International Journal of Innovative
Computing, Information and Control
Volume **17**, Number **4**, August **2021**

 ISSN 1349-4198
pp. 1401–1419

AN EFFICIENT SERVICE MIGRATION MODEL BASED ON IMPROVED GENETIC ALGORITHM IN MOBILE EDGE COMPUTING ENVIRONMENT

Xiuguo Zhang[1], Yufei Liu[1,*], Zhiying Cao[1,*], Huijie Zhou[2] and Fengge Zhang[1]

[1]School of Information Science and Technology
Dalian Maritime University
No. 1, Linghai Road, Dalian 116026, P. R. China
Zhangxg@dlmu.edu.cn; *Corresponding authors: { lyf; czysophy }@dlmu.edu.cn

[2]Wangshen Information Technology (Beijing) Co., Ltd.
No. 26, Xizhimen South Road, Xicheng District, Beijing 100032, P. R. China
zhouhjstudy@163.com

Received January 2021; revised May 2021

Abstract. *Mobile edge computing (MEC) reduces network operation and service delivery delay by providing IT service environment and cloud computing capability at the edge of mobile network. However, in mobile edge computing environment, resource constrained mobile servers may not be able to achieve efficient computing requirements, and thus cannot guarantee the quality of service execution. Therefore, it is necessary to select the adjacent MEC servers to get computing power support. This paper focuses on service migration in mobile edge computing environment. Considering factors such as mobile server monitoring cost, service execution cost and data transmission cost, a threshold-based edge server selection strategy is designed and an efficient service migration model related to communication distance, network bandwidth and other factors is constructed. A genetic algorithm incorporating back learning and Levy flight mechanism is adopted to solve the service migration model. Experiment results show that the edge server selection strategy and service migration model proposed in this paper have obvious advantages in convergence speed, convergence accuracy, energy consumption and accuracy.*

Keywords: Mobile service, Edge server, Service migration, Server selection, Genetic algorithm

1. **Introduction.** With the development of mobile devices, the number of wireless terminals and the energy consumption of mobile networks have increased dramatically. However, mobile devices have limits of battery power, computing resources and continuous supply capacity, especially in the case of few network facilities or mobile devices, which greatly affects the user experience. Therefore, making full use of the surrounding computing resources and reducing the energy consumption and delay of service execution have become inevitable requirements to improve the user experience [1]. Meanwhile, the development of new mobile applications such as augmented reality and gesture recognition is also underway. These applications are not only computation-intensive, but also sensitive to latency. However, mobile servers (MS) are limited in terms of resources and computing power [2,3]. Mobile edge computing (MEC) will provide a powerful platform to solve the problems of network delay, congestion and capacity in the future and realize various visions of the next generation Internet [4]. It breaks the boundary between the unprecedented increase in computing demand and the ever-increasing computing quality

DOI: 10.24507/ijicic.17.04.1401

required by user services [5]. By deploying rich computing and storage resources on the edge server, people can migrate the services on the mobile server to the edge server for calculation by means of wireless communication, which enlarges the computing power of the mobile server [6].

One of the major challenges for MEC service migration is computing migration. Running computation-intensive mobile services on resource-constrained MS consumes a lot of resources and energy. To solve this problem, the concept of computing migration emerged. Calculating to determine whether the service is executed on MS or on the edge server, so the computing power of MS is further enhanced and the energy consumption of MS is also reduced when executing the service. In order to design an efficient and reliable computing migration scheme, it is necessary to comprehensively consider the dynamic changes of the load, service attributes and network status of the MS [7].

In the process of service migration, energy consumption and time delay are important factors. Gao et al. [8] studied the choice of optimal dynamic computing migration mode in the mechanical low-delay Internet of Things, and constructed all feasible patterns of arrangement by the social connection structure among mobile intelligent device users. Wei et al. [9] studied the segmentation of services uploading to the edge server, and proposed a maximum energy-saving service priority algorithm to solve the problem of minimum energy consumption. However, types of services are different, and the demands for resources are naturally different. Therefore, it is necessary to combine the energy consumption and delay. It is feasible to assign dynamic weights according to the different types of services.

In addition, the mobility of mobile servers is an important factor that cannot be ignored. In order to ensure the quality of services and improve the resource utilization of edge servers, it is essential to develop a good dynamic MEC server selection strategy for mobile server mobility [10]. For example, Fan et al. [11] proposed a novel scheme of joint WiFi and cellular offloading is proposed to optimally reduce the latency and energy consumption of MTs in task processing. Based on the statistical characteristics of MTs' task generation, the scheme serves as strategic guidance for computation offloading without frequent execution of the optimization algorithm. Li and Jiang [12] proposed a distributed task unloading strategy to low load base station group under MEC environment. Firstly, the communication resource, computing resource and task queue of low load base station group are modeled to quantify the energy cost in the process of task unloading. Then, the potential game model is used to solve the problem of distributed task unloading. The target function of energy optimization based on delay limitation is transformed into the potential game equation, and the mobile device selects MEC nodes according to the game results to calculate the unloading. The mobile device migrates all services to the edge server for processing.

In this paper, the service migration in the mobile edge computing environment is studied. The energy consumption and delay in the service migration are considered comprehensively, and the dynamic selection and service migration of the edge server are realized according to the mobility of the mobile server.

The rest of this paper is organized as follows. Section 2 is related research. Section 3 presents the service migration model. Section 4 introduces the solution algorithm and server selection strategy. Section 5 shows experiments and result analysis. Section 6 summarizes the research of this paper.

2. **Related Research.** When MEC technology was not mature, many similar concepts appeared, such as mobile micro cloud (MMC), and small cellular cloud (SCC). In the current MEC technology, Jiang et al. [13] proposed a user interest community evolution model based on subgraph matching for social networking in mobile edge computing

environments. Mashhadi et al. [14] proposed an optimal auction for delay and energy constrained task offloading. Fan et al. [11] proposed a novel scheme of joint WiFi and cellular offloading in order to optimally reduce the latency and energy consumption of MTs in task processing.

The above methods accurately locate or predict the location of the server, but they only consider the location relationship between the servers and ignore the impact of the data transmission delay or transmission energy consumption of each server on subsequent service migration.

Regarding the minimum energy consumption as the optimization goal of service migration, the migration problem was transformed into a joint optimization problem of computing and communication by Sardellitti et al. [15]. Precoding matrix information was transmitted by edge server through Ethernet, and its own computing resources were allocated to each migrated service according to the CPU cycle for calculation. The time problem of optimization service calculation was transformed into a non-convexity problem, and an iterative algorithm was proposed to converge to the local optimal solution of the non-convex problem. However, this migration strategy is only applicable to services with a small amount of data and CPU cycles for processing.

Labidi et al. [16] proposed a deterministic online strategy based on a co-location decision learning framework. This strategy is suitable for the migration decision of all service migrations in a multi-user scenario. The dynamic environment of time-varying channels is considered to ensure user mobile terminal experience quality and latency constraints. Although this method considers the dynamic change of channel, interference of multiple devices is ignored in the multi-user scenario in this paper.

According to the extreme value theory, Liu et al. [17] modeled the real-time and reliability constraints by the length of the user's task queue. Task transfer consists of communication and calculation. However, these efforts rarely take computational reliability into account.

Regarding the minimum delay as the optimization goal, Meng et al. [18] studied the problem of delayed optimal computing migration for computationally constrained MEC systems, considering the computational task queues on MEC servers. Chen and Hao [19] proposed a solution to meet the requirements of minimum delay in ultra-dense network scenarios, which represented the migration problem as a mixed integer nonlinear calculation process and divided it into two problems of the migrated service placement and the resource allocation. Ding et al. [20] applied game theory to designing a service migration scheme, in which the game theory is used to solve the problem of computing offload, based on the proposed beneficial task offload theory.

In summary, there are still various problems with the current mainstream service migration methods. However, regarding the service migration strategy with the optimization goal of minimum delay, no consideration is given on whether the mobile device can bear the energy consumption of service execution, which may cause the migration strategy to fail to be used normally due to insufficient mobile device hardware conditions during the service migration process. Algorithm with the optimization goal of minimum energy consumption can meet the mobile device's delay limit. The efficiency of such a strategy depends on the mobile device's transmit power and channel quality. Since the currently studied strategies are verified through simulation experiments, they cannot be restored in real migration conditions, and the time variability of channel quality can be ignored.

In the context of studying edge computing, some migration algorithms have low execution efficiency and often fall into the situation of local optimal solutions, so the desired target value cannot be guaranteed to be optimal.

3. Service Migration Mobile Construction.

3.1. Scene description.

In the mobile edge computing environment, mobile servers can perform computing intensive or delay sensitive services. When the amount of computation of services is large, the resource constrained mobile servers may not be able to achieve efficient computing requirements, and thus cannot guarantee the quality of service execution. Therefore, it is necessary to select the adjacent MEC servers to provide computing power support. However, MEC servers which have limitation of coverage and the contradiction between the mobility of mobile servers may lead to high transmission delay and even interrupt the running mobile services. It makes it difficult to guarantee the continuity of services. Therefore, when selecting MEC servers, it is necessary to select MEC servers according to the communication distance between mobile servers and edge servers, network bandwidth and other dynamic factors.

This paper shows that each service can be divided into several independent and series executed sub services, in which the CPU operation (data size) is carried by these sub services. Some sub services can be migrated from the mobile server to the edge server for calculation. With the computing power of the edge server, the service integrity can be improved execution performance. The mobile service migration scenario is shown in Figure 1.

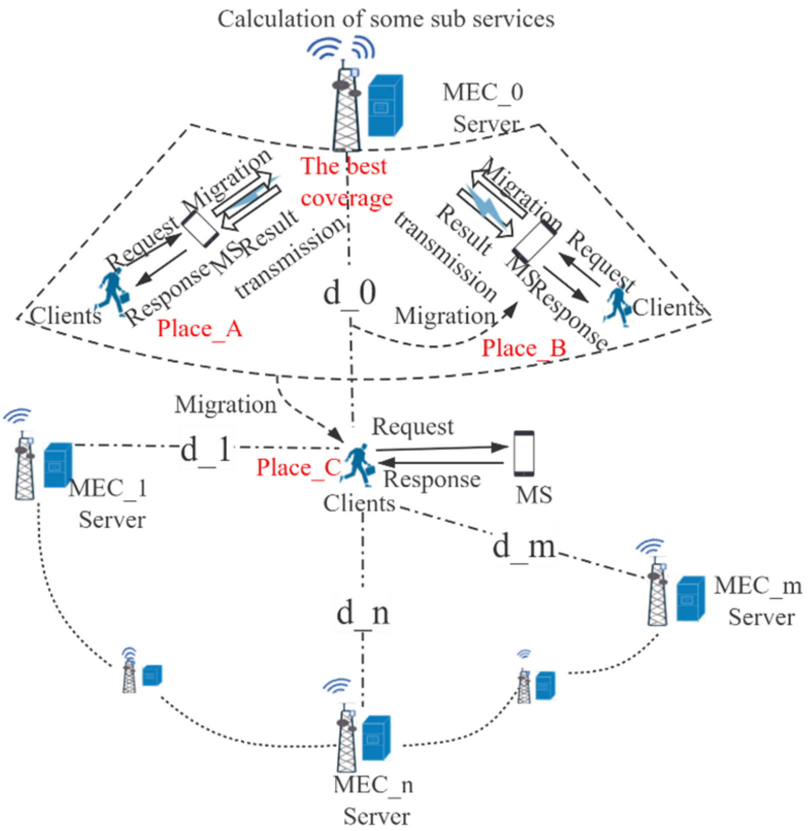

FIGURE 1. Simple scenario diagram of service migration

The mobile service migration scenario in Figure 1 is described as follows.

1) Clients (mobile users) carry mobile servers (MS) between edge servers. A series of services is deployed on mobile servers.

2) There are $m+1$ MEC servers around the mobile server, namely $MEC_0, MEC_1, \ldots, MEC_m$. In the $m+1$ mobile service migration scenario, the following edge servers are also deployed with the same services as the mobile server.

3) The user requests a service S on the mobile server. S includes N sub services, which are recorded as $S_1, S_2, \ldots, S_n$. There are antecedent and successor dependencies between two adjacent sub services.

4) After the mobile server receives the user's request, it starts from the first sub service S_1. We need to use the edge server selection algorithm based on energy consumption threshold to select an edge server with the best communication distance and execution status among the $m+1$ MEC servers adjacent to the mobile server.

5) In order to alleviate the computing pressure of mobile server and ensure the execution quality of each sub service, it is necessary to obtain the resource information of the server and judge the current sub service S_n ($n \in \{1, \ldots, N\}$) on the mobile server, or to migrate to the sub service S_n ($n \in \{1, \ldots, N\}$) on the corresponding edge server.

6) For N sub services, mobile server and edge server conduct interactive calculation, and finally respond the execution result of service request to the client.

This paper considers that the execution position of each sub service in the service storage queue of mobile server is uncertain. It is executed on the mobile server side or on the MEC side. No matter which end is executed, there will be execution energy consumption, upload energy consumption, receiving energy consumption, execution delay, upload delay and receiving delay. The dependency relationship of sub services is shown in Figure 2.

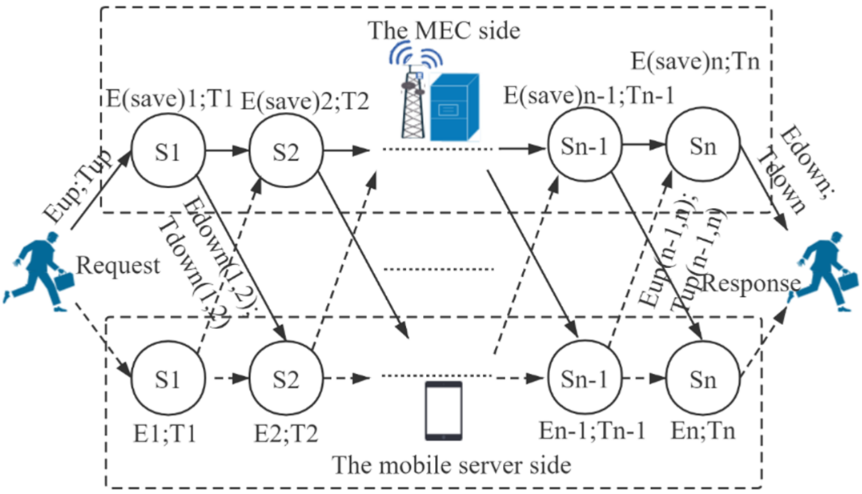

Figure 2. Sub service execution flowchart

According to [21], the partition granularity of migration can be divided into methods, objects and classes of services. In this paper, services are divided into N sub services according to methods and objects and stored in the computing service queue, so that N sub services can be executed on different servers to expand computing resources.

The execution energy consumption, upload energy consumption, receiving energy consumption, execution delay, upload delay and receiving delay are all expressed in the vertex and directed edge of the digraph, and a sub service execution process and dependency relationship based on digraph is established. D is the set of directed edges, and it also represents the dependency relationship among sub services. In the directed graph, the vertex represents each sub service S_n ($n \in \{1, \ldots, N\}$). On the mobile server side, the two values on each vertex are execution energy consumption E_n ($n \in \{1, \ldots, N\}$) and execution delay T_n ($n \in \{1, \ldots, N\}$). The sub service is on the edge server side, and the vertex represents the energy consumption saved by the mobile server $E(save)_n$ ($n \in \{1, \ldots, N\}$) and execution delay T_n ($n \in \{1, \ldots, N\}$). Each edge represents two values, which are the upload energy consumption $E_{up}(n, n+1)$ and upload delay $T_{up}(n, n+1)$ or down transmission energy consumption $E_{down}(n, n+1)$ and downlink delay $T_{down}(n, n+1)$.

3.2. **Mobile server monitoring costs.** Mobile server is always moving. This paper makes the mobile server into two kinds of performance when the service is migrated. One is getting distance information for edge servers, which is the moving distance between the mobile user and the edge server. The other is getting the status information of the edge server.

About the cost calculation of total delay when mobile server monitoring, the total delay during monitoring T_d is equal to the sum of the delay required to obtain the distance information of the edge servers T_{dis} and the delay required to obtain the status information of the surrounding edge servers T_{stu}.

$$T_d = T_{dis} + T_{stu} \tag{1}$$

About cost calculation of total energy consumption during mobile server monitoring, the total energy consumption during monitoring E_d is equal to the sum of the energy consumption E_{dis} required to obtain the distance information of the edge servers and the energy consumption E_{stu} required to obtain the status information of the surrounding edge servers.

$$E_d = E_{dis} + E_{stu} \tag{2}$$

3.3. **Service execution costs.** When the service migration strategy decides to execute the sub service on the mobile server, the mobile server does not need to upload the sub service data. It calculates in the mobile server processing unit. The execution time of each sub service is T_{MS}, the energy consumption of execution is E_{MS}, the CPU operation number (data size) of the sub service is C_{MS}, the CPU clock frequency of mobile server is F_{MS}, and mobile server execution power is P_{MS}. This paper assumes that the execution of the sub service is in a good network environment and sufficient mobile server resources.

Cost of time spent executing sub services on the mobile servers T_{MS} is equal to the number of CPU operations (data size) of the sub service C_{MS} divided by the CPU clock frequency of the mobile server F_{MS}.

$$T_{MS} = \frac{C_{MS}}{F_{MS}} \tag{3}$$

Calculation of energy consumption of sub services when executed on mobile servers E_{MS} is equal to the time consumed by the sub service execution T_{MS} and the product of the power of mobile server P_{MS}.

$$E_{MS} = T_{MS} P_{MS} = \frac{C_{MS}}{F_{MS}} P_{MS} \tag{4}$$

When the service migration strategy decides to migrate the sub service to the MEC server for execution, the mobile server needs to upload the sub service data and calculate the service in the edge server processing unit.

Cost calculation of time spent when sub services execute on edge servers T_{MEC} is equal to the number of CPU operations (data size) of the sub service C_{MEC} divided by the CPU clock frequency of the edge server F_{MEC}.

$$T_{MEC} = \frac{C_{MEC}}{F_{MEC}} \tag{5}$$

Cost calculation of energy consumption when sub services are executed on edge servers is also discussed. The energy saved on its mobile server (MS) E_{save} is equal to the product of the time consumed by the sub services at the edge server execution T_{MEC} and the power of the mobile server when the edge server is executing P_{slp}.

$$E_{save} = T_{MEC} P_{slp} = \frac{C_{MEC}}{F_{MEC}} P_{slp} \tag{6}$$

The execution location of each sub service is uncertain. This paper defines a service migration decision variable V_n ($n \in \{1, \dots, N\}$), which is used to indicate the execution location of each sub service. The formula for the decision variable is as follows:

$$V_n = \begin{cases} 0 & \text{Sub services execute on the edge server} \\ 1 & \text{Sub services execute on the mobile server} \end{cases} \tag{7}$$

Total service execution time T_{EX} is equal to the execution time of the sub service on the MEC server T_{MEC} and execution time on mobile server T_{MS}. The formula of total time consumed during execution is as follows:

$$T_{EX} = (1 - V_n) T_{MEC} + V_n T_{MS} = (1 - V_n) \frac{C_{MEC}}{F_{MEC}} + V_n \frac{C_{MS}}{F_{MS}} \tag{8}$$

Total execution energy consumption of service E_{EX} is equal to the energy consumption saved on the mobile server when the sub service is executed on the MEC server E_{save} and executing energy consumption on mobile E_{MS}. The formula of total energy consumption execution is as follows:

$$E_{EX} = (1 - V_n) E_{save} + V_n E_{MS} \tag{9}$$

3.4. **Data transmission costs.** This paper assumes that the execution location of each sub service is different. When the execution location of the current sub service is MEC server, and the execution location of the post sub service is the mobile server, the transmission delay and energy consumption of the downlink data should be calculated. Downlink data transmission time is T_r^{down}. The energy consumption of downlink data transmission is E_r^{down}. Data transmission is C_{trs}. Downlink broadband is B_r. Power received by mobile server is P_r^{down}. This paper assumes that the sub service is in a good network environment.

Cost calculation of sub service downlink transmission time T_r^{down} is equal to the data transmission amount C_{trs} divided by the downlink broadband B_r.

$$T_r^{down} = \frac{C_{trs}}{B_r} \tag{10}$$

Calculation of mobile server received power accords to Shannon's formula. The formula is as follows:

$$R_r = B_r \log_2(1 + SNR) \tag{11}$$

As shown in Shannon's formula, the data migration rate R_r is determined by channel bandwidth B_r and signal to noise ratio (SNR). The data migration rate R_r is set to a fixed value in the migration model of this article. The signal to noise ratio (SNR) is derived as follows:

$$SNR = 2^{R_r/B_r} - 1 \tag{12}$$

There is a mathematical relationship between the signal to noise ratio (SNR). The formula of received power P_r^{down} of the device and the noise power σ is as follows:

$$SNR = \frac{P_r^{down}}{\sigma} \tag{13}$$

Therefore, combining Formulas (12) and (13) can obtain the received power of the device P_r^{down}. The formula is as follows:

$$P_r^{down} = \sigma SNR = \sigma \left(2^{R_r/B_r} - 1\right) \tag{14}$$

Cost calculation of sub service downlink transmission energy consumption E_r^{down} is equal to the product of the downlink transmission time T_r^{down} and the received power of the mobile server P_r^{down}.

$$E_r^{down} = T_r^{down} P_r^{down} = \frac{C_{trs}}{B_r} \left[\sigma \left(2^{R_r/B_r} - 1 \right) \right] \tag{15}$$

When the execution location of the current set sub service is the mobile server, and the execution location of the post sub service is the MEC server, the transmission delay and energy consumption of the uplink data should be calculated. Uplink data transmission time is T_s^{up}. The energy consumption of uplink data transmission is E_s^{up}. Data transmission is C_{trs}. Uplink broadband is B_s. Power sent by mobile server is P_s^{up}. This paper assumes that the sub service is in a good network environment.

Cost calculation of uplink transmission time for sub services T_s^{up} is equal to the data transmission amount C_{trs} divided by the uplink broadband B_s.

$$T_s^{up} = \frac{C_{trs}}{B_s} \tag{16}$$

According to the relationship between the transmitted power of the signal and the received power, the transmit power P_s^{up} can be obtained as follows:

$$P_s^{up} = \frac{\sqrt{P_r^{down}}}{g} = \frac{P_r^{down}}{g^2} = \frac{\sigma \left(2^{R_r/B_r} - 1 \right)}{g^2} \tag{17}$$

Since the channel attenuation factor g depends on the communication distance d and the coefficient γ and satisfies $g = \gamma/d$, the expression of g is taken into Formula (17) to obtain the following:

$$P_s^{up} = \frac{\sigma \left(2^{R_r/B_r} - 1 \right) d^2}{\gamma^2} \tag{18}$$

Cost calculation of uplink transmission energy consumption of sub service E_s^{up} is equal to the uplink transmission time T_s^{up} and sending power of mobile server P_s^{up}. The energy consumption formula of uplink transmission is as follows:

$$E_s^{up} = T_s^{up} P_s^{up} = \frac{C_{trs}}{B_s} \frac{\sigma \left(2^{R_r/B_r} - 1 \right) d^2}{\gamma^2} \tag{19}$$

Transmission time of sub service T_{Trs} is equal to the uplink transmission time of end sub service computation $T_{s(n)}^{up}$, downlink transmission time of front sub service execution results $T_{r(n)}^{down}$ and the upstream transmission time of post sub service computation $T_{s(n+1)}^{up}$.

$$T_{Trs} = T_{s(n)}^{up} + T_{r(n)}^{down} + T_{s(n+1)}^{up} = \frac{C_{trs(n)}}{B_s} + \frac{C_{res(n)}}{B_r} + \frac{C_{trs(n+1)}}{B_s} \tag{20}$$

In the above formula, the subscript n is the front sub service. The subscript $n+1$ is the post sub service. $C_{trs(n)}$ denotes the computation amount of front sub service. $C_{trs(n+1)}$ denotes the computation amount of post sub service. B_s is the uplink bandwidth. $C_{res(n)}$ represents the data amount of the front sub service execution result. B_r is the uplink bandwidth.

Transmission energy consumption of sub service E_{Trs} is equal to the calculation amount of front sub service $E_{s(n)}^{up}$, downstream transmission energy consumption of end sub service execution results $E_{r(n)}^{down}$ and the calculation of post sub service $E_{s(n+1)}^{up}$.

$$E_{Trs} = E_{s(n)}^{up} + E_{r(n)}^{down} + E_{s(n+1)}^{up} = \frac{C_{trs(n)}}{B_s} P_s^{up} + \frac{C_{res(n)}}{B_r} P_r^{down} + \frac{C_{trs(n+1)}}{B_s} P_s^{up} \tag{21}$$

The total consumption time formula is as follows:

$$\begin{aligned}
&T_{Trs} = 0, \quad V_{n+1} = V_n = 1;\\
&T_{Trs} = \frac{C_{res(n)}}{B_s} + \frac{C_{trs(n+1)}}{B_s}, \quad V_n = 1 \text{ and } V_{n+1} = 0;\\
&T_{Trs} = 0, \quad V_{n+1} = V_n = 0 \text{ and in the same execution location};\\
&T_{Trs} = \frac{C_{trs(n)}}{B_s} + \frac{C_{trs(n+1)}}{B_s}, \quad n = 1 \text{ and } V_{n+1} = V_n = 0\\
&\text{and in different execution locations};\\
&T_{Trs} = \frac{C_{trs(n)}}{B_s} + \frac{C_{res(n)}}{B_r} + \frac{C_{trs(n+1)}}{B_s}, \quad V_{n+1} = V_n = 0\\
&\text{and in different execution locations};\\
&T_{Trs} = \frac{C_{trs(n)}}{B_s} + \frac{C_{res(n)}}{B_r}, \quad n = 1 \text{ and } V_n = 0 \text{ and } V_{n+1} = 1;\\
&T_{Trs} = \frac{C_{res(n)}}{B_r}, \quad V_n = 0 \text{ and } V_{n+1} = 1
\end{aligned} \tag{22}$$

The total energy consumption formula for data transmission is as follows:

$$\begin{aligned}
&E_{Trs} = 0, \quad V_{n+1} = V_n = 1;\\
&E_{Trs} = \frac{C_{res(n)}}{B_s} P_s^{up} + \frac{C_{trs(n+1)}}{B_s} P_s^{up}, \quad V_n = 1 \text{ and } V_{n+1} = 0;\\
&E_{Trs} = 0, \quad V_{n+1} = V_n = 0 \text{ and in the same execution location};\\
&E_{Trs} = \frac{C_{trs(n)}}{B_s} P_s^{up} + \frac{C_{trs(n+1)}}{B_s} P_s^{up}, \quad n = 1 \text{ and } V_{n+1} = V_n = 0\\
&\text{and in the same execution location};\\
&E_{Trs} = \frac{C_{trs(n)}}{B_s} P_s^{up} + \frac{C_{res(n)}}{B_r} P_r^{down} + \frac{C_{trs(n+1)}}{B_s} P_s^{up}, \quad V_{n+1} = V_n = 0\\
&\text{and in different execution locations};\\
&E_{Trs} = \frac{C_{trs(n)}}{B_s} P_s^{up} + \frac{C_{res(n)}}{B_r} P_r^{down}, \quad n = 1 \text{ and } V_n = 0 \text{ and } V_{n+1} = 1;\\
&E_{Trs} = \frac{C_{res(n)}}{B_r} P_r^{down}, \quad V_n = 0 \text{ and } V_{n+1} = 1
\end{aligned} \tag{23}$$

3.5. **Edge server selection strategy.**

3.5.1. *Limited selection conditions.*

1) Communication distance

The distance between MEC servers which are distributed randomly around mobile servers and mobile servers will be different. If the distance is too far, the signals will be unstable during communication. It will directly affect the data interaction, and even lead to data loss and communication interruption. Therefore, this paper sets the communication safety distance safeDis, $d \leq$ safeDis. Based on the experimental data value of [22], this paper sets the communication safety distance of mobile server receiving signals to be 750m when the mobile server is accessible.

2) Execution status of the edge server

When selecting the surrounding MEC server, it is necessary to obtain the current execution status of the server. According to the CPU computing power and memory utilization rate of the server, we can judge whether the server is in a busy state.

3) Communication signal to noise ratio

This paper specifies that the following SNR should be met during data transmission:

$$SNR \geq \frac{U_{MS\_MEC} P_s^{up}}{g \sum_{m=1}^{M} U_{MS\_MEC} P_{s(m)}^{up} + \sigma} \tag{24}$$

g is the attenuation factor of wireless channel, which conforms to the uniform distribution of $[0, 1]$. U_{MS_MEC} represents the channel power gain from the mobile server to the surrounding MEC server. P_s^{up} is the transmitting power. σ is the noise power. SNR is affected by the above variables.

3.5.2. *Energy consumption threshold calculation.* In this paper, a cost function is designed to calculate the threshold. It assumes that $M + 1$ MEC servers are randomly distributed around the mobile server. The calculation formula of cost and energy consumption is as follows:

$$V_m = \left(E_{s(m-1)}^{up} + E_{save(m-1)} + E_{r(m-1)}^{down}\right) - \left(E_{s(m)}^{up} + E_{save(m)} + E_{r(m)}^{down}\right) \tag{25}$$

Energy efficiency ϑ is defined as the average energy consumption of data transmission function. The formula is as follows:

$$\vartheta = \frac{\sum_{m=1}^{M} V_m}{M} \tag{26}$$

The threshold when selecting a new MEC server and the corresponding migration path can be described as follows:

$$V^* = \sup_{m \in M} \{E[V_m]\} \tag{27}$$

3.6. **Service migration model.** In this paper, there are three factors that determine the time required for the service to complete, time required for mobile server monitoring, service execution time, and data transmission time. The formula about them is as follows:

$$T_{total} = \sum_{n=1}^{N} \left(T_{d(n)} + T_{EX(n)} + T_{Trs(n)}\right) \tag{28}$$

The factor that determines the energy consumption for service completion E_{total} is the same as above. The formula is as follows:

$$E_{total} = \sum_{n=1}^{N} \left(E_{d(n)} + E_{EX(n)} + E_{Trs(n)}\right) \tag{29}$$

In this paper, different weights are given to the delay and energy consumption indicators. The sum of the two values has the minimum, which is the migration strategy of maximum revenue. The delay weight of the sub service is ω_t. Energy consumption weight of sub services ω_e depends on the specific needs of the current sub service. The formula of the migration model in this paper is as follows:

$$Q_n = Min(\omega_t T_{total} + \omega_e E_{total}) \tag{30}$$

3.7. **Normalized calculation.** As mentioned above, service migration is optimized for both energy consumption and execution delay. The paper uses the method of vector normalization to obtain the value of the objective function. The normalization formula is as follows:

$$y_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{n} x_{ij}^2}} \tag{31}$$

In the above formula, x_{ij} refers to the attribute value of the jth attribute of the migration decision strategy of the ith group. j only takes two values "1" and "2" to represent energy consumption and time delay respectively.

4. **Model Solving.**

4.1. **Service migration model implementation.** The implementation process of edge server selection algorithm is as follows.

1) Initialization parameters are obtained. Calculation amount of service is C. The number of optional MEC servers is $m+1$. Wireless channel attenuation factor is g. Noise power is σ. Communication coefficient is γ. Network bandwidth is B.

2) The model gets the location information of the mobile server and the kth MEC server. The model calculates the distance d between the edge server and the mobile server, and obtains the CPU clock frequency of the edge server.

3) The SNR is calculated by Formula (12). Then the data transmission power between MEC server and MS P_s^{up} is calculated by SNR. Finally, the initialization parameters and P_s^{up} are used to calculate the comparison condition of SNR.

4) Whether the limited selection conditions are met is determined. If the conditions are met, continue to execute step 5). Otherwise continue to monitor the next edge server and return to step 2).

5) The mobile server obtains the running status of MEC. When the running status is not busy and the performance is good, execute step 6). Otherwise continue to monitor the next MEC and return to step 2).

6) According to Formula (19) and Formula (15), the energy consumption of uplink transmission E_s^{up} and downlink transmission energy consumption E_r^{down} are calculated. According to Formula (6), the energy consumption of the mobile server E_{save} is calculated when the sub service is executed on the current edge server. Finally, the above parameters are substituted into Formula (25) to calculate the cost energy consumption value V_M.

7) Number of the current edge servers, E_s^{up}, E_r^{down}, E_r^{down}, path length corresponding to current edge server d and cost energy consumption value V_M are saved.

8) The cost and energy consumption value V_M of all edge servers are calculated. The optimal expected value V^* is calculated according to Formula (27), energy consumption threshold.

9) The sum of transmission energy consumption and energy saving of each edge server is compared with the energy consumption threshold V^*. When the sum of the first energy consumption is less than the threshold V^*, the edge server corresponding to the sum of current energy consumption is selected as the target migration server of the current sub service.

10) If there is no energy consumption less than V^* between all edge servers and mobile servers, the MEC server that meets the constraints and is closest to the mobile server is selected as the target migration server of the current sub service.

4.2. **Solution of mobile service migration model based on improved genetic algorithm.**

4.2.1. *Improvements to genetic algorithms.*

1) Initial population construction based on reverse learning mechanism

Aiming at the problem of low convergence accuracy and slow convergence speed, the traditional method of generating initial population of genetic algorithm is to generate solutions randomly in the solution space and combine them as the initial population. This paper adopts the initial population construction strategy based on reverse learning mechanism. In the algorithm, the positive solution and the reverse solution corresponding to each positive solution are searched at the same time. The fitness function is successively substituted to select the solution with better fitness value as the initial population. In this way, the probability of finding the best individual is improved and the performance of the algorithm is improved.

2) Using Levy flight mechanism to solve local optimal problem

It is easy to get into the local optimal solution. In order to ensure the robustness of the algorithm and prevent premature convergence, this paper embeds Levy flight mechanism into the crossover process of the algorithm. When the traditional genetic algorithm performs crossover operation, the individuals of the population are crossed in order. And the search space is slightly small. While Levy flight core is a random step according to normal distribution. The step size is small. There are both short-distance local walk and occasional large step jump. In other words, it has not only the ability of local fine search, but also the ability to jump out of local optimum.

In order to reflect the advantages of Levy flight, the motion laws of Levy flight and Brownian motion are compared for 50 and 100 times respectively. The motion laws of Levy flight and Brownian motion are shown in Figure 3.

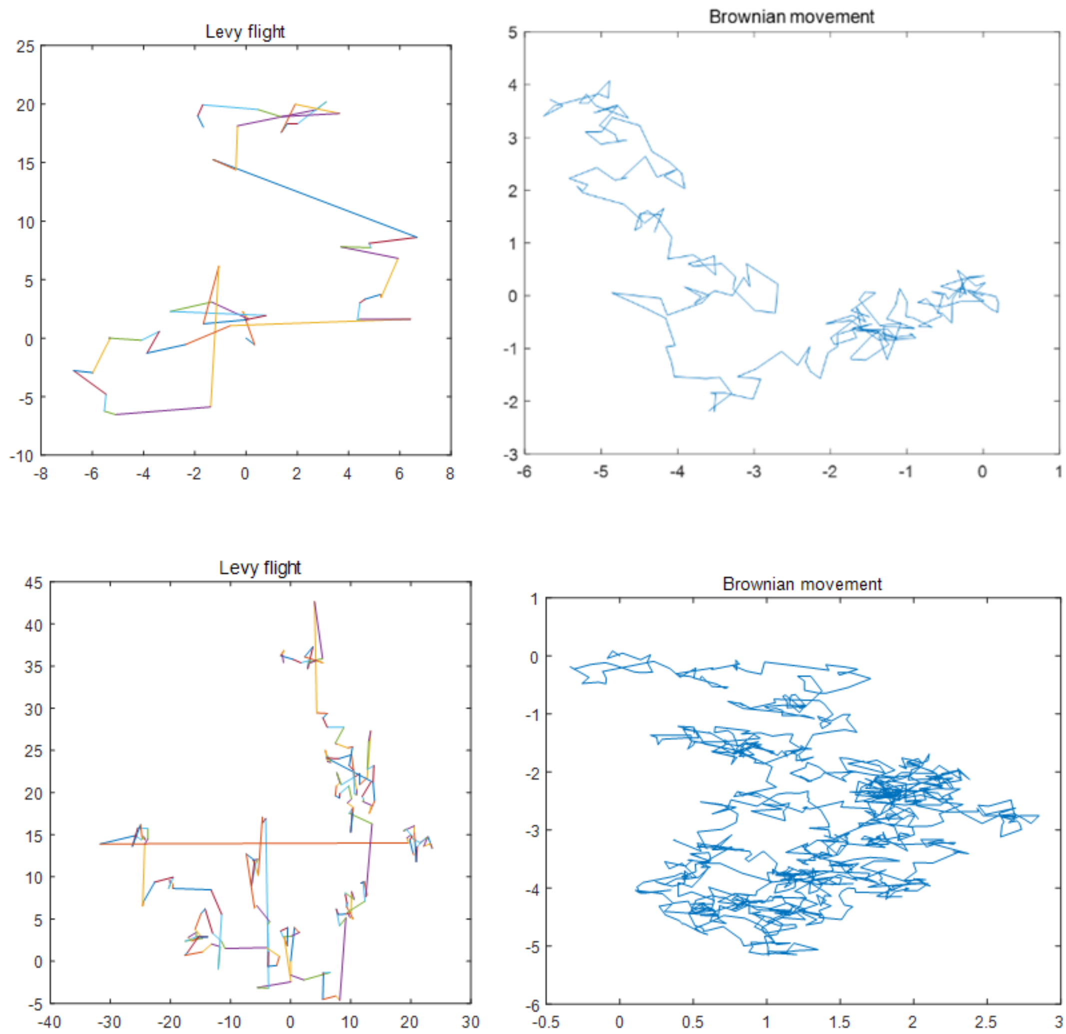

FIGURE 3. Levy flight and Brownian movement

4.2.2. *Solution process of mobile service migration model based on improved genetic algorithm.* Genetic algorithm has been widely used in the field of computer. Santoso et al. [23] have applied genetic algorithm to neural network modelling for time series. Genetic algorithm has been improved in this paper, in which the main purpose is to find the optimal DNA individual. It can be named the optimal service migration strategy.

1) Construction of fitness function

Generally, the fitness function is constructed based on the objective function. The objective function set in this paper should take the minimum value. The solution with high fitness value should be selected in the algorithm. Therefore, this paper needs to construct the fitness function into the form that the smaller the objective function value is, the higher the fitness function value is. The fitness function is shown below.

$$Fitness = 1 - \frac{Q_n}{\sum_{n=1}^{N} Q_n} \tag{32}$$

In the above formula, Q_n is the value of the objective function. $\sum_{n=1}^{N} Q_n$ is the sum of all individual objective functions in the population. In this paper, the minimum sum of energy consumption and delay weights is taken as the optimization objective. Thus, the higher the fitness value of DNA individuals in the algorithm is, the smaller the corresponding objective function value is.

2) The edge selection algorithm

It is necessary to call the edge server selection algorithm based on energy consumption threshold to select the corresponding target migration server for each sub service. The server number is k_n. The resource information of the mobile server and the selected edge server is obtained. The resource information of the server and the value of each gene bit are substituted into the objective function for calculation.

3) Initial population generation process

In this paper, the time complexity of the algorithm is reduced. Firstly, $2^{N/2}$ (N is the number of labeled sub services) initial solutions x_i ($i \in 2^{N/2}$) generated randomly in the solution space $[a, b]$ form a positive solution space, such as Formula (2). And then Formula (1) is used to find the corresponding inverse solution x for each initial solution x_i^* ($i \in 2^{N/2}$) forms the reverse solution space, such as Formula (5). The values of the two solution spaces are converted into binary form and substituted into Formula (32). Through comparison, $2^{N/2}$ solutions with the best fitness value are selected as the final initial population. '0' and '1' forms of each body in the population represent the service migration strategy variable V_n. '0' is migrated to the edge server for calculation. And '1' is for calculation on the mobile server.

4) Selection operation

The selection operator in this paper adopts the best selection algorithm which is preserved. The concept of "elite retention strategy" is to copy the best individuals (elite individuals) in the process of population evolution directly to the next generation without pairing, crossover and mutation.

5) Cross operation

In this paper, a single point crossover algorithm based on Levy flight mechanism is adopted. The crossover operator in traditional genetic algorithm is to cross two chromosomes as a pair of individuals in the population, which is easy to cause the problem of too small search space and far away from the optimal individual. The Levy flight mechanism is a Markov process. It can increase the search step size, expand the search range, and increase the diversity of the population. It ensures that the search will not fall into local optimum.

The walking step length of Levy flight satisfies a heavy tail Levy distribution, such as Formula (9), $L(step) \sim \frac{\beta \tau(\beta)\sin(\pi\beta/2)}{\pi |step|^{1+\beta}}$, $0 < \beta \leq 2$, $step \to \infty$. Using the step size obtained, such as Formula (12), $step = \frac{\mu}{|\varphi|^{1/\beta}}$, $1 < \beta \leq 2$, the random chromosome number 'index' in the algorithm is replaced.

After the cross chromosome is obtained by searching, the gene crossover point of a chromosome is generated randomly. So the gene sites after the crossover point in a pair of chromosomes are crossed and exchanged, and the gene before the gene position can be copied. Two new chromosomes were obtained and combined into a new group of new species, as shown in Figure 4.

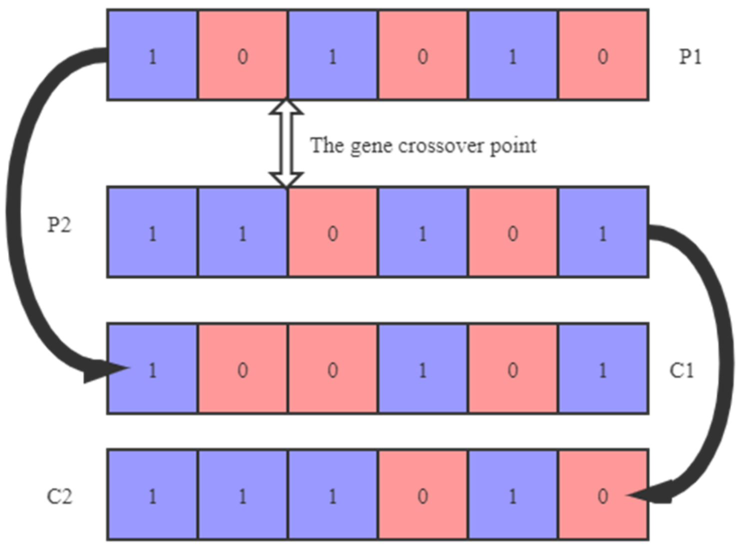

Figure 4. Cross process

6) Mutation operation

The mutation operator used in this paper is the basic bit mutation algorithm. The population size generated in this paper is $2^{N/2}$. The number of genes in each chromosome is N. So the total number of genes in the population is $2^{(N/2)} * N$. Then, the number of mutation genes is calculated by using the probability of variation. Finally, the mutation loci are randomly generated, and the genes of gene loci are mutated to generate the next generation of new species group, as shown in Figure 5.

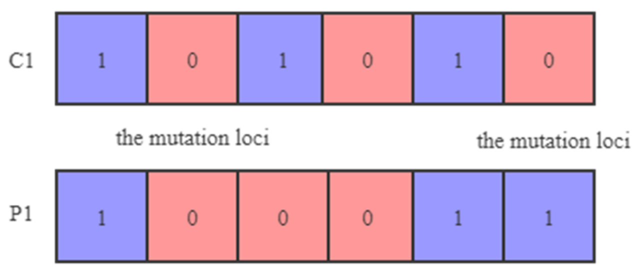

Figure 5. Mutation process

7) Adjust crossover probability and mutation probability

The values of crossover probability and mutation probability in this paper vary with the individual fitness of the population. In the crossover operation, if the fitness value of the two chromosomes to be crossed is higher than the average fitness value of the current population, the crossover probability should be reduced to maintain a better individual, which can accelerate the convergence of the algorithm. Cross probability P_c and selection probability P_m are as follows.

$$P_c = \begin{cases} P_{c1}, & f > f_{\text{avg}} \\ P_{c1} - \dfrac{(P_{c1} - P_{c2})(f_{\text{avg}} - f)}{(f_{\text{avg}} - f_{\min})}, & f \leq f_{\text{avg}} \end{cases} \quad (33)$$

$$P_m = \begin{cases} P_{m1}, & f^* > f_{\text{avg}} \\ P_{m1} - \dfrac{(P_{m1} - P_{m2})(f_{\text{avg}} - f^*)}{(f_{\text{avg}} - f_{\min})}, & f^* \leq f_{\text{avg}} \end{cases} \quad (34)$$

In Formula (33) and Formula (34), P_{c1}, P_{c2} $(0.0 < P_{c1} < P_{c2} < 1.0)$ denote the minimum and maximum values of the crossover probability, in which the specific value range has been given in the initialization stage.

5. Results and Analysis.

5.1. Experimental environment.

The network environment of this experiment is a laboratory local area network. 3G and 4G cellular networks are not supported at this time. Model parameter table is shown in Table 1.

TABLE 1. Model parameter table

Variable	Meaning	Value
F_{MS}	MS's CPU clock frequency	real-time acquisition
P_{MS}	MS's execution power	0.6W
P_{slp}	MS's sleep power	0.02W
F_{MEC}	MEC's CPU clock frequency	real-time acquisition
B_s	Uplink broadband	real-time acquisition
B_r	Downlink broadband	real-time acquisition
C_{MEC}	Service calculation	$(1 \sim 20) * 2 * 10^4$bit
C_{MS}	Service calculation	$(1 \sim 20) * 2 * 10^4$bit
C_{trs}	Data transfer volume	$(1 \sim 20) * 2 * 10^4$bit
R_r	Data migration rate	$(1 \sim 20) * 2 * 10^3$bps
E_d	Monitoring energy consumption	$\{0.1, 1\}$J
d_x	Communication distance between MS and MEC	real-time acquisition
σ	Signal noise power	3
U_{MS_MEC}	Channel power gain	$0 \sim 4$
ω_t/ω_e	Delay/Energy Weight	$\{0.3, 0.5, 0.7\}$

5.2. Comparison of algorithm results.

In this section, in order to test the advantages and disadvantages of the algorithm, it is compared with other selection algorithms. The comparison algorithm is as follows.

1) It is compared with the strategy based on the Shortest Node Distance (SMSND) in [24]. The idea of the algorithm is to select the migration MEC which is closest to the mobile server as the migration node.

2) It is compared with the Dynamic Service Migration Strategy based on Immediate Execution (DAMIE) in [25]. The idea of this algorithm is that when the mobile server monitors the distance between the mobile server and the mobile server, $d < K$ (k is the maximum distance of the mobile server to receive signals), the current mobile server immediately migrates the service with the execution state to the monitored MEC.

Secondly, it evaluates the astringency of integrate reverse backward learning and Levy flight genetic algorithm (BL-LFGA) which this work proposed. By comparing with other algorithms, the performance of this algorithm is verified.

1) It is compared with the hierarchical genetic particle (HGP) swarm optimization algorithm in [26]. Firstly, the genetic algorithm is used to generate a group of better individuals, and then the group of individuals is used as the initial population of particle swarm optimization algorithm for optimization.

2) It is compared with the integer particle swarm optimization (IPSO) in [27]. The idea is to express the position of the particle as a vector in the integer space, and the velocity as a vector in the real space. The position of the particle is obtained by rounding the actual position value to the nearest integer value.

This paper compares and analyzes the strategies from two aspects, data transmission energy consumption and selection time.

The first is data transmission energy consumption. As shown in Figure 6, assuming that there are 10-18 edge servers randomly distributed around the mobile server, this paper compares the energy consumption of the edge servers selected by the three strategies for different number of edge servers to determine the advantages and disadvantages of the strategies.

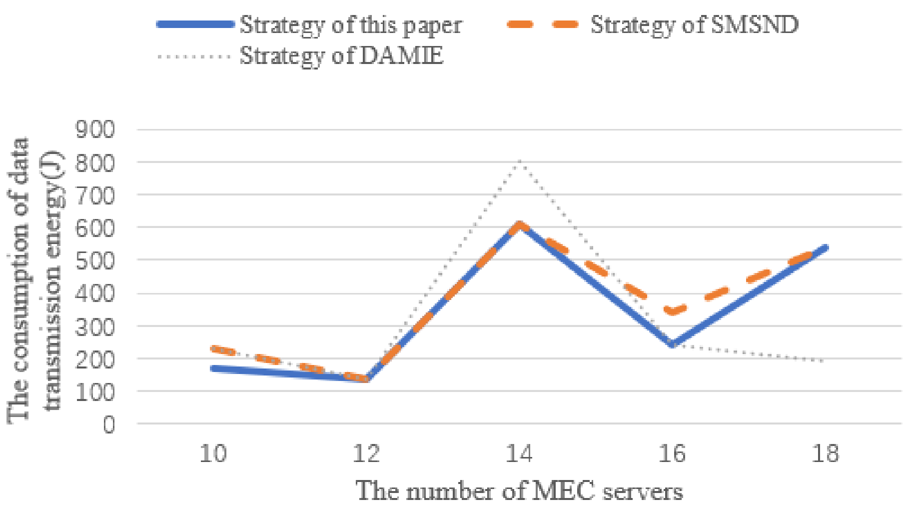

FIGURE 6. Comparison of the consumption of data transmission energy of different strategies

The second is selection time. The size of selection time is a good indicator to measure the performance of selection strategy. For mobile applications, the timeliness requirement is very high. If the selection efficiency is low, it will cause stuck or data loss, which will seriously affect the user experience. Therefore, the lower the delay of the selection, the higher the timeliness of the service, as shown in Figure 7.

And then we first show the performance of BL-LFGA. Figure 8 and Figure 9 show the comparison of average response time and objective function values of the three algorithms under different number of sub services. It can be seen from the figure that the average response time of IPSO and BL-LFGA is slightly faster and the time complexity is lower when the number of sub services is large. HGP mainly combines genetic algorithm and particle swarm optimization algorithm, and the time complexity is relatively higher. And the value of the optimized objective function is indeed of reference value. It combines the advantages of global search of genetic algorithm and local search of particle swarm

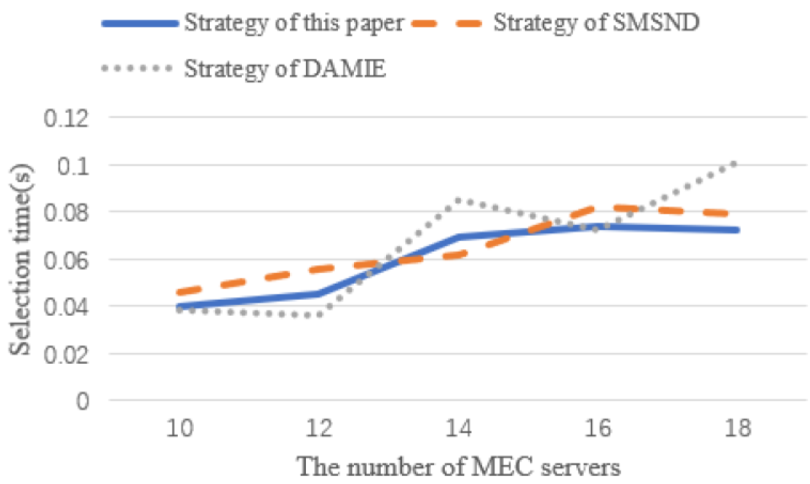

FIGURE 7. Comparison of different strategy selection time

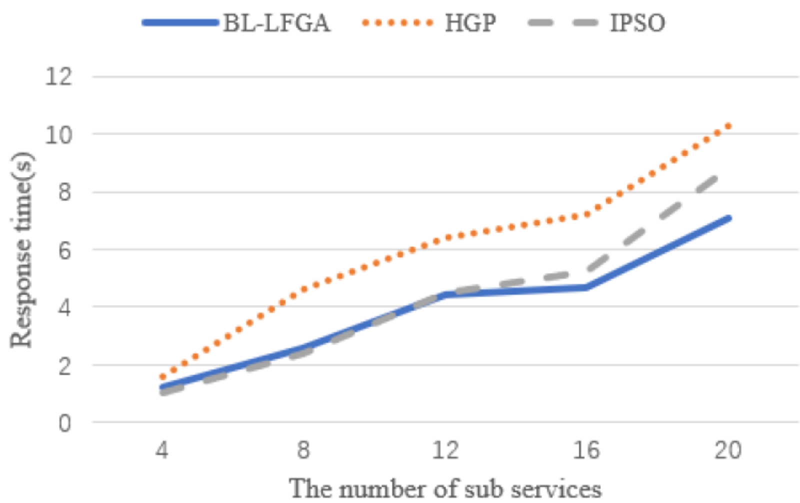

FIGURE 8. Comparison of algorithm response time

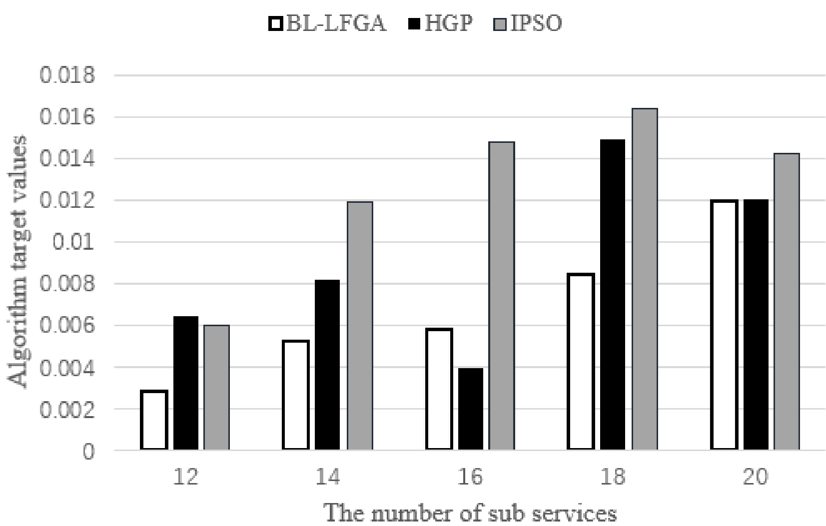

FIGURE 9. Comparison of algorithm target values

optimization; it can be seen from Figure 9 that the value of this algorithm is slightly close to that of HGP. It highlights the superiority of the algorithm.

Secondly, the migration strategy of this paper, one of the purposes is to reduce energy consumption during the execution of compute-intensive services. Comparison of energy consumption of different strategies is shown in Figure 10.

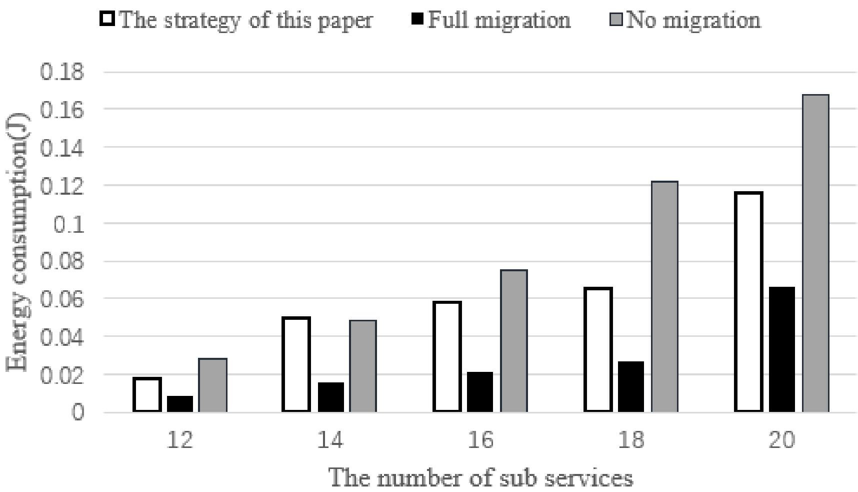

FIGURE 10. Comparison of energy consumption of different strategies

6. **Conclusions.** In this paper, we construct a service migration model with joint energy consumption and delay for mobile service migration, and transform the problem into a nonlinear 0-1 programming problem. We design an improved genetic algorithm with reverse learning and Levy flight mechanism and a threshold-based edge server selection strategy. The convergence and performance of the improved genetic algorithm are analyzed by comparing with the algorithms involved in the research status. The migration strategy obtained by the algorithm reduces the energy consumption and communication delay of the mobile server. The advantages and disadvantages of the proposed strategy are compared with other two strategies. Experiments show that the proposed strategy improves the service migration efficiency.

Acknowledgment. This work is supported by the National Key R&D Program of China (Grant No. 2018YFB1601502) and the Liaoning Revitalization Talents Program (Grant No. XLYC1902071).

REFERENCES

[1] S. Wang, Edge computing: Applications, state-of-the-art and challenges, *Advances in Networks*, vol.7, no.1, pp.8-15, 2019.

[2] C. Wang, Y. He, F. R. Yu, Q. Chen and L. Tang, Integration of networking, caching, and computing in wireless systems: A survey, some research issues, and challenges, *IEEE Communications Surveys and Tutorials*, vol.20, no.1, pp.7-38, 2018.

[3] A. Khan, M. Othman, S. Madani and S. U. Khan, A survey of mobile cloud computing application models, *IEEE Communications Surveys and Tutorials*, vol.16, no.1, pp.393-413, 2014.

[4] P. Mach and Z. Becvar, Mobile edge computing: A survey on architecture and computation offloading, *IEEE Communications Surveys and Tutorials*, vol.19, no.3, pp.1628-1656, 2017.

[5] Y. Yu, Mobile edge computing towards 5G: Vision, recent progress, and open challenges, *China Communications*, vol.13, no.Supplement2, pp.89-99, 2016.

[6] X. Shao et al., Editorial: Recent Advances on intelligent mobility and edge computing, *Mobile Networks and Applications*, 2021.

[7] K. Peng et al., Joint optimization of service chain caching and task offloading in mobile edge computing, *Applied Soft Computing*, vol.103, 2021.
[8] Y. Gao, W. Tang, M. Wu, P. Yang and L. Dan, Dynamic social-aware computation offloading for low-latency communications in IoT, *IEEE Internet of Things Journal*, vol.6, no.5, pp.7864-7877, 2019.
[9] F. Wei, S. Chen and W. Zou, A greedy algorithm for task offloading in mobile edge computing system, *China Communications*, vol.15, no.11, pp.149-157, 2018.
[10] J. Plachy, Z. Becvar and P. Mach, Path selection enabling user mobility and efficient distribution of data for computation at the edge of mobile network, *Computer Networks*, vol.108, pp.357-370, 2016.
[11] W. Fan, J. Han, L. Yao, F. Wu and Y. Liu, Latency-energy optimization for joint WiFi and cellular offloading in mobile edge computing networks, *Computer Networks*, vol.181, 2020.
[12] Y. Li and C. Jiang, Distributed task offloading strategy to low load base stations in mobile edge computing environment, *Computer Communications*, vol.161, pp.240-248, 2020.
[13] L. Jiang, L. Liu, J. Yao and L. Shi, A user interest community evolution model based on subgraph matching for social networking in mobile edge computing environments, *Journal of Cloud Computing*, vol.9, no.1, 2020.
[14] F. Mashhadi, S. Salinas, A. Bozorgchenani and D. Tarchi, Optimal auction for delay and energy constrained task offloading in mobile edge computing, *Computer Networks*, vol.183, 2020.
[15] S. Sardellitti, G. Scutari and S. Barbarossa, Joint optimization of radio and computational resources for multi-cell mobile cloud computing, *2014 IEEE 15th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, Toronto, ON, Canada, 2014.
[16] W. Labidi, M. Sarkiss and M. Kamoun, Joint multiuser resource scheduling and computation offloading in small cell networks, *2015 IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, Abu Dhabi, United Arab Emirates, 2015.
[17] C. F. Liu, M. Bennis and H. V. Poor, Latency and reliability-aware task offloading and resource allocation for mobile edge computing, *IEEE Globecom Workshops (GC Wkshps)*, 2017.
[18] X. Meng, W. Wang, Y. Wang, V. K. N. Lau and Z. Zhang, Closed-form delay-optimal computation offloading in mobile edge computing systems, *IEEE Transaction on Wireless Computations*, vol.18, no.10, pp.4653-4667, 2019.
[19] M. Chen and Y. X. Hao, Task offloading for mobile edge computing in software defined ultradense network, *IEEE Journal on Selected Areas in Communications*, vol.36, no.3, pp.587-597, 2018.
[20] X. Ding, W. Zhang and Q. Zhu, Computing unloading strategy of massive Internet of Things devices based on game theory in mobile edge computing, *Mathematical Problems in Engineering*, 2021.
[21] W. Zhang, B. Guo, Y. Shen, Y. Wang, W. Xiong and L. Duan, Computing offloading on intelligent mobile terminal, *Chinese Journal of Computers*, vol.39, no.5, pp.1021-1038, 2016.
[22] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan and K. K. Leung, Dynamic service migration in mobile edge-clouds, *IFIP Networking Conference*, Toulouse, France, 2015.
[23] R. Santoso, B. Warsito and H. Yasi, Graphical interface of genetic optimization in neural network modelling for time series, *ICIC Express Letters, Part B: Applications*, vol.12, no.3, pp.301-306, 2021.
[24] Y. Mao, J. Zhang and K. B. Letaief, Dynamic computation off-loading for mobile-edge computing with energy harvesting devices, *IEEE Journal on Selected Areas in Communications*, vol.34, no.12, pp.3590-3605, 2016.
[25] Y. Qi, Y. Zhou, L. Liu, L. Tian and J. Shi, MWC coordinated future 5G mobile wireless networks, *Journal of Computer Research and Development*, vol.55, no.3, pp.178-186, 2018.
[26] F. Guo, H. Zhang, H. Ji, X. Li and V. C. M. Leung, An efficient computation offloading management scheme in the densely deployed small cell networks with mobile edge computing, *IEEE/ACM Transactions on Networking*, vol.26, no.6, pp.2651-2664, 2018.
[27] J. Liu and Q. Zhang, Code-partitioning offloading schemes in mobile edge computing for augmented reality, *IEEE Access*, pp.11222-11236, 2019.