

## Research Article

# Workflow Scheduling Based on Mobile Cloud Computing Machine Learning

**Fanghai Gong** 

*Guangdong Lingnan Institute of Technology, Guangzhou, 510663 Guangdong, China*

Correspondence should be addressed to Fanghai Gong; 1311410609@st.usst.edu.cn

Received 29 March 2021; Revised 21 May 2021; Accepted 3 June 2021; Published 6 July 2021

Academic Editor: Wenqing Wu

Copyright © 2021 Fanghai Gong. 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.

In recent years, cloud workflow task scheduling has always been an important research topic in the business world. Cloud workflow task scheduling means that the workflow tasks submitted by users are allocated to appropriate computing resources for execution, and the corresponding fees are paid in real time according to the usage of resources. For most ordinary users, they are mainly concerned with the two service quality indicators of workflow task completion time and execution cost. Therefore, how cloud service providers design a scheduling algorithm to optimize task completion time and cost is a very important issue. This paper proposes research on workflow scheduling based on mobile cloud computing machine learning, and this paper conducts research by using literature research methods, experimental analysis methods, and other methods. This article has deeply studied mobile cloud computing, machine learning, task scheduling, and other related theories, and a workflow task scheduling system model was established based on mobile cloud computing machine learning from different algorithms used in processing task completion time, task service costs, task scheduling, and resource usage. The situation and the influence of different tasks on the experimental results are analyzed in many aspects. The algorithm in this paper speeds up the scheduling time by about 7% under a different number of tasks and reduces the scheduling cost by about 2% compared with other algorithms. The algorithm in this paper has been obviously optimized in time scheduling and task scheduling.

## 1. Introduction

With the widespread popularization and application of Internet technology, as well as the rapid growth of information, the data that scientific research and business need to face and process has become increasingly large and complex, far exceeding the computing power of the existing IT infrastructures. In order to solve the problem of large-scale and massive processing, the concept of cloud computing is proposed. Cloud computing is a new resource delivery and service provision model. Cloud computing service providers can transfer various software and hardware resources to the cloud computing environment to provide users with abundant computing resources and computing services, such as large-scale scientific computing services and data storage services. At this point, users only need to dynamically select the appropriate resources according to their own needs and pay a certain fee to the cloud service provider, avoiding the need for users to buy large-scale software and hardware equipment

(saving resource investment costs) or to develop specific applications for time overhead.

In recent years, the application range of cloud computing has become wider and wider, covering all aspects of our lives. With the continuous improvement of cloud computing processing task capability, people's requirements for hardware have gradually changed from pure device functionality to service quality claim. For example, service cost, time, flexibility, security, scalability, and reliability are all factors that need to be considered in cloud computing. This is also the main reason why cloud computing has become a research hotspot today.

According to Tawalbeh et al., mobile devices are increasingly becoming an indispensable part of people's daily lives and are conducive to performing various useful tasks. Mobile cloud computing integrates mobile and cloud computing to expand its functions and benefits, overcoming their limitations, such as limited memory, CPU power, and battery life. Big data analysis technology can extract values from data. Data has four aspects: volume, change, speed, and accuracy.

Tawalbeh et al. discussed online healthcare and the role of mobile cloud computing and big data analysis in its implementation. With the application of cloud computing in the field of healthcare, the development and momentum of networked medical applications and systems have also emerged. They reviewed the techniques, tools, and applications of big data analysis. Finally, the research results of using big data and mobile cloud computing technology to design networked medical systems are summarized. The prospects for networked medical care are put forward. However, their research results did not actually solve the application problems in related fields, and there are still many problems in actual operations [1]. Buczak and Guven introduced a centralized literature review of machine learning (ML) and data mining (DM) methods for network analysis to support intrusion detection. A short tutorial description of each ML/DM method is provided. Based on the number of citations or the relevance of emerging methods, they identified, read, and summarized papers that represent each method. Since data is very important in ML/DM methods, they introduced some well-known network data sets for ML/DM. Then, they discussed the complexity of ML/DM algorithms, discussed the challenges of using ML/DM algorithms in network security, and put forward some suggestions on when to use a given method. However, they did not innovate in this area nor did they combine with other applications to explore practical applications [2]. According to Masdari et al., workflow scheduling is a prominent problem in cloud computing. Its goal is to complete the execution of the workflow by considering the service quality requirements of the workflow (such as deadlines and budget constraints). Aiming at the simple and scientific workflow scheduling problem in cloud computing, many latest workflow scheduling schemes have been proposed in the literature, and they have conducted a comprehensive review and analysis of these schemes. They clarified the goals of scheduling schemes in cloud computing, and they classified the proposed schemes according to the type of scheduling algorithm applied in each scheme. In addition, each program is explained and a comprehensive comparison is made to highlight their goals, characteristics, and limitations. Finally, the conclusions and future research directions are put forward. However, their research plan only considers a single effect, and they did not conduct a combined study of multiple goals [3, 4].

The innovations of this article are as follows: (1) We combined qualitative research with quantitative research and fully analyzed the research data. (2) We combined theoretical research with empirical research based on mobile cloud computing and machine learning theory, and then we investigated based on the specific situation of workflow task scheduling.

## 2. Research Method of Workflow Scheduling Based on Mobile Cloud Computing Machine Learning

### 2.1. Cloud Computing

*2.1.1. Cloud Computing Overview.* Cloud computing is developed on the basis of utility computing, distributed comput-

ing, and virtualization. It can integrate different software and hardware resources into a large “resource pool” through virtualization technology [5], and users can use the Internet Visit as the “resource pool” and then purchase these computing resources according to their own needs [6]. At present, there is no unified definition of cloud computing. Many research scholars and institutions have defined cloud computing according to their own understanding.

According to the National Institute of Standards and Technology (NIST), cloud computing is defined as a kind of computing resource (including networks, servers, storage, applications, and services) that can be obtained through the Internet in a convenient and on-demand manner. These resources come from a configurable and shared resource pool [7], and they only need very little management effort from the users and enable users to quickly configure and release related computing resources without interacting with cloud providers [8]. In summary, cloud computing can also be called grid computing; it can complete the processing of tens of thousands of items of data in a short period of time (a few seconds), so as to achieve powerful network services, and through a system composed of multiple servers, it can process and analyze these small programs to get the results and return them to the user.

*2.1.2. Features of Cloud Computing.* The concept of cloud computing is constantly changing with the development of technology. However, regardless of the future development of cloud computing, cloud computing will have the following major characteristics:

- (1) Virtualization [9]: cloud computing can integrate many different hardware physical resources into a large virtual resource pool through virtualization technology, which can be managed uniformly by cloud computing. Users can purchase various services through the Internet and then use various terminal devices anytime and anywhere, without knowing the specific location of the resources used [10]
- (2) High scalability: cloud computing can realize the scalability of IT resource utilization. Users can dynamically purchase resources from cloud service providers according to their own needs. When users do not need redundant computing resources, they can also release them in time [11]. Cloud service provider Yi can provide more virtualized resources to meet the needs of different users to obtain more users
- (3) Transparency and pooling of resources: the transparency of resources is aimed at users. Users do not need to understand the internal structure of cloud computing resources [12], but only need to pay attention to whether their own needs are met. Pooling is for cloud providers. In the cloud computing environment, the cloud provider virtualizes all resources into a “virtual resource pool,” and then it performs unified management and scheduling of virtual resources [13] and provides corresponding services according to the different requirements of users

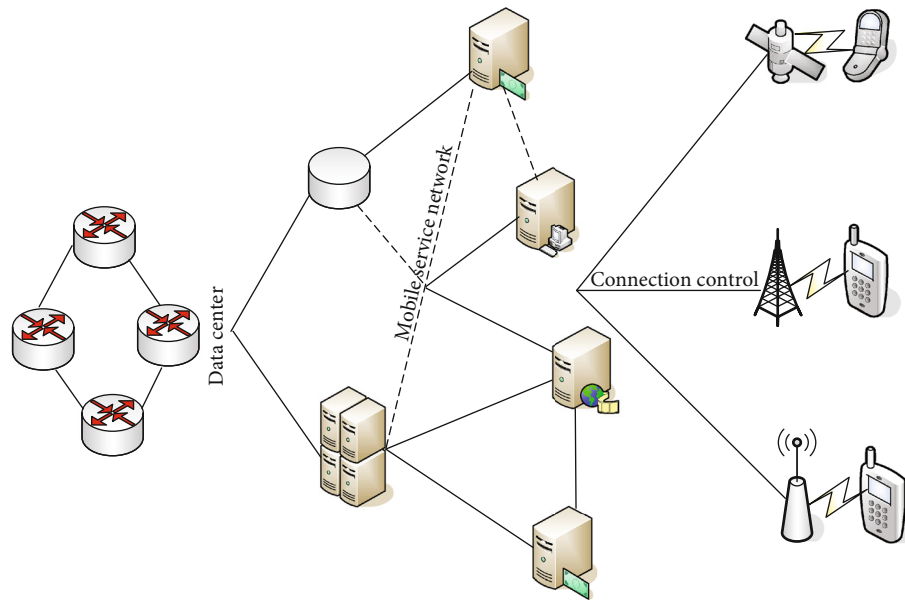


FIGURE 1: Mobile cloud computing structure.

- (4) The scale is huge: the scale of cloud computing is quite large. For example, Google's cloud computing platform is built with more than 1 million servers. Companies such as IBM, Microsoft, and Amazon are also building cloud computing platforms through spectacular physical computer clusters [14, 15] and providing users with super computing capabilities and storage capabilities
- (5) Pay on-demand: users can pay corresponding fees based on the resources used, which is the same as the payment method for public infrastructure such as electricity, water, and natural gas
- (6) Reliable automatic management: the cloud computing data center is in an unsupervised state, realizing high-reliability automated management [16]. The data can be backed up in time, and the failed nodes can be automatically detected and eliminated without affecting the normal operation of the system, thereby avoiding server overload, reducing the waste of resources, and ensuring that cloud computing can provide users with stable, safe, and sustainable service [17]

**2.1.3. Cloud Computing Classification.** There are many forms of cloud computing. According to the scope of services provided by cloud computing, it can be divided into the public cloud, the private cloud, and the hybrid cloud.

The public cloud is a platform for several enterprises and public users. Cloud service providers usually provide users and enterprises with related services through the Internet, such as storage, computing, and other types of services [18]. Enterprises or users do not need to build and manage software and hardware platforms by themselves, but they only

need to pay on-demand to easily obtain cloud services. The private cloud is the opposite of the public cloud. It is mainly for enterprise users. It is a cloud computing platform independently built by enterprises or organizations, and it only provides resources and services for internal users of enterprises or organizations [19]. And compared with public clouds, private clouds are more secure. The hybrid cloud is formed by the integration of the public cloud and the private cloud. This is the development direction of cloud computing in recent years. For some enterprises, considering the security of data, they are more willing to store data in a private cloud. At the same time, they want to obtain public cloud computing resources [20, 21]. Therefore, the hybrid cloud is adopted by more and more enterprises. Because the hybrid cloud makes full use of the advantages of both public and private clouds, it maximizes enterprise benefits. Figure 1 shows the structure of mobile cloud computing.

**2.2. Neural Network.** A neural network is a commonly used method in artificial intelligence, which mainly imitates the working principle of a human neural network to process information. A neural network is an arithmetic model composed of a large number of nodes, and each layer of nodes is connected with each other. The output of each node of the neural network needs a transformation function to limit it. In addition, the nodes between two adjacent layers are connected by a variable called weight, which is equivalent to the memory of a human neural network [22, 23].

A neural network has a strong information synthesis capability, it can process qualitative and quantitative information at the same time, it can coordinate well this relationship, and it is suitable for processing complex nonlinear and uncertain objects. The design of a neural network has strong plasticity, so it can realize self-learning, self-organization, and self-adaptation, and it can conveniently deal with

uncertain systems. Since the information in the neural network is distributed and stored in the neurons of the neural network, it has strong robustness and fault tolerance. Each neuron of the neural network is structurally parallel, which can increase the calculation speed during design.

**2.3. Workflow Scheduling.** Workflow is a business process that is automatically executed according to a series of defined rules. Documents and tasks are transferred between different actors and completed by multiple participants. In the process of workflow execution, the workflow management system executes and manages the workflow through computer technology and coordinates the information interaction between members or jobs [24].

The traditional workflow system allows users to build visual applications without complicated and time-consuming programming. However, the disadvantage of the workflow system is that it cannot easily access the software components, which prevents the workflow system from being widely used [25]. It happens that cloud computing provides a service for accessing resource pools and software and hardware, and therefore, the workflow system has been well developed. Among them, the cloud workflow task scheduling algorithm is one of the core technologies of the cloud workflow system. The quality of its scheduling strategy will directly affect the performance of the cloud workflow system.

With the continuous development of cloud computing, distributed workflow systems have gradually evolved to the direction of cloud computing workflow systems [26]. This is a brand-new application mode produced by the workflow management system in the cloud computing environment, which is referred to as the cloud workflow for short. The cloud workflow management system uniformly manages the computation or storage of resources and realizes the orderly and efficient execution of cloud workflow task scheduling [27], thereby realizing the automation of business processes.

The cloud workflow has the following characteristics:

- (1) **Transparency:** in the cloud computing environment, all resources are virtualized. The operating environment, operating system, and implementation language of all services in the cloud can be the same or different. The user does not need to know the internal implementation structure of the service used, let alone where the resource used is located
- (2) **Scalability:** cloud resources have the characteristics of on-demand allocation, and users can purchase the resources they need through the Internet anytime, anywhere. Users can also release redundant resources to reduce the cost of renting resources [28]. This dynamic resource management method enables efficient execution of workflow tasks. In addition, this scalability not only enables users to adapt to computing resources but also enables cloud service providers to maximize their use of resources and maximize their own benefits

- (3) **Real-time monitoring:** in the cloud computing environment, the monitoring management module can realize resource load balancing, fault monitoring, and node scale control by monitoring the running status of cloud workflow tasks [29]

In addition to some of the above features, the cloud workflow also includes its strong security features, and it can implement flexible out-of-office and overtime management strategies, which are in line with the flexible task configuration requirements required by the cloud workflow. All in all, the cloud workflow is an optimized solution for cloud computing systems that require flexible configuration, automatic task scheduling, optimized management of resources, and computing process. The cloud workflow can not only compress the cost of cloud computing but also improve the quality of cloud services. Therefore, the cloud workflow is bound to receive more and more attention, and it will develop rapidly, thereby promoting the development of the entire cloud computing industry.

### **3. Workflow Scheduling Research System Model Based on Mobile Cloud Computing Machine Learning**

**3.1. Research Goals of Workflow Scheduling Based on Cloud Computing.** For the goal optimization problem of cloud workflow task scheduling, most of them pay attention to the following QOS indicators:

- (1) **Execution time:** it represents the time required for the completion of cloud workflow tasks, and it is the most important goal in the scheduling strategy. For users, if the submitted cloud workflow tasks have an urgent need for completion time, they only need to map them to computing resources with good performance, which can greatly reduce the execution time of cloud workflow tasks
- (2) **Execution cost:** this is the cost incurred from the use of service resources after the execution of a cloud workflow task, including bandwidth transmission costs and calculation costs. The execution cost of a cloud workflow task is related to the performance of the resources it uses, that is to say, if the user wants to submit the workflow task to be completed quickly, the task can be assigned to a virtual resource with strong processing capabilities for execution, and at the same time, users need to pay more for it
- (3) **Reliability:** in some fields such as aerospace and aviation, the reliability of the execution of workflow tasks is very high, because a data uncertainty may bring unexpected disasters. However, it is difficult to maintain the reliability of data in the cloud computing environment, because some abnormalities may occur in the processing of the task, or the task cannot be

executed due to problems in the underlying software and hardware

- (4) Security: cloud computing is cross-domain, and the virtual resources used by users may be distributed in different regions. When the cloud workflow tasks submitted by users are executed on virtual machine resources in different regions, tasks with dependencies will have data that is transmitted on the network. At this time, some network hackers may steal the user's data and cause the leakage of some important data of the user. Therefore, cloud workflow tasks need to ensure data security during the execution process and meet the security requirements of users in the cloud computing environment

In the cloud computing environment, the workflow tasks submitted by users will be assigned to each virtual resource for execution, and then the scheduling results will be fed back to the user. Therefore, this section first establishes a cloud workflow task model and a virtual resource model.

### 3.2. Workflow Task Scheduling System Model Based on Mobile Cloud Computing Machine Learning

**3.2.1. Cloud Computing Workflow Task Model.** The cloud workflow task in the cloud computing environment is composed of interdependent tasks. This paper uses an undirected cyclic graph (DAG) to describe them. Use  $H = (Y, R, V)$  to describe cloud workflow tasks; the meaning of each attribute and related definitions are expressed as follows:  $Y = (y_1, y_2, y_3, \dots, y_i)$  represents a cloud workflow task set, where  $y_i$  is the  $i$ th task ( $i = 1, 2, \dots, n$ ).  $y_i = \{y_{id}, y_{length}\}$  represents different attributes of the task, where  $y_{id}$  represents the number of the task  $y_i$  and the length of the task  $y_{length}$ .  $R = (r_{ij} | r_{ij})$  represents the edge of task  $y_i$  to task  $y_j$ , task  $y_i$  is the predecessor task of task  $y_j$ , task  $y_j$  is the successor task of task  $y_i$ , and task  $y_j$  must be executed after all of its predecessor tasks are completed.  $V = \{v(y_i, y_j) | v(y_i, y_j)\}$  represents the communication time between task  $y_i$  and task  $y_j$ . The following interdependent tasks are represented as follows:

- (a) Predecessor task set

$$\text{Pre}(y_i) = \{y_j | r_{ji} \in R\}. \quad (1)$$

- (b) Successor task set

$$\text{Suc}(y_i) = \{y_j | r_{ij} \in R\}. \quad (2)$$

- (c) Best precursor

$$Z\text{Pre}(y_i) = y_j | \text{Est}(y_j) + v(y_j, y_i) \geq \text{Est}(y_k, y_i), \quad (3)$$

$$y_j, y_i \in \text{Pre}(y_i), \quad j \neq k. \quad (4)$$

**3.2.2. Resource Model.** Cloud resources are composed of a series of heterogeneous virtual machine resources. The heterogeneity of virtual machines means that the computing power, memory, computing service unit price of different virtual machines, and the service unit price of communication bandwidth are different. Therefore, remember  $B_i B_j (B_1, B_2, B_3, \dots, B_m)$  as the set of virtual machines in the cloud computing data center,  $B_i = (B_{id}, B_{pe}, B_{mips}, B_{ram})$  represents the different attributes of virtual machines,  $B_{id}$  represents the number of the virtual machine resource  $B_i$  in the data center,  $B_{pe}$  represents the number of cores of the virtual machine  $B_i$ ,  $B_{mips}$  represents the computing power of the virtual machine  $B_i$ ,  $B_{ram}$  represents the memory size of the virtual machine  $B_i$ , and  $\text{price}(B_i)$  represents the task in the virtual machine  $B_i$ . The service unit price is calculated above, where  $c_{wij}$  represents the communication bandwidth between the virtual machine  $B_i$  and the virtual machine  $B_j$  and  $\text{tr}(b_i, b_j)$  represents the service unit price of the communication between the virtual machine  $B_i$  and the virtual machine  $B_j$ .

In order to facilitate the study of subsequent algorithms, the following assumptions are made for the model: (1) A virtual machine can only perform one task in a period of time, and virtual resources cannot be preempted when a task is being performed. (2) There are two dependent relationships, and the communication time of a task in the same virtual machine is 0. (3) The communication bandwidth between a virtual machine and another virtual machine is 1, that is, the communication time between tasks is equal to the amount of data transfer between tasks. (4) For tasks and tasks, the data transmission between virtual machines can be completed concurrently. (5) The communication of tasks between virtual machines does not affect the execution of tasks on virtual machines.

**3.2.3. Scheduling Algorithm for Task Migration.** The HEFT algorithm is used to find the earliest completion time of each cloud workflow task, the latest completion time is defined as the global latest completion time, and then makespan is used to replace the global latest completion time. For the convenience of the subsequent algorithm description, the following definitions are given.

**Definition 1.** Task execution time refers to the execution time allocated to the virtual machine resource  $b_k$  for the cloud workflow task  $y_i$ , which is recorded as follows:

$$e(y_i, b_k) = \frac{y_{length}}{b_{mips}}. \quad (5)$$

**Definition 2.** The earliest start time  $\text{Est}$  refers to the earliest start time of the cloud computing workflow task  $y_i$  scheduled

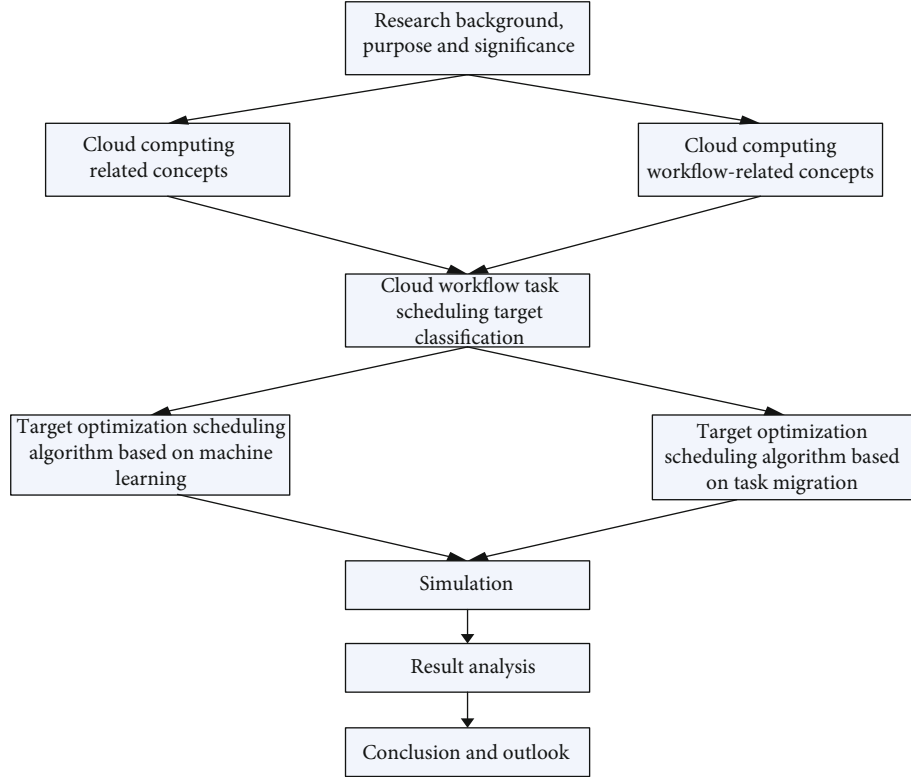


FIGURE 2: The research framework of this article.

to the virtual machine resource  $b_k$ . The calculation formula is as follows:

$$\text{Est}(y_i, b_k) = \begin{cases} 0, & \text{Pre}(y_i) = 0, \\ \max \{ \text{ava}(b_k), \max \{ \text{Aft}(y_j) + v(y_j, y_i) \} \}, & \text{Pre}(y_i) \neq 0, y_j \in \text{Pre}(y_i). \end{cases} \quad (6)$$

Among them,  $\text{ava}(b_k)$  represents the earliest available time of the virtual machine.

*Definition 3.* The earliest completion time Eft, that is, the earliest completion time of a cloud computing workflow task on a virtual machine, is recorded as follows:

$$\text{Eft}(y_i, b_k) = \text{Est}(y_i, b_k) + e(y_i, b_k). \quad (7)$$

*Definition 4.* Actual start time (Ast)/actual completion time (Aft) refers to the actual start time/actual completion time of the task on the assigned virtual machine.

*Definition 5.* The total time to complete a cloud computing workflow task makespan, that is, the longest time to complete a cloud computing workflow from the beginning of the task to the end of the task, is denoted as follows:

$$\text{Mak} = \max \{ \text{Eft}(y_i) \}. \quad (8)$$

*Definition 6.* The latest completion time Lft, that is, the latest completion time of a cloud workflow task on a virtual machine, is denoted as follows:

$$\text{Lft}(y_i, b_k) = \begin{cases} \text{makespan}, & \text{Suc}(y_i) = 0, \\ \min \{ \text{Lst}(y_j) - v(y_i, y_j) \}, & \text{Suc}(y_i) \neq 0, y_j \in \text{Suc}(y_i). \end{cases} \quad (9)$$

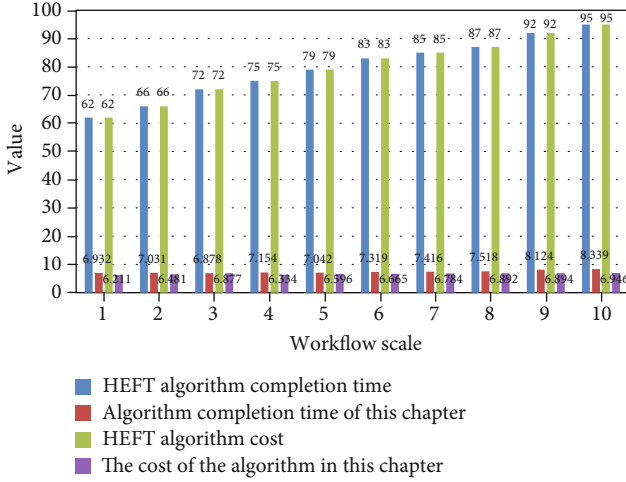


FIGURE 3: 10 simulation effects with 9 tasks assigned to 3 virtual machines.

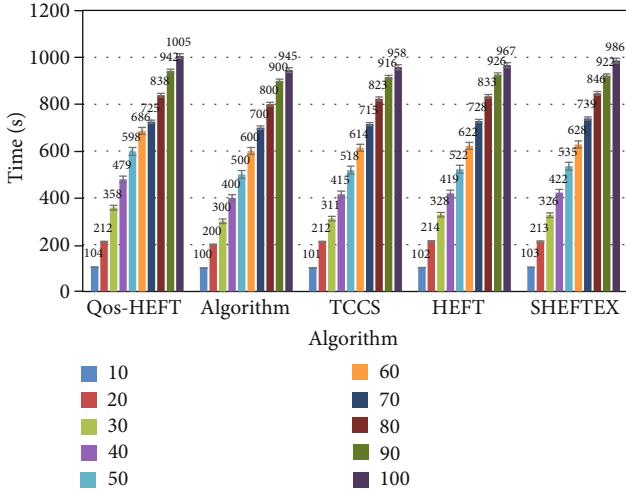


FIGURE 4: Comparison of task completion time of five algorithms under a different number of tasks.

**Definition 7.** The latest start time  $Lst$ , that is, the latest start time of the cloud workflow task on the virtual machine, is recorded as follows:

$$Lst(y_i, b_k) = Lft(y_i, b_k) - e(y_i, b_k). \quad (10)$$

**Definition 8.** Critical Path (CP), the longest path of the cloud workflow from the starting task to the ending task, is defined as the critical path of the cloud workflow.

**Definition 9.** Critical tasks refer to the tasks on the critical path. For cloud workflow tasks, if the task satisfies

$$Eft(y_i) = \max \{Eft(y_j)\}, \quad (11)$$

then  $y_i$  is the key task, and right again  $y_q \in \text{Pre}(y_i)$ , if the task  $y_q$  satisfies  $Eft(y_q) + v(y_q, y_i) = \max_{y_q \in \text{Pre}(y_i)} \{Eft(y_q) + v(y_q,$

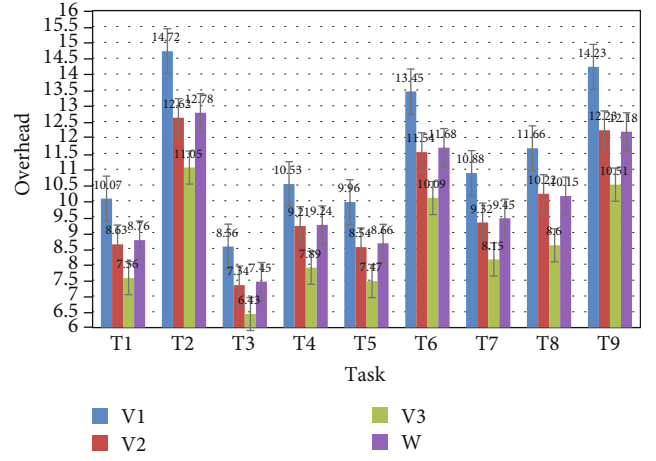


FIGURE 5: Heterogeneous computing overhead of DAG tasks.

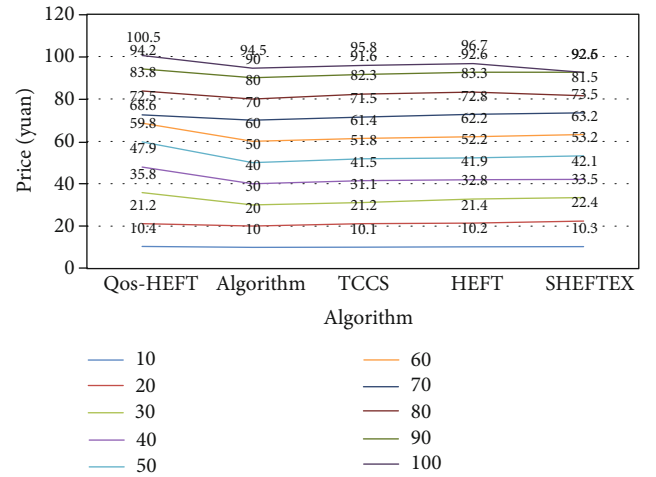


FIGURE 6: Task total service cost.

$y_i\}$ , then  $y_q$  is the key task. This way, we get all the key tasks on the critical path in turn.

**Definition 10.** Execution cost refers to the sum of the cost of cloud workflow tasks mapped to virtual machine resources and the communication cost of all predecessor tasks and tasks:

$$\text{Cos}(y_i) = \sum_{y_j \in \text{Pre}(y_i)} \left( v(y_j, y_i) * \text{tr}(b_c, b_k) + e(y_i, b_k) * \text{Pri}(b_k) \right), \quad (12)$$

where  $b_c$  is the virtual machine allocated by the predecessor task  $y_j$ . The completion time and economic cost of task  $y_i$  are related to the processing performance of resource  $b_k$  and the communication time between tasks.

When migrating noncritical tasks, you need to find a suitable idle time period  $[D_s, D_f]$  on other virtual machines with low execution costs. First, determine whether the idle time

TABLE 1: Details of algorithm task scheduling in this article.

Task	Data transmission before task execution				Task performance				
	Parent task	Subtasks	Source resource	Target resource	Transmission time	Resources	Starting time	Operation hours	Complete time
1	—	—	—	—	—	—	15.8	12.6	32.8
2	1-1	1-3	3	3	0.0	3	17.8	14.0	31.8
3	1-1	1-2	3	1	0.0	1	17.8	15.3	33.2
4	1-2	1-4	1	3	0.0	3	33.1	12.5	45.5
5	1-3	1-4	3	3	0.0	3	33.1	12.5	45.5
6	1-6	1-4	3	3	0.0	3	45.5	9.4	54.9
7	1-7	1-4	3	1	0.0	1	45.5	7.5	52.9
8	2-2	—	—	—	—	4	0.0	13.2	13.2
9	2-1	—	—	—	—	2	0.0	14.5	14.5
10	2-3	2-1	2	4	0.0	4	14.5	19.8	34.3
11	2-1	2-5	2	2	0.0	2	14.5	9.4	23.9
12	2-2	2-4	4	2	0.0	2	23.9	4.6	28.5
13	2-4	2-6	2	4	0.0	4	34.3	16.2	50.4
14	2-3	2-7	4	2	0.0	2	36.3	15.8	50.1
15	3-3	3-5	2	4	0.0	4	58.4	13.7	72.1
16	3-1	3-6	4	2	0.0	2	58.4	13.8	72.2
17	3-2	3-4	4	3	14.2	3	68.6	5.2	73.8
18	3-1	3-7	4	1	8.4	1	66.0	9.3	75.3

period can execute the task to be migrated, and then determine after the task is migrated whether it will affect the completion time of the subsequent tasks; if the idle time period [Ds, Df] can execute the task to be migrated and does not affect the completion time of the subsequent tasks, then the task will be migrated. That is, for a given cloud workflow task  $y_i \in Y$ , if  $Succ(y_i) \neq \emptyset$ , the migration condition (10), condition (11), and condition (12) are met, and the task can be migrated; if  $Succ(y_i) = \emptyset$ , only condition (10) and condition (11) can migrate, where  $b_k$  is its newly allocated virtual machine,  $b_p$  is its previously allocated virtual machine, and  $Cost'_{sum}$  is the cost of all tasks after task  $y_i$  migrates from virtual machine  $b_k$  to virtual machine  $b_p$ .

$$\max \left\{ \max \left\{ Aft(y_j) + v(y_j, y_i), F_s \right\} + e(y_i, b_k) \right\} \leq F_f, \quad (13)$$

$$Cost'_{sum} < Cost_{sum}, \quad (14)$$

$$Aft(y_i, b_k) + v(y_i, y_c) \leq \min \{Aet(y_c)\}, \quad y_c \in Succ(y_i). \quad (15)$$

When the current noncritical tasks have migrated, provide more free time for migration for the tasks to be migrated later, and move the current noncritical tasks to the latest completion time of the assigned virtual machine. If the current noncritical task does not meet the migration conditions, it will also be moved to the latest completion time of the allocated virtual machine.

The ranku(ti) value of the task is determined by the HEFT algorithm, then the priority of the task is determined, then the cloud workflow tasks with high priority are scheduled to the virtual machine with the minimum completion time in turn, and finally the total of the cloud workflow tasks is obtained. Completion time is makespan, and total completion cost is costsum. We define makespan as the global latest completion time. And according to the abovementioned knowledge, the total completion time of cloud workflow tasks is determined by the critical path, that is, determined by all the key tasks on the critical path. The cost optimization scheduling algorithm based on task migration is mainly for reasonably migrating some noncritical tasks to virtual machine resources with low execution cost under the premise of ensuring that the total completion time of the task remains unchanged, so as to optimize the total service cost. Figure 2 shows the research framework of this article.

#### 4. Workflow Scheduling Based on Mobile Cloud Computing Machine Learning

*4.1. Cloud Computing Workflow Task Scheduling Experiment Results.* Nine DAG workflow tasks are randomly generated and assigned to 3 virtual machines for simulation, and then the results of 10 simulation experiments of the algorithm in this section and the HEFT algorithm are compared as shown in Figure 3. It can be seen that the algorithm in this section does not increase cloud work. In the case of the total completion time of the flow task, the algorithm in this section can save costs more effectively than the HEFT algorithm. This is mainly because the algorithm in this section migrates some noncritical tasks to low execution costs while ensuring that



TABLE 2: Details of resource usage in the algorithm scheduling process of this article.

Resources	Previous task completion time	Idle time	Task	Starting time	Operation hours	Complete time	Resource utilization
1	0.0	17.8	1-2	17.8	15.3	33.1	
2	33.2	12.5	1-7	45.6	7.4	52.9	45.0
3	52.9	0.0	1-5	52.9	2.0	54.9	
4	54.9	11.1	3-7	66.0	9.3	75.3	
5	0.0	0.0	2-1	0.0	15.5	14.5	
6	14.5	0.0	2-5	14.5	9.4	23.9	
7	23.9	0.0	2-4	23.9	4.6	28.5	
8	28.6	5.8	2-7	34.3	15.8	52.1	88.2
9	50.2	0.0	3-3	50.2	8.3	58.4	
10	58.4	0.0	3-6	58.4	13.8	72.3	
11	0.0	0.0	1-1	0.0	17.6	17.8	
12	17.8	0.0	1-3	17.8	14.0	31.8	
13	31.8	1.3	1-4	33.2	12.5	45.5	78.1
14	45.6	0.0	1-6	45.6	9.5	54.9	
15	54.9	13.7	3-4	68.6	5.3	73.8	
16	0.0	0.0	2-2	0.0	13.2	13.2	
17	13.2	1.3	2-3	14.5	19.8	34.3	93.0
18	34.3	0.0	2-6	34.4	16.2	52.4	

the total completion time of cloud workflow tasks is not affected. It executes on virtual machine resources. The worst-case cost of the algorithm in this section is the same as the cost of the HEFT algorithm.

*4.2. Comparison of Task Completion Time.* Compare the algorithms in this section with the HEFT, QOS-HEFF, and TCCS algorithms. Among them, the QOS-HEFF algorithm is an improved algorithm for the HEFT algorithm according to user preferences (time and cost). This section evaluates the performance of this section's cloud workflow task scheduling algorithm through two indicators: cloud workflow task completion time and total service cost. This experiment uses five types of cloud workflows: 10, 20, 30, 40, and 50 tasks. The number of layers is set to 4, 6, 8, 10, 12, and 10 of each type of cloud workflow task that is randomly generated and then assigned to 3 virtual machines for simulation experiments, and each cloud work is recorded. The average values of the completion time and cost of the flow task and the experimental results are shown below.

It can be seen from Figure 4 that when the cloud workflow task scale is small, the task completion time of the algorithm in this section is slightly lower than the TCCS and HEFT algorithms, but far lower than the QOS-HEFT algorithm. As the cloud workflow task scale increased, the difference in completion time between the algorithm in this section and the TCCS, HEFT, and QOS-HEFT algorithms is becoming more and more obvious. This is mainly because the algorithm in this section reasonably copies the predecessor tasks of the current task and reduces the communication time between tasks. Thus, the current task is completed early, and the total completion time of the cloud workflow tasks is ultimately reduced.

TABLE 3: Energy consumption composition of cloud data center.

Energy consumption composition	Composition ratio
Computing equipment	46%
Cooling equipment	32%
Power system	5%
Lighting system	5%
Other devices	12%

*4.3. Task Cost Comparison.* Assuming that the  $v_{mips}$  of virtual machines V1, V2, and V3 are 600, 700, and 800 Mbits/s, respectively, the corresponding virtual machine service prices are 0.02, 0.05, and 0.09 yuan/sec, and the communication bandwidth service price is 0.03 yuan/sec. It can be seen from Figure 5 that by determining the rank (ti) of each task, the priority order of the cloud workflow task scheduling is determined as T2, T1, T5, T3, T4, T6, T7, T9, and T8. First, the cloud workflow task is prescheduled through the HEFT algorithm, and the total completion time of the cloud workflow task is  $madepan = 61.75$  seconds, the total cost spent is  $costsum = 6.8569$  yuan, and makespan is defined as the global latest completion time.

The total service cost refers to the cost of completing all tasks. As shown in Figure 6, the algorithm in this section requires the least cost to complete all tasks, which is significantly better than the TCCS and QOS-HEFT algorithms. The performance of the HEFT algorithm is the worst. This is because the algorithm in this section takes into account the communication cost between tasks. The replication stage reasonably replicates the precursor tasks to reduce the execution cost, then goes through the recopying stage to further

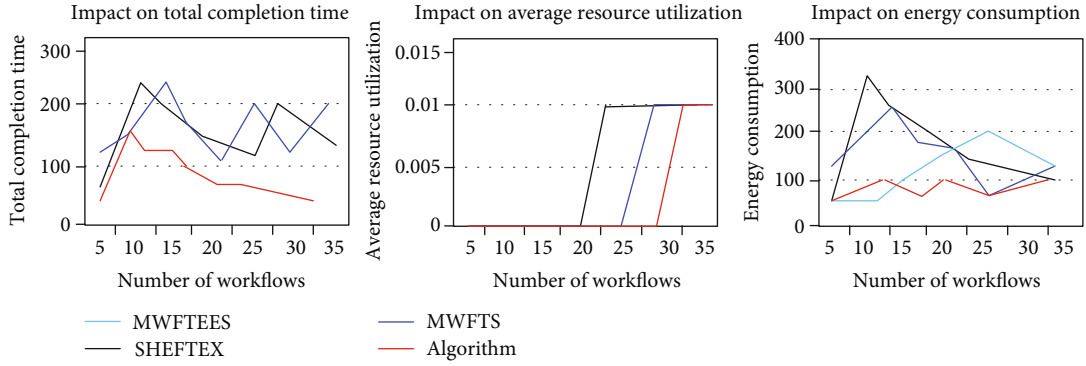


FIGURE 7: Data intensive: the influence of the number of workflows on experimental results.

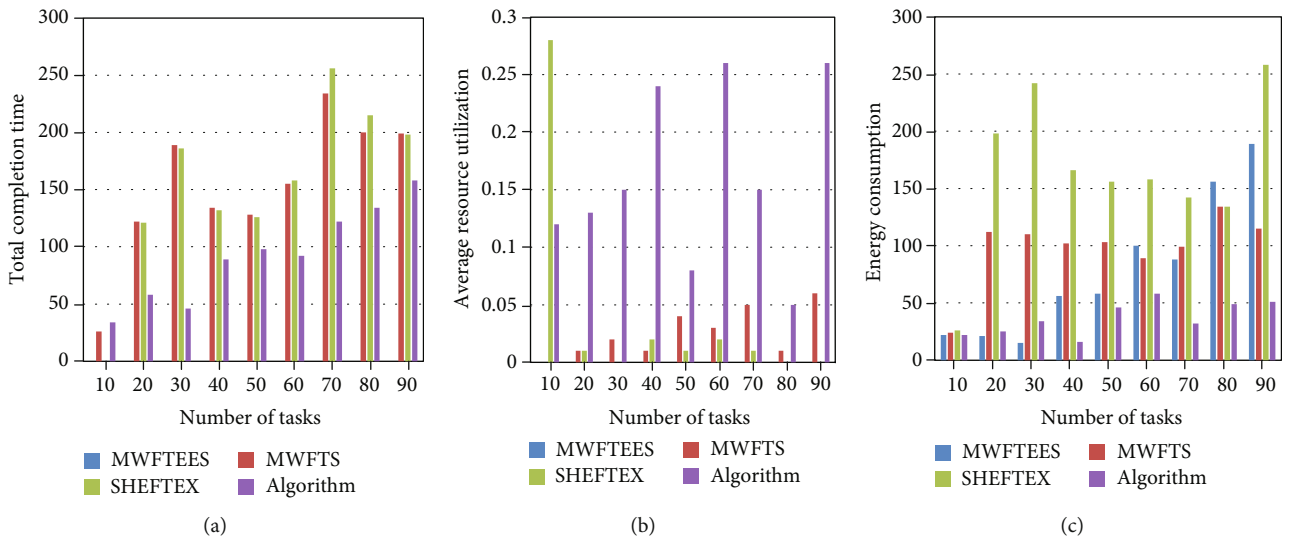


FIGURE 8: The influence of the number of tasks in each workflow on the experimental results.

reduce the execution cost, and finally optimizes the overall service cost by deleting redundant tasks.

**4.4. Task Scheduling and Resource Usage.** According to the algorithm in this paper, for the tasks of the three workflows, first, the time weight of the task is calculated according to the average execution time of the task, the average transmission time of the task in the resource, the task entry time, and the task exit time; then the tasks are sorted into the task execution sequence; and then appropriate resources are assigned to the task for execution. The execution scheduling results of the tasks are shown in Table 1:

The scheduling algorithm in this article is based on the task execution sequence. For each task in the sequence, first traverse all resources, then determine the idle time period in the resource, whether the task is satisfied, and if it is satisfied, it will be scheduled; otherwise, the running time will be selected.

It can be seen from Table 2 that in the process of the scheduling algorithm, resource 1 generates a total of 3 idle time periods, resource 2 generates a total of 1 idle time period, resource 3 generates a total of 2 idle time periods, and resource 4 generates a total of 2 idle time periods. The

resource utilization rates of the 4 resources are 45.0%, 88.2%, 78.1%, and 93.0%, respectively, and the average resource utilization rate is 74.1%.

**4.5. Influence of the Number of Workflows and Tasks on the Experimental Results.** The current development trend of cloud data centers is towards high-density and large-scale models, and its energy consumption is mainly concentrated on computer equipment, refrigeration equipment, and power systems. Many companies and researchers have investigated the energy consumption rate of various types of machines in data centers. It can be seen from Table 3 that other machinery and equipment account for about 22%, and the energy consumption of the computing machine as the processor accounts for 46% of the total energy consumption. Therefore, computer equipment has become an important factor in data center energy consumption.

Figure 7 shows that for data-intensive workflows, as the number of workflows increases, the total completion time of the tasks of the four algorithms does not change much, while the average resource utilization rate has increased but the growth is slower, and the maximum value of the resource utilization rate is always small and does not exceed 0.15. This

is also due to the fact that the data transmission time of the data-intensive workflow is much longer than the task execution time, which results in an excessively long period of resource idle time.

MWFTS is a task scheduling algorithm based on multiple workflows, and on this basis, the MWFTEES energy-saving strategy is proposed. The SHEFTEX algorithm is mainly used to deal with resource idleness that easily occurs in the workflow process. Through the comparison of the data in Figure 8, it can be found that whether it is the total completion time of the task or the average utilization of resources, the performance of the algorithm in this paper is far better or slightly better than the MWFTS algorithm, the MWFTEES algorithm, and the SHEFTEX algorithm in terms of scheduling; in the system energy consumption above, the energy-saving effect of the algorithm in this paper is obviously better than the other three algorithms. From this set of experiments, we can fully see the superiority of the algorithm in this paper for task scheduling based on multiple workflows.

Figure 8 shows the comparison between the SHEFTEX algorithm, the MWFTS algorithm, and the MWFTEES algorithm, and the experimental results of this algorithm from the three aspects of the total completion time of workflow tasks, the average utilization of resources, and system energy consumption. It can be seen from graphical comparison that for data-intensive workflows, the total task completion time and the average resource utilization change of the SHEFTEX algorithm and the MWFTS algorithm are basically the same; but the resource utilization is very small, not exceeding 0.3. This is also due to the fact that the data transmission time of the data-intensive workflow is much longer than the task execution time, which results in an excessively long period of resource idle time. In terms of energy consumption, the algorithm in this paper is better than the other three algorithms.

This paper defines the standard deviation of the total number of tasks running on a virtual machine as a load factor. If a large number of tasks are allocated on some virtual machines and some are idle, the load of the system is unbalanced. A 10-layer DAG workflow model is used for experiments; Gaussian random is used to set virtual machine and task parameters, and 20 virtual machines are configured. The average processing capacity of the virtual machine is 1200, and the standard deviation is 150; the average of the task length is 15000, the standard deviation is 200, and the number of tasks is 100~400. The experimental results are shown in Figure 9. As can be seen from the data in Figure 9, the algorithm in this paper achieves simple load balancing.

According to the random search function of the algorithm, the results of multiple simulation experiments are selected, and the results are shown in Figure 10. Because the number of LIGO operations is increased from 30 to 150, and the overall average and variance of the solution are compared, the algorithm proposed in this article has specific advantages. The reason for this phenomenon is that the algorithm proposed in this paper improves the grid coordinate adaptation system and dynamically adapts to changes based on network distribution, so that the algorithm takes into account a certain degree of time and cost.

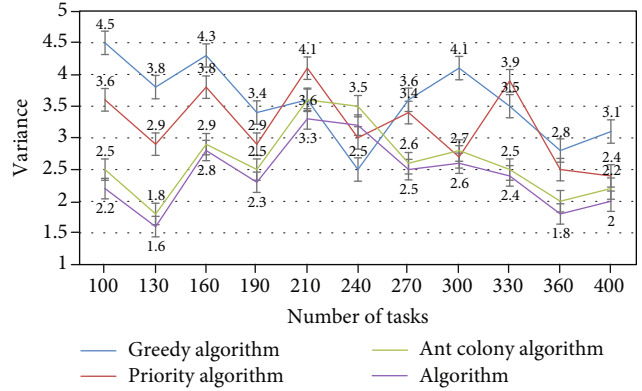


FIGURE 9: The influence of load balancing factor on the algorithm model of this paper.

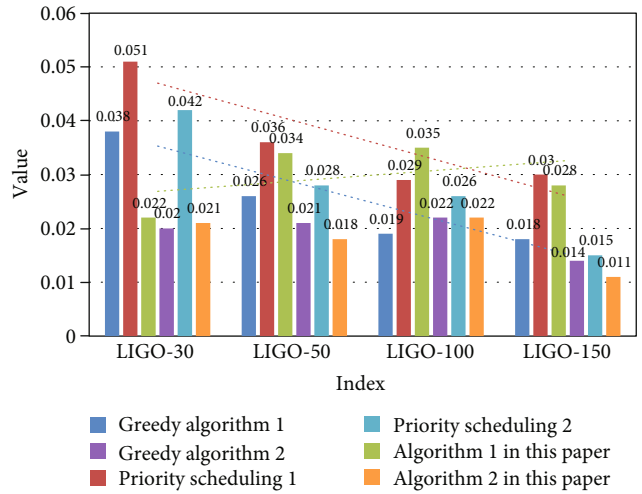


FIGURE 10: Mean and variance of various algorithms.

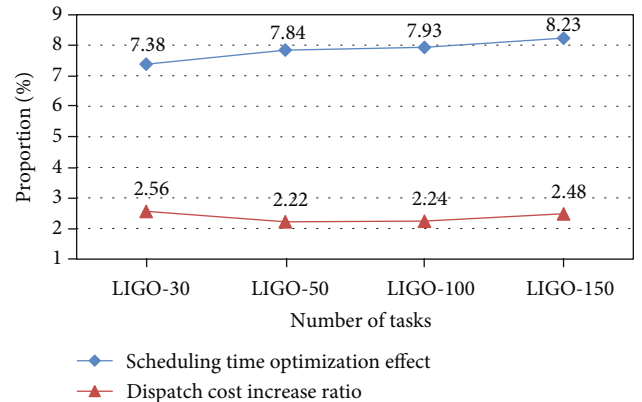


FIGURE 11: Comparison chart of scheduling time optimization effect and cost increase.

4.6. Task Scheduling Optimization Effect of the Algorithm. Figure 11 shows that the algorithm in this paper is 7.38%, 7.84%, 7.93%, and 8.23% faster in scheduling time. Moreover, as the workload increases, the effect of the algorithm in this

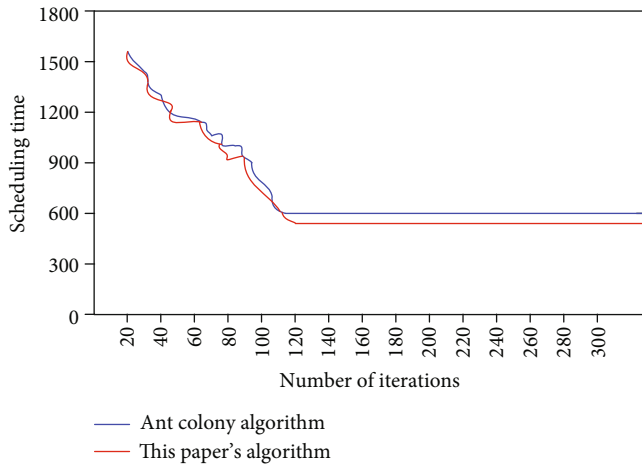


FIGURE 12: Convergence speed comparison.

paper to optimize the time will be more obvious. Compared with the HEFT algorithm, this algorithm only increases the service overhead by 2.56%, 2.22%, 2.24%, and 2.48%. Experimental results show that by adding a small amount of cost, the scheduling time can be greatly optimized. This is especially important in time-sensitive mission planning.

In this paper, the number of LIGO tasks is set to 150, and the time to plan to obtain the optimal solution is used as an indicator. As shown in Figure 12, the ant colony algorithm will iterate about 90 times for convergence, but the algorithm in this paper will converge about 105 times for iteration. Therefore, the experimental results show that the ant colony algorithm is slightly better than the algorithm proposed in this paper in terms of convergence speed. From the perspective of scheduling time, the final scheduling time of the ant colony algorithm is about 640 s, but the algorithm scheduling time of this article is about 588 s. Therefore, the algorithm in this paper is better than the ant colony algorithm in terms of time scheduling. It can be seen that task aggregation plays a better role in work scheduling.

## 5. Conclusion

This article is mainly about the research of workflow scheduling based on mobile cloud computing machine learning. Unlike previous research, this article adds new concepts and uses new applications of machine learning based on this research, and considers the cost and time dual-objective optimization problem. The algorithm in this paper is better than other algorithms on the whole, and has obvious improvements in time scheduling and task scheduling, in shortening the total completion time of the entire workflow task, and in saving the cost of task service. The innovation of this paper is to conduct a large number of simulation experiments, combining theory with empirical research and analysis, highlighting the focus of this paper. In the future, we will consider the next step of multiworkflow task scheduling, expanding the application of mobile computing and machine learning, and hope that there will be better breakthroughs.

## Data Availability

No data were used to support this study.

## Conflicts of Interest

The authors declare no conflicts of interest.

## References

- [1] L. A. Tawalbeh, R. Mehmood, E. Benkhelifa, and H. Song, "Mobile cloud computing model and big data analysis for healthcare applications," *IEEE Access*, vol. 4, no. 99, pp. 6171–6180, 2016.
- [2] A. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.
- [3] M. Masdari, S. ValiKardan, Z. Shahi, and S. I. Azar, "Towards workflow scheduling in cloud computing: a comprehensive analysis," *Journal of Network & Computer Applications*, vol. 66, pp. 64–82, 2016.
- [4] Z. Lv and H. Song, "Mobile internet of things under data physical fusion technology," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4616–4624, 2020.
- [5] L. Yang, Z. Han, Z. Huang, J. Ma, Han et al., "A remotely keyed file encryption scheme under mobile cloud computing," *Journal of Network and Computer Applications*, vol. 106, pp. 90–99, 2018.
- [6] X. Jin, Y. Liu, W. Fan, F. Wu, and B. Tang, "Energy-efficient resource management in mobile cloud computing," *IEICE Transactions on Communications*, vol. 101, no. 4, pp. 1010–1020, 2018.
- [7] N. Parajuli, A. Alsadoon, P. Prasad, R. S. Ali, and O. H. Alsadoon, "A recent review and a taxonomy for multimedia application in mobile cloud computing based energy efficient transmission," *Multimedia Tools and Applications*, vol. 79, no. 41–42, article 9516, pp. 31567–31594, 2020.
- [8] Z. Almusaylim and N. Z. Jhanjhi, "Comprehensive review: privacy protection of user in location-aware services of mobile cloud computing," *Wireless Personal Communications*, vol. 111, no. 1, article 6872, pp. 541–564, 2020.
- [9] Q. Jiang, V. Leung, H. Tang, and H. S. Xi, "Adaptive scheduling of stochastic task sequence for energy-efficient mobile cloud computing," *IEEE Systems Journal*, vol. 13, no. 3, pp. 3022–3025, 2019.
- [10] S. Misra, B. E. Wolfinger, M. P. Achuthananda, T. Chakraborty, S. N. Das, and S. Das, "Auction-based optimal task offloading in mobile cloud computing," *IEEE Systems Journal*, vol. 13, no. 3, pp. 2978–2985, 2019.
- [11] D. Facchinetti, G. Psaila, and P. Scandurra, "Mobile cloud computing for indoor emergency response: the IPSOS assistant case study," *Journal of Reliable Intelligent Environments*, vol. 5, no. 3, pp. 173–191, 2019.
- [12] Y. Zhang, A. Wu, and D. Zheng, "Efficient and privacy-aware attribute-based data sharing in mobile cloud computing," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 4, pp. 1039–1048, 2018.
- [13] H. S. Lee and J. W. Lee, "Task offloading in heterogeneous mobile cloud computing: modeling, analysis, and cloudlet deployment," *IEEE Access*, vol. 6, no. 99, pp. 14908–14925, 2018.

- [14] R. U. Amin, I. Inayat, B. Shahzad, K. Saleem, and L. Aijun, "An empirical study on acceptance of secure healthcare service in Malaysia, Pakistan, and Saudi Arabia: a mobile cloud computing perspective," *Annals of Telecommunications*, vol. 72, no. 5-6, pp. 253-264, 2017.
- [15] A. Tharwat, H. Mahdi, M. Elhoseny, and A. E. Hassanien, "Recognizing human activity in mobile crowdsensing environment using optimized  $k$ -NN algorithm," *Expert Systems with Applications*, vol. 107, pp. 32-44, 2018.
- [16] Y. Ding, G. Xu, C. Wu, L. Hu, Y. Zhai, and J. Zhao, "Explore virtual machine deployment to mobile cloud computing for multi-tenancy and energy conservation in wireless network," *Cluster Computing*, vol. 20, no. 4, article 1054, pp. 3263-3274, 2017.
- [17] Z. Lv and L. Qiao, "Optimization of collaborative resource allocation for mobile edge computing," *Computer Communications*, vol. 161, pp. 19-27, 2020.
- [18] N. Jean, M. Burke, M. Xie, W. M. Davis, D. B. Lobell, and S. Ermon, "Combining satellite imagery and machine learning to predict poverty," *Science*, vol. 353, no. 6301, pp. 790-794, 2016.
- [19] N. D. Sidiropoulos, L. de Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551-3582, 2017.
- [20] Y. Li, M. Chen, W. Dai, and M. Qiu, "Energy optimization with dynamic task scheduling mobile cloud computing," *IEEE Systems Journal*, vol. 11, no. 1, pp. 96-105, 2017.
- [21] M. Elhoseny and K. Shankar, "Reliable data transmission model for mobile ad hoc network using signcryption technique," *IEEE Transactions on Reliability*, vol. 69, no. 3, pp. 1077-1086, 2020.
- [22] C. Stergiou and K. E. Psannis, "Recent advances delivered by mobile cloud computing and internet of things for big data applications: a survey," *International Journal of Network Management*, vol. 27, no. 3, article e1930, pp. 1-12, 2017.
- [23] C. Helma, T. Cramer, S. Kramer, and L. de Raedt, "Data mining and machine learning techniques for the identification of mutagenicity inducing substructures and structure activity relationships of noncongeneric compounds," *Journal of chemical information and computer sciences*, vol. 44, no. 4, pp. 1402-1411, 2004.
- [24] Z. Zhu, G. Zhang, M. Li, and X. Liu, "Evolutionary multi-objective workflow scheduling in cloud," *IEEE Transactions on Parallel & Distributed Systems*, vol. 27, no. 5, pp. 1344-1357, 2016.
- [25] H. Xu, B. Yang, W. Qi, and E. Ahene, "A multi-objective optimization approach to workflow scheduling in clouds considering fault recovery," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 10, no. 3, pp. 976-995, 2016.
- [26] K. Sathish and A. RamaMohan Reddy, "Workflow scheduling in grid computing environment using a hybrid GAACO approach," *Journal of the Institution of Engineers*, vol. 98, no. 1, pp. 121-128, 2017.
- [27] S. Shukla, A. Kumar, S. Saxena, and S. Kumar, "An evolutionary study of multi-objective workflow scheduling in cloud computing," *International Journal of Computer Applications*, vol. 133, no. 14, pp. 14-18, 2016.
- [28] B. Lin, W. Guo, N. Xiong, G. Chen, A. V. Vasilakos, and H. Zhang, "A pretreatment workflow scheduling approach for big data applications in multi-cloud environments," *IEEE Transactions on Network & Service Management*, vol. 13, no. 3, pp. 581-594, 2016.
- [29] A. Deldari, M. Naghibzadeh, and S. Abrishami, "CCA: a deadline-constrained workflow scheduling algorithm for multicore resources on the cloud," *The Journal of Supercomputing*, vol. 73, no. 2, pp. 756-781, 2017.