculated by

$$E_{base}(t) = \sum_{n=1}^N \gamma_n \cdot t_n(t), \quad (13)$$

where γ_n is the basic energy consumption rate of c_n .

During the VM migrations, APs frequently receive and send data, contributing to the energy consumption in complex systems. The energy consumption rate of w_q is calculated by

$$\rho_q(t) = b_q + tm_q \cdot tc_q \cdot \sum_{p \in W} \frac{td_q}{lr_{pq}}, \quad (14)$$

where $b_q, tm_q, tc_q, td_q, lr_q$ respectively represents the baseline power of w_q , the signal transmission power factor of w_q , the signal transceiver power factor of w_q , the traffic demand of w_q and the link rate between w_q and w_p .

Based on the energy consumption rate of APs, the energy consumption of APs is calculated by

$$E_{AP}(t) = \sum_{q=1}^Q \rho_q(t). \quad (15)$$

Besides, the energy consumption of the switch operation of VMs is calculated by

$$E_{switch}(t) = \sum_{z=1}^Z \sum_{n=1}^N \sum_{n'=1}^N 2\xi_z(t) \cdot F_z^n(t) \cdot P_z^{n,n'}(t) \cdot \alpha_z. \quad (16)$$

Definition 4 (energy consumption of VM migrations). The energy consumption of VM represents the consumed amount of energy after a VM migration. The total energy consumption is calculated by

$$E_{all}(t) = E_{active}(t) + E_{idle}(t) + E_{base}(t) + E_{AP}(t) + E_{switch}(t). \quad (17)$$

During the whole execution period $[T_s, T_e]$, the energy consumption is calculated by

$$E = \frac{1}{T_e - T_s} \int_{T_s}^{T_e} E_{all}(t) dt. \quad (18)$$

2.4. Problem Definition. From the foregoing, the migration cost of VMs and the energy consumption of VM migrations are analyzed and quantified. In this paper, we aim to achieve the goal of minimizing the migration cost presented in (9) and reducing the energy consumption presented in (18) while meeting the capacity constraints of ECDs. The problem is formulated by

$$\min D, \min E. \quad (19)$$

$$s.t. \sum_{z=1}^Z pv_z \cdot nv_z \cdot F_z^n(t) \leq \varphi_n. \quad (20)$$

3. BRSM: A Balanced Resource Scheduling Method

In this section, the occupation conditions of edge servers are detected and updated during the dynamic resource scheduling process. Then NSGA-III is adopted to generate resource scheduling strategies to select. Finally, SAW and MCDM are employed to evaluate the resource scheduling strategies and select the appropriate resource scheduling strategy.

3.1. Edge Server Status Detecting and Updating. As the formulation of the resource scheduling strategies largely depends on the load of edge servers, it is necessary to detect the resource usage of each edge server. In this section, the analysis of the allocation record is utilized to preserve the resource usage of edge servers. Assume that there are M allocation records during the execution period, which are denoted as $A = \{a_1, a_2, \dots, a_M\}$.

```

Input:  $t, s_m, dc, dv$ 
Output:  $A$ 
1:  $x = -1, ft = 0$ 
2: for  $m = 1$  to  $|A|$  do
// Migrating a VM to a destination VM
3: if  $ae_m == s_z$  then
4: if  $t > as_m$  &&  $t < as_m + ad_m$  then
5:  $x = m, ft = as_m + ad_m$ 
6:  $ad_m = t - as_m$ 
7: Update  $a_m$  according with  $ad_m$ 
8: end if
9: end if
10: end for
11:  $M = |A| + 1$ 
12:  $ae_M = ae_x, ac_M = dc, av_M = dv, as_M = t, ad_M = ft - t$ 
13: Generate the record  $a_M$ 
14: Add  $a_M$  to  $A$ 
15: Return  $A$ 

```

ALGORITHM 2: Allocation record updating.

Definition 5 (allocation record a_m). The m -th ($m = \{1, 2, \dots, M\}$) allocation record, which is denoted as $a_m = (ae_m, ac_m, av_m, as_m, ad_m)$, preserves the distribution for the packaged VM. In the m -th allocation record, ae_m, ac_m, av_m, as_m and ad_m represent the allocated mobile service, the occupied edge server, the employed VM, the requested start time and the duration time respectively.

According to the number of VM instances for a mobile service, the same number of allocation records are generated. By means of the allocation records, the idle space of each edge server is presented dynamically, considering that the allocation record combines the elapsed time of an edge server.

Algorithm 1 specifies the key idea for edge server idle space acquisition. Through identifying the occupied VM instances, the key idea is to acquire the number of idle VM instances. We input the allocation record collection A , the scheduling instant t and the edge server c_n . The output is the spare space $is_{t,n}$ for the edge server c_n at t .

On the other hand, since the mobile services are offloaded almost continuously, the load distribution of the edge servers changes dynamically. Therefore, migrating the VMs from the overload edge servers to the underload edge servers is urgent.

Due to the real holding time of VMs, when a VM is migrated from one edge server to another, the original allocation record needs to be updated and revised. Moreover, the migration leads to a generation of new allocation records for the referred VM instance. In this new allocation record, the migration time is the start occupation time while the duration time is determined by the rest hosting time of the mobile service.

In Algorithm 2, the allocation record updating is specified. We input the scheduling instant t , the processed objective edge server s_m , the destination edge server dc and the destination VM dv . The output is the updated allocation record

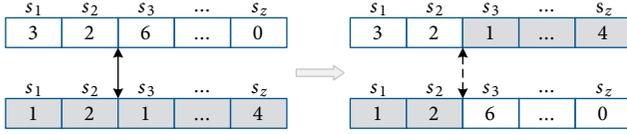


FIGURE 2: An instance of crossover operation.

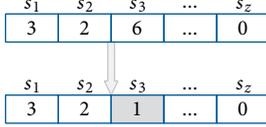


FIGURE 3: An instance of mutation operation.

collection A . At the migration instant t , the allocation record is updated for the objective mobile service s_z (Lines 1–10). Besides, a new allocation record with the VM instance d_v is produced at the destination edge server d_c and the new allocation record is added to the allocation record collection A (Lines 11–16).

3.2. Resource Scheduling Method Using NSGA-III. As the VM instances are migrated from the overload edge server to the underload edge server, the resource scheduling problem is defined as a multi-objective optimization problem to minimize the VM migration cost and the energy consumption of VM migrations. Compared with traditional genetic algorithms, NSGA-III reduces the complexity of the noninferior sorting genetic algorithm and expands the sampling space, thus greatly improving the convergence speed [14]. Considering that NSGA-III solves multi-objective optimization problem efficiently, NSGA-III is employed to solve the optimization problem.

3.2.1. Encoding. In this subsection, each packaged VM has a scheduling strategy and we encode for the VM scheduling strategies. In the genetic algorithm (GA), a gene represents a migration strategy of a packaged VM and a chromosome, which is composed of a set of genes, represents a hybrid migration strategy of VMs in the same schedule. The destination edge server of VM migrations is encoded, depending on the edge server set $C = \{c_1, c_2, \dots, c_N\}$.

3.2.2. Fitness Functions and Constraints. A chromosome represents the offloading strategies of all packaged VMs which are in the same schedule. In the population, each chromosome confirms a solution in the aspect of the resource allocation optimization problem. Therefore, fitness functions emerge as standards to measure the superiority of possible solutions. In this paper, the fitness functions include two categories: the VM migration cost and the energy consumption of VM migrations, presented respectively in (9) and (18). As shown in (19) and (20), the design intent of the method is to select an appropriate resource allocation strategy for trade-offs between the two fitness functions as well as satisfying the potential constraints. With efficiency and effectiveness, NSGA-III is

adopted to solve the optimization problem while satisfying the constraints simultaneously.

3.2.3. Initialization. In the initialization, the parameters of GA are firstly determined, including the population size H , the maximum times of iteration I , the probability of crossover P_c and the probability of mutation P_m .

In GA, each chromosome represents the computation offloading strategies of the services or VMs in the same schedule. Besides, the offloading strategy of the z -th mobile service in the h -th chromosome is denoted as $c_{h,z}$ and the h -th chromosome is represented as $C_{h,i} = (c_{h,1}, c_{h,2}, \dots, c_{h,z}) (h = 1, 2, \dots, H)$.

3.2.4. Crossover and Mutation. The crossover operation, which combines two chromosomes, is the single-point crossover in this paper. Figure 2 shows the crossover operation for two chromosomes in one schedule. In this instance, we select a crossover point and swap the genes which are around the selected crossover point. In particular, the crossover operation starts at s_3 and the swap creates two new chromosomes.

Aiming to enhance the fitness values, the mutation operation is taken to modify genes to create new chromosomes. Each gene has the same probability of being modified. Figure 3 illustrates an instance of the mutation operation. In this instance, the code of s_3 is mutated from 6 to 1.

3.2.5. Selection Operation for the Next Generation. For better solutions, the individuals are selected to generate the next population in this subsection. After the operation of crossover and mutation, the size of population is $2H$. The migration cost and the energy consumption of each scheduling strategy are calculated respectively to judge whether the scheduling strategy is optimal. According to fitness values, the scheduling strategies, which are also known as solutions, are sorted by the usual domination principle. Therefore, the nondominated fronts are successfully generated.

As the preparation is completed, the selection operation of chromosomes is conducted. Each time one solution is selected from the highest nondominated front until there are H selected solutions. We define that the last added solution is in the u -th nondominated front. Provided that all of the solutions in the u -th nondominated front are selected, the selection operation finishes and the H selected solutions form the next generation. This kind of selection is called primary selection.

Whereas in reality, all solutions in the u -th nondominated front are not selected. As the solutions in the u -th nondominated front are not all included, further selection needs to be conducted. Consider the number of selected solutions in the u -th nondominated front is w . We need a series of operations to ensure that the selected w solutions are better. This kind of selection is called further selection.

First, we normalize the $2Z$ fitness values of individuals in the population. We search the minimum values of the VM migration cost and the energy consumption, which are denoted as D_m and E_m . Thus, the fitness values for the solutions are updated as

$$D' = D - D_m, \quad (21)$$

$$E' = E - E_m. \quad (22)$$

Input: P_t, R
Output: P_{t+1}
1: $Q_t = \text{Crossover and mutation}(P_t)$
2: $R_t = P_t \cup Q_t$
3: for the solutions in R_t **do**
4: Calculate B by formula (8)
5: Calculate D by formula (16)
6: end for
7: for the $2H$ solutions in R_t **do**
8: Non-dominant sorting the solutions
9: end for
10: Do primary selection
11: if partial solutions in the u -th front are included **do**
12: Do further selection
13: Normalize solutions by formulas (19)–(24)
14: Generate the reference points
15: Associate solutions with the closest reference points
16: Do the selection of w solutions
17: end if
18: return P_{t+1}

ALGORITHM 3: Selecting using NSGA-III.

Based on the fitness values, the extreme values of the VM migration cost and the energy consumption of VM migrations, which are denoted as EV_D and EV_E , are calculated by

$$EV_D = \max \frac{D'}{W_D}, \quad (23)$$

$$EV_E = \max \frac{E'}{W_E}, \quad (24)$$

where W_B and W_D is the weight vector of the two fitness functions respectively.

We use each fitness function as a measure of an axis. In the hyperplane, the intercept of each axis is calculated respectively. The $2Z$ fitness values of individuals in the population are normalized as

$$D'' = \frac{D'}{EV_D}, \quad (25)$$

$$E'' = \frac{E'}{EV_E}. \quad (26)$$

After the normalization process, the two fitness values are put in the domain $[0, 1)$. Then the distributed reference points are connected with selected solutions, which improves the diversity of solutions. With the normalization, the intercept of each axis is 1 and each axis is divided into k parts. Besides, the number of the reference points is denoted as χ . To ensure that every solution is associated with a reference point, χ is

approximately equal to the number of solutions H , which is calculated by

$$\chi = \binom{2Z + k - 1}{k}. \quad (27)$$

According to the number of reference points that each solution associates with, the solutions in the u -th nondominated front are sorted and one of the solutions is randomly selected. This process repeats until w solutions are selected.

Algorithm 3 represents the selection operation for the next generation. In this algorithm, we input the t -th generation population P_t and a set of reference point R while the output is the $(t + 1)$ -th generation population P_{t+1} . In each iteration, the fitness values of every solution are calculated (Lines 1–4). With the operation of crossover and mutation, there are $2H$ solutions to be chosen (Lines 5–6). Then the $2H$ solutions are sorted according to the nondominated principle (Lines 7–9). Finally, by means of the reference points, H solutions are selected as the next generation (Lines 10–17). The process of Algorithm3 would repeat until the maximum iteration.

3.3. Scheduling Strategy Based on SAW and MCDM. In a population, there are H chromosomes which are also called solutions. Each chromosome represents a feasible hybrid scheduling strategy. For the chromosomes, our designed method is to achieve the trade-offs between reducing the VM migration cost as well as optimizing the energy consumption of VM migrations. For the generated solutions, SAW and MCDM are employed to select the relatively optimal solutions.

As the criteria are divided into positive criteria and negative criteria, the migration cost is a negative criterion, which means that the lower the migration cost is, the better the solution is. Similarly, the energy consumption is also a negative criterion.

The migration cost is normalized as

$$V(D) = \begin{cases} \frac{D^{\max} - D}{D^{\max} - D^{\min}}, & D^{\max} - D^{\min} \neq 0, \\ 1, & D^{\max} - D^{\min} = 0, \end{cases} \quad (28)$$

where D^{\max} and D^{\min} represent the maximum and minimum of the migration cost in the population respectively. Moreover, the energy consumption is normalized as

$$V(E) = \begin{cases} \frac{E^{\max} - E}{E^{\max} - E^{\min}}, & E^{\max} - E^{\min} \neq 0, \\ 1, & E^{\max} - E^{\min} = 0, \end{cases} \quad (29)$$

where E^{\max} and E^{\min} represent the maximum and minimum of the energy consumption in the population respectively.

There is no feasible way how to minimize both latency and energy at the same time, it is rather a trade-off between these two objectives. Consequently, each of the two objectives is given a weight. Through different values of the two weights, the impact of each objective on the offloading strategy changes accordingly and the trade-off can be realized. Let μ_1, μ_2 be the weight of the migration cost and the energy consumption respectively. For the comprehensive utility values evaluation of the solutions, the optimization of the VM migration cost

and energy consumption of VM migrations are both taken into consideration. Since the two objectives have the same degree of influence on the selection of the offloading strategy, the weights of the fitness function are set equal in this paper. Therefore, the utility value of the h -th solution is calculated by

$$V(C_h) = \mu_1 V(D) + \mu_2 V(E) \quad (\mu_1 = \mu_2 = 0.5). \quad (30)$$

Based on the utility value of the h -th solution, the optimal solution, which is denoted as $V(C)$, is calculated by

$$V(C) = \max_{h=1}^H V(C_h) \quad (1 \leq h \leq H). \quad (31)$$

At this point, the optimal solutions with maximum utility value have been selected from the population.

3.4. Method Overview. In this paper, we aim to minimize the migration cost and the energy consumption of each service offloading strategy. The resource scheduling problem is quantified as a multi-objective optimization problem. NSGA-III is utilized to generate the resource scheduling strategies. First, we specify the real-time conditions of edge servers and build the resource allocation record collection. Then the randomly generated resource scheduling strategies are encoded. Besides, the fitness functions and potential constraints are proposed to judge each solution. In order to generate new solutions, the operation of crossover and mutation are taken, leading to a total of $2H$ solutions. For the next generation, the usual domination principle and the reference points are used to select the relatively appropriate H solutions. Finally, in spite of SAW and MCDM, the optimal solutions are selected from the population.

Algorithm 4 shows the overview of the proposed method BRSM. The input includes the initialized population X , the allocation record set A and the maximum iteration times I . The output is the new allocation record set A . In this algorithm, we first update the running mobile service collection S and obtain the spare space of edge servers (Lines 3–6). Then we formulate the initial resource scheduling strategies randomly and conduct crossover and mutation operation to generate $2H$ solutions (Line 9). Aiming to select the appropriate H solutions, we calculate the fitness functions for each solution and do the selection operation according to Algorithm 3 (Lines 10–13). This process would repeat until the number of iterations reached I . Based on the utility value of each solution, the alternative solutions are sorted and we select the optimal solution which has the highest utility value (Lines 16–20). The flow chart of selecting the optimal strategy is shown in Figure 4. Finally, the resource allocation records are updated and output (Line 21).

4. Experimental Results and Analysis

In this section, simulations and experiments are conducted to evaluate the performance of the proposed method BRSM. First, the parameter settings are introduced. Then, the performance evaluation on BRSM is presented.

Input: X, A, I, S

Output: A

1: $t = T_s$

2: **While** $t < T_e$ **do**

3: Update the running mobile service collection S

4: **for** $n = 1$ to N **do**

5: Obtain $is_{t,n}$ by Algorithm 1

6: **end for**

7: $i = 1$

8: **While** $i \leq I$ **do**

9: Conduct crossover and mutation

10: **for** the solutions in X **do**

11: Calculate fitness values

12: **end for**

13: Conduct selection by Algorithm 3

14: $i = i + 1$

15: **end while**

16: **for** $h = 1$ to H **do**

17: Calculate utility values by formulas (26)–(28)

18: **end for**

19: Sort the solutions according to the utility values

20: Select the optimal strategy by formula (29)

21: Update A by Algorithm 2

22: **end while**

23: **return** A

ALGORITHM 4: Balanced resource scheduling method **BRSM**.

4.1. Simulation Setup. In the experiment, we select LENOVO TS250 as the edge server, which consists Intel Xeon-E3-1225V6, Quad-Processor clocked at 3.4 GHz and 4 GB of RAM. To conduct the experiment evaluation, 5 datasets of different scales are generated, whose number of mobile services are 1000, 2000, 3000, 4000, 5000 respectively. Additionally, the concrete parameter settings in the experiment are presented in Table 2 [18]. Specifically, datasets can be downloaded at https://pan.baidu.com/s/1coXNu8B5_YdXAWTBJSn6uQ.

The VM migration cost and the energy consumption of VM migrations are the two criteria to judge whether the method BRSM is efficient. Aiming to represent the superiority of BRSM intuitively, we adopt comparative methods to conduct the experiment evaluation. The comparative methods are specifically described as follows.

- (i) *First Fit (FF)*. In the order of starting addresses of edge servers, the resources in each edge server are checked and the mobile services are respectively assigned to the first edge server encountered with enough resources.
- (ii) *Best Fit (BF)*. The mobile services are offloaded to the edge server with enough resources respectively while the destination edge server has the least resources among all edge servers.

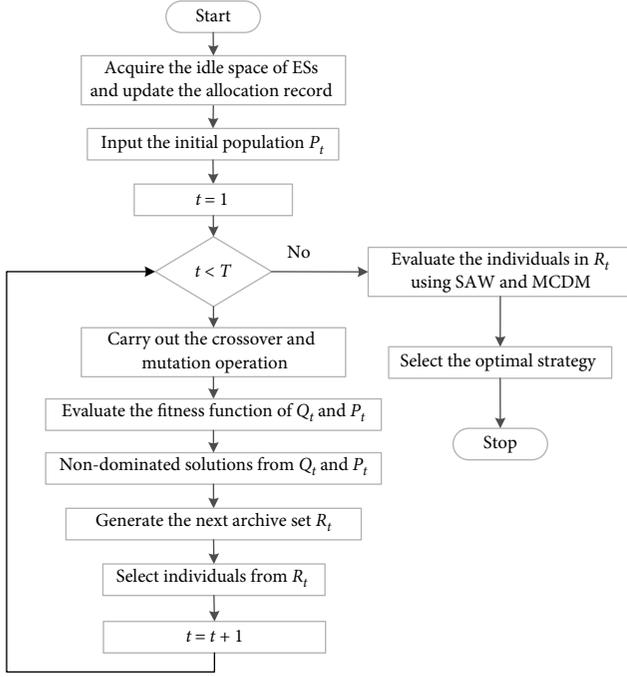


FIGURE 4: The flow chart of selecting the optimal strategy.

TABLE 2: Parameter settings.

Parameter description	Value
The number of VMs in each edge server	11
The number of running VMs in each edge server	[1, 7]
The transmission rate between Aps	540 Mb/s
The transmission rate between AP and edge server	1200 Mb/s
The duration time of VM	[1, 3]
The transmission data of VM (G)	[0.5, 0.8]

- (iii) *First Fit Decreasing (FFD)*. The mobile services are sorted in descending order according to the amount of required resources. Then the first mobile service is offloaded to the first edge server encountered with enough resources. This procedure continues until all mobile services are offloaded to edge servers.
- (iv) *Best Fit Decreasing (BFD)*. The mobile services are sorted in descending order according to the amount of required resources. Then the first mobile service is offloaded to the edge server which has the least resources among all edge servers but enough for the mobile service. This procedure is repeated until all mobile services are offloaded to edge servers.

4.2. Performance Evaluation of BRSM. In this section, we evaluate the number of employed edge servers, the resource utilization of edge servers, the VM migration cost and the energy consumption of VM migrations. The corresponding evaluation results are shown as follows.

4.2.1. Evaluation on the Utility Value of the Solutions. In the five sub-figures in Figure 5, the comparison of the utility value

of solutions at different mobile service scales is presented. As can be seen from the figures, regardless of the number of mobile services, there are always 3 solutions generated by BRSM. Among the generated solutions, the most balanced service offloading strategy is acquired, depending on the utility value in (30). The service offloading strategy with maximum utility value is selected as the most optimal offloading strategy. For instance, in Figure 4(a), the final service offloading strategy is solution 2 as it has the highest utility value.

4.2.2. Evaluation on the Amount of Employed Edge Servers. In the edge server-based WMAN, the number of employed edge servers is a key factor to measure the resource utilization of edge servers. Therefore, it is necessary to evaluate the number of employed edge servers. Given the 5 different scale datasets employed in FF, BF, FFD, BFD, and BRSM, the number of employed edge servers differs from each other after VM migrations, which is illustrated in Figure 6. In Figure 6, it is obvious that FF and BF utilize the same number of edge servers. FFD as well as BFD also utilizes the same number of edge servers. However, BRSM employs fewer edge servers than the other four methods.

4.2.3. Evaluation on Resource Utilization. Resource utilization of edge servers is of great significance, which is calculated by virtue of the number of running VMs in each edge server. High resource utilization represents the rational utilization of resources.

In Figure 7, the comparison of average resource utilization by FF, BF, FFD, BFD, and BRSM is presented. FF and BF have the same resource utilization, while FFD and BFD likewise achieve the same utilization. Distinctly, compared with FF, BF, FFD and BFD, BRSM has the highest resource utilization value, proving the superiority of the proposed method BRSM in the respect of resource utilization.

In the process of dynamic VM migrations, the resource utilization of edge servers, which changes with time instants, is taken into consideration. To monitor resource usage in real time, we track the resource utilization of edge servers when there are 5000 mobile services during the execution period. Figure 8 illustrates the comparison of real-time resource utilization by FF, BF, FFD, BFD, and BRSM. According to the analysis, the resource utilization of edge servers exceeds 80% during the execution period. Besides, BRSM achieves better resource utilization than FF, BF, FFD, and BFD in almost every instant, guaranteeing the good performance of BRSM.

4.2.4. Evaluation on Migration Cost. The migration cost represents the time consumption of VM migrations across edge servers. However, the migration cost is divided into two parts, which are the cost between AP and edge server as well as the cost between APs.

Moreover, the transmission time of dirty pages caused by last round of data transmission is considered. Figure 9 shows the comparison of migration cost by FF, BF, FFD, BFD, and BRSM, using 5 different scales datasets. It is intuitive that FF and BF own the same migration cost while FFD and BFD also own the same migration cost. Nevertheless, our proposed

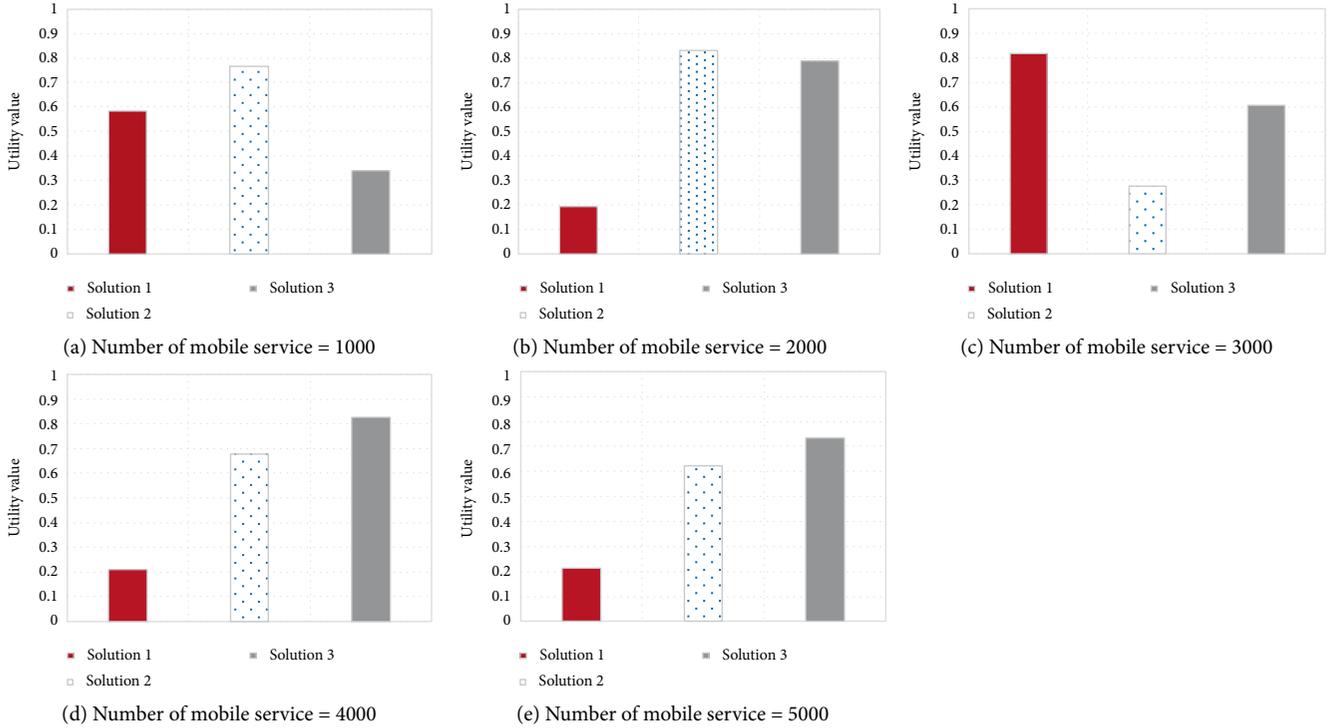


FIGURE 5: Comparison of the utility value of solutions at different mobile service scales.

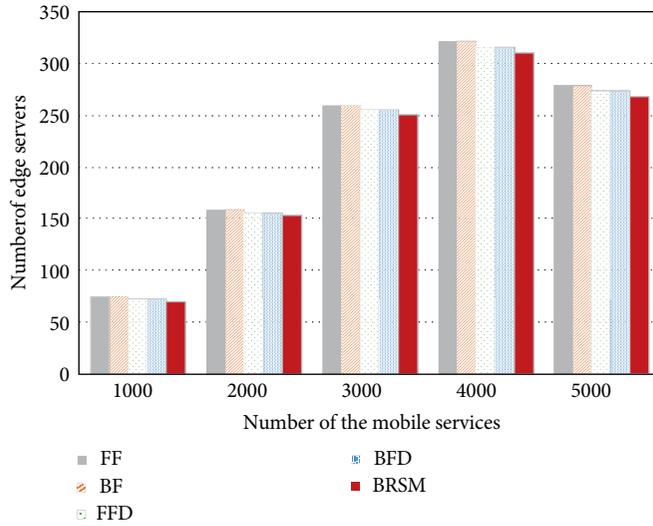


FIGURE 6: Comparison of the number of employed edge servers by FF, BF, FFD, BFD and BRSM.

method BRSM has the least migration cost when compared with the other 4 methods.

4.2.5. *Evaluation on Energy Consumption.* As mentioned in section 2, the energy consumption of VM migrations is an important criterion to judge whether the offloading situation of the offloading strategy is suitable. The lower the energy consumption is, the more efficient the offloading strategy becomes. Thus, the energy consumption of FF, BF, FFD, BFD,

and BRSM are compared in Figure 10. Under the different scales of mobile services, the energy consumption of VM migrations are different. It is obvious that FF and BF has the same energy consumption during the VM migrations, and FFD, as well as BFD, also has the same energy consumption. Moreover, the energy consumption of BRSM is lower than the energy consumption of FF, BF, FFD, and BFD, which draws the conclusion that BRSM achieves better energy conservation than the 4 comparative methods.

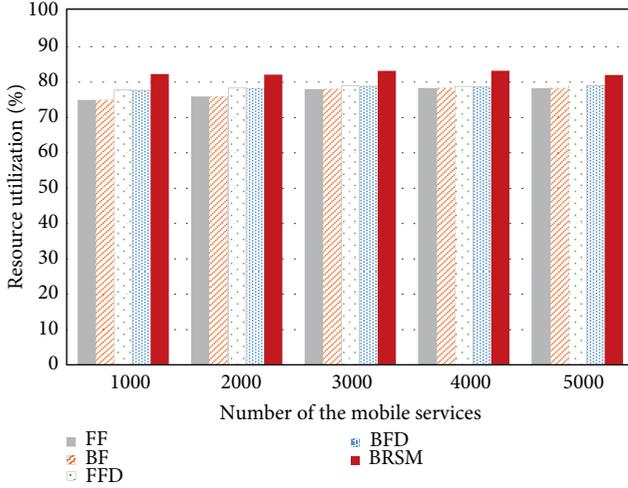


FIGURE 7: Comparison of the average resource utilization of edge servers by FF, BE, FFD, BFD and BRSM.

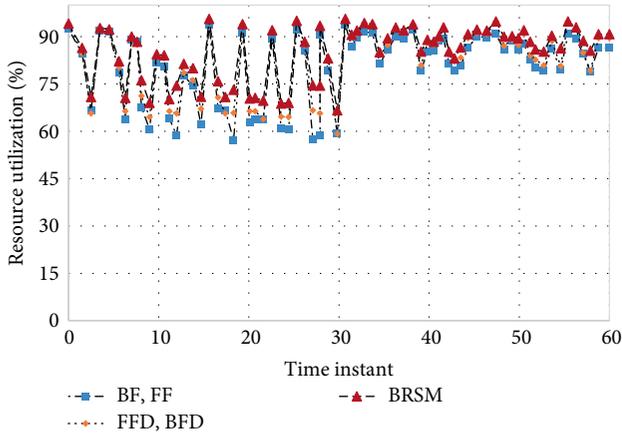


FIGURE 8: Comparison of real-time resource utilization of 5000 mobile services by FF, BE, FFD, BFD and BRSM.

5. Related Work

With the adoption of fossil fuels, various problems such as air pollution and rapid resource consumption have aroused, jeopardizing the sustainable development of environment [1, 15]. To improve the environment, renewable energy is used as a substitute for fossil fuels [15, 16]. However, the supply of renewable energy depends on many uncontrollable natural factors, which is intermittent. Consequently, complex systems are utilized to integrate kinds of renewable energy to achieve continuous supply of stable energy [17, 18]. With the large-scale data in complex systems, cloud computing emerges as an applied paradigm [19, 20]. The mobile devices transmit the data to the cloud platform while the long distance between mobile devices and the cloud leads to unneglectable transmission delay. Thus, the data are transmitted to the edge servers which are close to mobile devices, reducing the transmission time consumption [21–23]. Edge computing has been studied thoroughly in many aspects, including virtual resource allocation strategy formulation [24], energy consumption of

mobile stations [25], QoE and operation cost of users movement [26], security problems of data transmission [27], load balance of the network [28].

Zhou et al. [24] formulated a heterogeneous networks framework to support computing and content caching. Sheu et al. [25] proposed an advanced power saving mechanism named A-PSM, which uses the average packet inter-arrival time to adjust the sleep window to get command of power usage. Guan et al. [26] divided a metropolitan area into different clusters to minimize the sum of handovers between MEC regions, increasing the QoE and reducing the operation cost in WMAN. Baskaran et al. [27] presented an enhanced scheme compared with Privacy Key Management Protocol Version 2 (e-PKMv2). By feat of a new technique named Blind Key Distribution (BKD), the security and reliability problems could be improved. Jakimoski et al. [28] designed a vertical handover decision algorithm to optimize the load of WMAN, relying on the velocity of the mobile devices and load of the whole network.

Generally, the applications would be more computation-intensive as various requirements would be added to them for resource response [5, 29]. Therefore, the transmission delay during the process of connecting the mobile users and servers is costly in complex systems [5, 30, 31]. Offloading tasks to edge servers is a suitable way to reduce the delay of applications. Edge servers are reliable server clusters and the physical resources of edge servers are provided in the form of different kinds of VMs. Thus, users could resort VMs to deal with computing tasks through establishing wireless communication with nearby edge servers.

Cui et al. [32] devised a software-defined cooperative offloading model based on the regularities of distribution of users and used an online task scheduling algorithm to reasonably distribute energy among devices. Similarly, Chen et al. [33] studied the edge server deploy problem as well, aiming to minimize the access delay of users and decrease the total cost which is represented by the number of edge servers. To achieve the goals, heuristic and clustering algorithms were designed for reducing delay while an integer linear programming is formulated to lessen the used edge servers. Panigrahi et al. [34] proposed an energy efficient offloading approach to increase the resource utilization of edge servers for better energy usage. Analogously, Gai et al. [35] designed a dynamic energy-aware edge server-based mobile cloud computing model for energy saving according to dynamic edge servers (DCL)-based model.

When the scale of mobile users expands rapidly, they access the edge servers frequently and the resources run out rapidly. Consequently, the new requests from mobile devices could be rejected and a large number of users are queued. As the access delay caused by waiting is nonnegligible, negative effects would be brought about and the QoE for users would drop dramatically [36, 37]. Different aspects of researches on edge servers are conducted to reduce the transmission delay [36–38].

Jia et al. [36] devised an algorithm to optimize the edge server placement problem in WMAN. The algorithm could assign mobile users to edge servers which have been placed to balance the workload. On the other hand, a novel model is

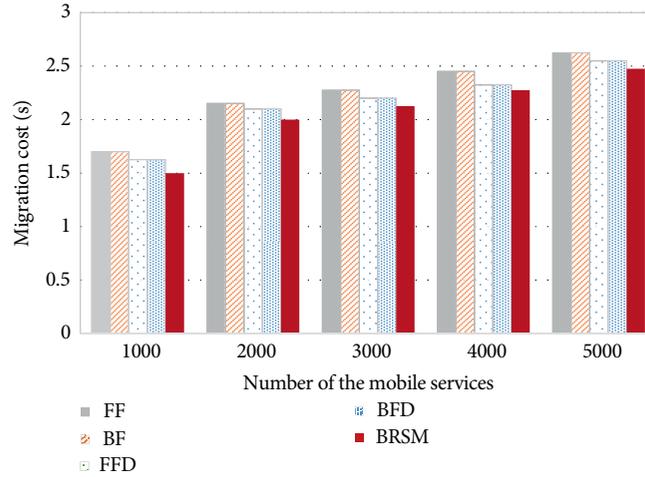


FIGURE 9: Comparison of migration cost by FF, BF, FFD, BFD and BRSM.

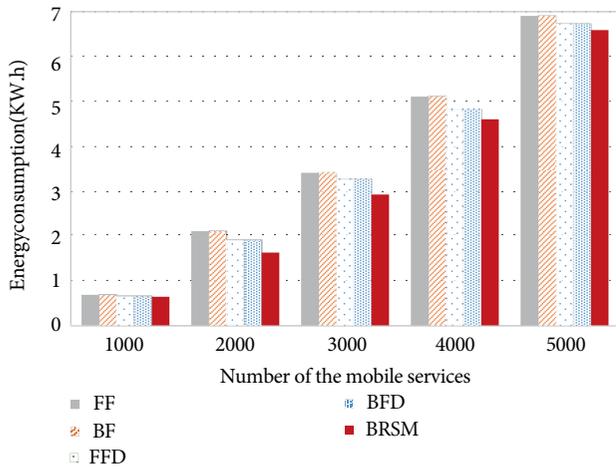


FIGURE 10: Comparison of energy consumption by FF, BF, FFD, BFD and BRSM.

introduced to calculate the response delay of offloading tasks and utilized two algorithms to solve the optimization problem which could minimize the maximum response time [39]. Ma et al. [40] presented a new heuristic algorithm (NHA) and particle swarm optimization (PSO) algorithm for reducing delay which are more effective than existing methods. Rodrigues et.al. [41] designed a scheme to minimize service delay with two edge server servers, focusing on communication and computation elements and using VM migration to control processing delay and transmission delay.

6. Conclusion and Future Work

The ability to provide multifarious cloud services is significant for the modern WMAN system. In this paper, a balanced resource scheduling method, named BRSM, is developed for trade-offs between VM migration cost and energy consumption for edge server management. In terms of the real-time load conditions of edge servers and the number of the

offloading requests, the proposed method allows the system to formulate appropriate service offloading strategies to relieve the load on the overload edge servers and realize efficient application execution. With the increasing service requests sent from mobile users, the cloud services are guaranteed not to be greatly influenced through offloading the services in the overload edge servers to other edge servers which have idle computing resources and storage. First, the offloading problem is defined as a multi-objective optimization problem to reduce the VM migration cost and achieve energy conservation during the offloading process. Then, NSGA-III is adopted to deal with the multi-objective optimization problem. Finally, relative experiments are conducted to evaluate the performance of the proposed method BRSM.

In future work, the proposed method BRSM would be extended to the real-world scenario of WMAN. Besides, execution time limits are specified for computing tasks to identify a computation offloading scheme to reduce the migration cost as well as the energy consumption.

Data Availability

The edge service data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research is supported by the National Natural Science Foundation of China under grant nos. 61702277 and 61872219. This work is also supported by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund, Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology (CICAET), and

College Students' Enterprise and Entrepreneurship Education Program of NUIST, CSEEEP.

References

- [1] B. V. Mathiesen, H. Lund, D. Connolly et al., "Smart energy systems for coherent 100% renewable energy and transport solutions," *Applied Energy*, vol. 145, pp. 139–154, 2015.
- [2] C. Wei, M. Benosman, and T. Kim, "Online parameter identification for state of power prediction of lithium-ion batteries in electric vehicles using extremum seeking," *International Journal of Control, Automation and Systems*, 2019.
- [3] D. Flórez-Orrego, I. B. Henriques, T.-V. Nguyen et al., "The contributions of prof. Jan Szargut to the exergy and environmental assessment of complex energy systems," *Energy*, vol. 161, pp. 482–492, 2018.
- [4] A. Alrawais, A. Althothaily, J. Yu, C. Hu, and X. Cheng, "SecureGuard: a certificate validation system in public key infrastructure," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 5399–5408, 2018.
- [5] V. Meena, M. HariPrasath, V. Kalpana, K. ArunKumar, and J. SenthilKumar, "Optimal resource reservation for offloaded tasks in mobile cloud computing," in *2017 2nd International Conference on Communication and Electronics Systems*, IEEE, Coimbatore, India, 2017.
- [6] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 974–983, 2015.
- [7] K. Gai, M. Qiu, H. Zhao, L. Tao, and Z. Zong, "Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing," *Journal of Network and Computer Applications*, vol. 59, pp. 46–54, 2016.
- [8] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo, "Capacitated cloudlet placements in wireless metropolitan area networks," in *2015 IEEE 40th Conference on Local Computer Networks*, IEEE, Clearwater Beach, FL, USA, 2015.
- [9] R. Morabito, "Virtualization on internet of things edge devices with container technologies: a performance evaluation," *IEEE Access*, vol. 5, pp. 8835–8850, 2017.
- [10] Y. Liu, M. J. Lee, and Y. Zheng, "Adaptive multi-resource allocation for edge server-based mobile cloud computing system," *IEEE Transactions on Mobile Computing*, vol. 15, no. 10, pp. 2398–2410, 2016.
- [11] M. Chen, Y. Hao, Y. Li, C. F. Lai, and D. Wu, "On the computation offloading at ad hoc edge server: architecture and service modes," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 18–24, 2015.
- [12] K. Zhang, Y. Mao, S. Leng et al., "Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks," *IEEE access*, vol. 4, pp. 5896–5907, 2016.
- [13] 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.
- [14] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2013.
- [15] A. H. Fathima and K. Palanisamy, "Optimization in microgrids with hybrid energy systems—a review," *Renewable and Sustainable Energy Reviews*, vol. 45, pp. 431–446, 2015.
- [16] L. Qi, Q. He, F. Chen et al., "Finding all you need: web APIs recommendation in web of things through keywords search," *IEEE Transactions on Computational Social Systems*, pp. 1–10, 2019.
- [17] D. Alahakoon and X. Yu, "Smart electricity meter data intelligence for future energy systems: a survey," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 1, pp. 425–436, 2016.
- [18] L. Qi, Y. Chen, Y. Yuan, F. Shucun, Z. Xuyun, and X. Xiaolong, "A QoS-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems," *World Wide Web*, 2019.
- [19] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: a survey," *Future Generation Computer Systems*, vol. 56, pp. 684–700, 2016.
- [20] G. Wenwen, Q. Lianyong, and X. Yanwei, "Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment," *Wireless Communications and Mobile Computing*, vol. 2018, 8 pages, 2018.
- [21] P. Mach and Z. Becvar, "Mobile edge computing: a survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.
- [22] Z. Yang, Y. Huang, L. Xing et al., "Efficient secure data provenance scheme in multimedia outsourcing and sharing," *Computers, Materials & Continua*, vol. 56, no. 1, pp. 1–17, 2018.
- [23] Y. Zhang, Y. Zhou, F. Wang, Z. Sun, and Q. He, "Service recommendation based on quotient space granularity analysis and covering algorithm on Spark," *Knowledge-Based Systems*, vol. 147, pp. 25–35, 2018.
- [24] Y. Zhou, F. R. Yu, J. Chen, and Y. Kuo, "Resource allocation for information-centric virtualized heterogeneous networks with in-network caching and mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 11339–11351, 2017.
- [25] S. T. Sheu, L. W. Chen, and J. Chen, "Advanced power saving mechanism in IEEE 802.16 m wireless metropolitan area networks," in *2012 IEEE 75th Vehicular Technology Conference*, IEEE, Yokohama, Japan, 2012.
- [26] X. Guan, X. Wan, J. Wang, X. Ma, and G. Bai, "Mobility aware partition of MEC regions in wireless metropolitan area networks," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops*, IEEE, USA, 2018.
- [27] S. B. M. Baskaran and G. Raja, "Blind key distribution mechanism to secure wireless metropolitan area network," *CSI transactions on ICT*, vol. 4, no. 2-4, pp. 157–163, 2016.
- [28] K. Jakimoski and T. Janevski, "Vertical handover improvements from WLAN to WMAN or WWAN technologies," in *2013 11th International Conference on Telecommunication in Modern Satellite, Cable and Broadcasting Services*, IEEE Serbia, 2013.
- [29] Y. Yuan and W. Banzhaf, "Arja: automated repair of java programs via multi-objective genetic programming," *IEEE Transactions on Software Engineering*, 2018.
- [30] J. Zhang, N. Xie, X. Zhang, K. Yue, W. Li, and D. Kumar, "Machine learning based resource allocation of cloud computing in auction," *Computers, Materials & Continua*, vol. 56, no. 1, pp. 123–135, 2018.

- [31] Y. Zhang, K. Wang, Q. He et al., "Covering-based web service quality prediction via neighborhood-aware matrix factorization," *IEEE Transactions on Services Computing*, 2019.
- [32] Y. Cui, J. Song, K. Ren et al., "Software defined cooperative offloading for mobile edge servers," *IEEE/ACM Transactions on Networking*, vol. 25, no. 3, pp. 1746–1760, 2017.
- [33] L. Chen, J. Wu, G. Zhou, and L. Ma, "QUICK: QoS-guaranteed efficient edge server placement in wireless metropolitan area networks," *The Journal of Supercomputing*, vol. 74, no. 8, pp. 4037–4059, 2018.
- [34] C. R. Panigrahi, B. Pati, and M. Tiwary, "EEOA: Improving energy efficiency of mobile edge servers using efficient offloading approach," in *2015 IEEE International Conference Advanced Networks and Telecommunications Systems*, IEEE Kolkata, 2015.
- [35] K. Gai, M. Qiu, H. Zhao, L. Tao, and Z. Zong, "Dynamic energy-aware edge server-based mobile cloud computing model for green computing," *Journal of Network and Computer Applications*, vol. 59, pp. 46–54, 2016.
- [36] M. Jia, J. Cao, and W. Liang, "Optimal edge server placement and user to edge server allocation in wireless metropolitan area networks," *IEEE Transactions on Cloud Computing*, vol. 5, no. 4, pp. 725–737, 2017.
- [37] Y. Yuan, Y. S. Ong, and A. Gupta, "Objective reduction in many-objective optimization: evolutionary multiobjective approaches and comprehensive analysis," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 189–210, 2018.
- [38] C. Wu, E. Zapevalova, Y. Chen, and F. Li, "Time optimization of multiple knowledge transfers in the big data environment," *Computers, Materials & Continua*, vol. 54, no. 3, pp. 269–285, 2018.
- [39] M. Jia, W. Liang, Z. Xu, M. Huang, and Y. Ma, "QoS-aware edge server load balancing in wireless metropolitan area networks," *IEEE Transactions on Cloud Computing*, p. 1, 2018.
- [40] L. Ma, J. Wu, L. Chen, and Z. Liu, "Fast algorithms for capacitated edge server placements," in *2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design*, IEEE, New Zealand, 2017.
- [41] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, "A PSO model with VM migration and transmission power control for low service delay in the multiple edge servers ECC scenario," in *2017 IEEE International Conference on Communications*, IEEE, Paris, France, 2017.

