



**REVIEW**

# Survey on Task Scheduling Optimization Strategy under Multi-Cloud Environment

Qiqi Zhang<sup>1</sup>, Shaojin Geng<sup>2</sup> and Xingjuan Cai<sup>1,\*</sup>

<sup>1</sup>School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan, 030024, China

<sup>2</sup>College of Electronics and Information Engineering, Tongji University, Shanghai, 201804, China

\*Corresponding Author: Xingjuan Cai. Email: xingjuancai@163.com

Received: 02 March 2022 Accepted: 03 August 2022

## ABSTRACT

Cloud computing technology is favored by users because of its strong computing power and convenient services. At the same time, scheduling performance has an extremely efficient impact on promoting carbon neutrality. Currently, scheduling research in the multi-cloud environment aims to address the challenges brought by business demands to cloud data centers during peak hours. Therefore, the scheduling problem has promising application prospects under the multi-cloud environment. This paper points out that the currently studied scheduling problems in the multi-cloud environment mainly include independent task scheduling and workflow task scheduling based on the dependencies between tasks. This paper reviews the concepts, types, objectives, advantages, challenges, and research status of task scheduling in the multi-cloud environment. Task scheduling strategies proposed in the existing related references are analyzed, discussed, and summarized, including research motivation, optimization algorithm, and related objectives. Finally, the research status of the two kinds of task scheduling is compared, and several future important research directions of multi-cloud task scheduling are proposed.

## KEYWORDS

Cloud computing; task scheduling; workflow; review; multi-cloud environment

## 1 Introduction

As a kind of efficient resource, cloud computing [1,2] provides users with supercomputing power over the Internet. It can easily obtain the required network, server, storage, and other resources from the configurable computing resource sharing pool. These resources can be used quickly and normally with less management work. Cloud computing [3,4] mainly uses virtualization technology to share datacenter resources between multiple virtual machines for more energy-efficient resource management, such as integrated development environment, server, application software, CPU, etc. It means that users can purchase complex IT infrastructures without any investment and they can pay for the services from anywhere in the world. In recent years, related research on fog computing [5,6] and edge computing [7] has further expanded the cloud computing framework to address the communication pressure due to distance.



The rapid development of cloud computing technology [8] is inseparable from the policy support of various countries. However, with the Internet of Things (IoT) [9,10] becoming more prosperous, the unprecedented data explosion has continuously increased the demand for resources. And the single cloud is likely to fail applications due to localized failures, resulting in business interruption and low availability. At the same time, it is prone to problems such as provider monopoly, Service Level Agreement (SLA) violation, low user data privacy security, and Quality of Service (QoS) reduction. Therefore, various countries advocate further innovation of key technologies such as hyper scale distributed storage elastic computing to improve the performance of cloud computing. Against this backdrop, the multi-cloud environment has emerged, which consists of multiple different IaaS providers with different advantages. It will provide help when users encounter difficulties in choosing, and the reason is that the heterogeneous advantages of a multi-cloud environment can provide users with diversified virtual resources to meet their QoS requirements. At the same time, existing multi-cloud computing research also considers fog computing and edge computing frameworks to construct collaborative computing resources, creating a better resource environment for real-time and efficient task execution.

Scheduling problems [11] can be seen everywhere in our daily life and production, such as workshop scheduling [12], reservoir benefit scheduling [13], path planning [14] scheduling. A good scheduling scheme can optimize [15] cost efficiency, reduce time consumption, and improve the overall business level, which is of great significance to the convenience of users and the economic prosperity of the country. With the continuous growth of the scale of various industries, most scheduling problems [16] require the processing of massive data and higher computing power platforms, and the characteristics of cloud computing can just provide a high-quality platform for existing large-scale scheduling problems. Cloud task scheduling [17] has become the main research focus of scholars from all walks of life, especially in the multi-cloud environment.

The multi-cloud environment [18] can prevent data loss and reduce the risk of downtime due to local component failure, avoid cloud provider lock-in, and have high scalability. In this paper, independent task scheduling and workflow scheduling are analyzed as two different task scheduling types, and there are a lot of references on these two kinds of scheduling strategies in the multi-cloud environment. In order to enable scholars to better and faster grasp the research history and current situation of multi-cloud scheduling [19], and pay attention to the future research trends, priorities, and hotspots of this discipline. It is essential to review the scheduling problems and highlight their functions and features.

This paper conducts a comprehensive analysis and research on the task scheduling strategy [20] under the multi-cloud environment. Firstly, the background of multi-cloud task scheduling is introduced, including the advantages and the challenges faced by multi-cloud task scheduling at this stage. It then further introduces the common objectives, constraints and scheduling algorithms of multi-cloud task scheduling. In addition, existing references on scheduling strategies for the independent task and workflow in the multi-cloud environment are reviewed according to task dependencies. And the detailed summary and comparative analysis of these references are carried on. Finally, several future research directions are summarized. It is worth noting that because the multi-cloud task scheduling strategy contains extremely abundant content and the algorithms [21] used are intricate, it is difficult for this paper to cover all methods. However, it is committed to systematically opening discussions on the potential research direction of multi-cloud task scheduling so that scholars can better and faster grasp the research history, current situation, future research trends, priorities, and hotspots of multi-cloud scheduling. This paper provides the necessary reference for the current research in the field of multi-cloud task scheduling.

The rest of this paper is structured as follows: [Section 2](#) provides the related characteristics, advantages, and challenges under the multi-cloud environment. On the basis of the latest and representative research references, the third section review, and summary of motivation, objectives, constraints, test environments and scheduling strategies. Finally, the fourth part summarizes and points out the future research direction.

## 2 Background

Scheduling problems [22], also known as combinatorial optimization [23] problems, exist in all walks of life and have an important impact on output value and capacity. Such as workshop scheduling [24], vehicle scheduling [25–27], UAV scheduling [28–30], cloud computing scheduling [31], and other task scheduling. As the scheduling scale gradually expands, the scheduling algorithm [32] is constantly updated and iterative, from static algorithm to dynamic algorithm, from single-objective algorithm to multi-objective algorithm [33]. Among them, multi-cloud computing, as an important representative of high-performance computing power in the new century, has far-reaching significance in the study of its scheduling problem [34]. This section first analyzes the advantages of multi-cloud and then summarizes the challenges faced by multi-cloud scheduling.

### 2.1 Multi-Cloud Environment

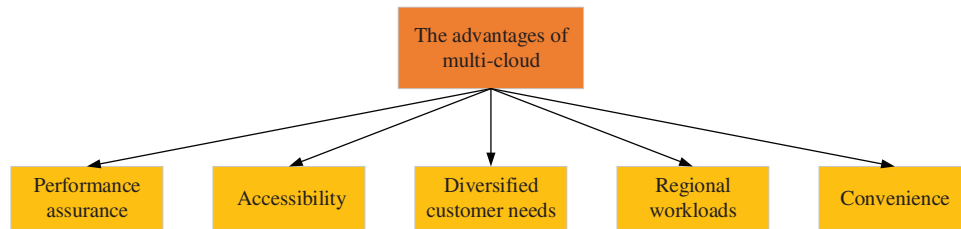
The multi-cloud computing environment has emerged when a single cloud environment cannot efficiently handle computing tasks during peak hours with the huge demand for virtual resources by users. The multi-cloud environment consists of multiple single clouds, specifically, it consists of several different IaaS cloud service providers that provide virtual machines with different prices and performance, and these cloud providers do not voluntarily connect and do not share virtual resources. It can realize the integrated utilization of cloud resources from the perspective of users or cloud providers. Users can build workload management programs by themselves to satisfy task scheduling needs [35], and each cloud provider designs corresponding API interfaces to achieve communication. In general, it is crucial for multi-cloud application portability [36].

The purpose of cloud resource scheduling [37] is to allocate reasonable resources to users. In the multi-cloud environment, in order to satisfy the diversity of customer needs, there is a resource manager on each cloud platform to send the current information of virtual machine resources to users in real-time. Users obtain resource information sent by different cloud platforms through a unified API interface or external agents and transfer tasks to different clouds for processing and calculation according to their own task conditions. Therefore, the decision-making of many scheduling schemes has a vital influence not only on users but also on providers.

The multi-cloud environment combines the characteristics of different cloud providers and then provides users [38] with diversified choices, which can avoid the limitations of single cloud lock-in. Compared with the traditional single cloud environment, the multi-cloud environment has the following advantages as shown in [Fig. 1](#).

- (1) Performance assurance: Once a single cloud suffers from service interruption, insufficient resources, etc., server performance can be maintained by the resources of other cloud providers.
- (2) Accessibility: The migration of virtual machines and data allows geographical differences between cloud providers and improves scheduling performance.

- (3) **Diversified customer needs:** Maintaining customer satisfaction is important to improve the customer experience. The heterogeneous nature of different providers offers customers diversified services.
- (4) **Regional workloads:** Because of geographic dispersion, workloads can be redirected to the cloud closer to the customer.
- (5) **Convenience:** Provides convenience for users with respect to services through non-operational and highly integrated virtual resources.



**Figure 1:** The advantage of multi-cloud environment

At the same time, data security and elastic services are the distinguishing characteristics of multi-cloud systems, which are reflected in the decentralization of user data and the ability to adaptively solve load problems. In other words, different providers offer more heterogeneous resources to complete the tasks submitted by cloud customers. Therefore, how to design the task scheduling strategy [39], namely efficient resource management under the multi-cloud environment, is a challenging problem.

## 2.2 Inherent Challenges

As an important trend in the development of cloud computing, multi-cloud will become one of the preferred solutions for most users. Despite the advantages of a multi-cloud environment, compared to the single cloud, multi-cloud scheduling faces many significant challenges, including data security, modeling complexity, and scheduling robustness.

### (1) Security

First of all, since multi-cloud means mutual cooperation between multiple cloud providers, a very important issue is security. Although it indirectly reduces the risk of privacy leakage by distributing the same user-submitted tasks across different clouds, it also increases the risk of the provider being attacked.

### (2) Modeling

Secondly, due to the complex constraints on scheduling tasks during the multi-cloud scheduling process, including the performance differences of different services in the same cloud or the complexity of the pricing models of different clouds. Therefore, how to accurately model is an important challenge, which directly affects the reliability of the scheduling scheme.

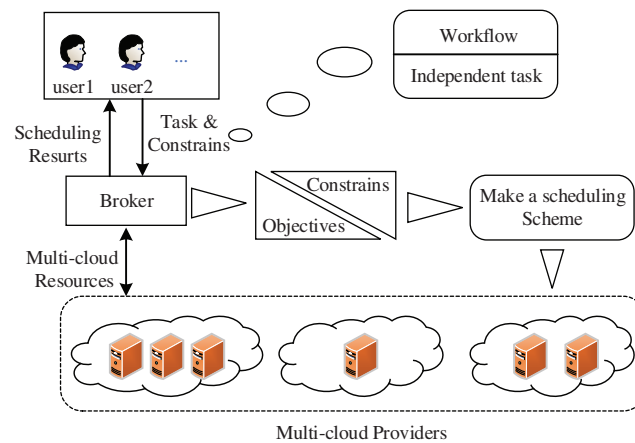
### (3) Robustness

In the process of multi-cloud scheduling, the problem of task transmission between clouds is often accompanied. When the transmission is interrupted, the task execution fails or the virtual machine fails, a reasonable and fast solution needs to be designed to improve the robustness of scheduling.

### 3 Scheduling Method under the Multi-Cloud Environment

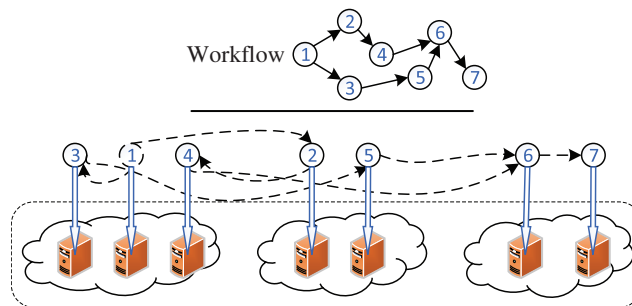
With the popularization of cloud computing optimization problems [40], scheduling problems [41] under the multi-cloud environment have attracted widespread attention from many scholars. Based on some basic scheduling algorithms, the majority of scholars have provided many advanced task scheduling solutions for the multi-cloud environment, so as to provide better services for customers.

According to the dependency relationship of tasks in the multi-cloud environment, task scheduling problems are divided into independent task scheduling and workflow scheduling. The task scheduling process under multi-cloud environment can be expressed in Fig. 2. Firstly, the user submits the tasks to be processed (independent tasks or workflows) and constraints to the broker. Then, the broker optimizes an appropriate scheduling scheme through the designed scheduling model. The scheduling scheme is the matching relationship between tasks and cloud resource virtual machines so that different tasks are allocated to the most suitable virtual machines for execution. Finally, the broker feeds back a satisfactory scheduling result to the user.



**Figure 2:** The execution of task scheduling in multi-cloud environment

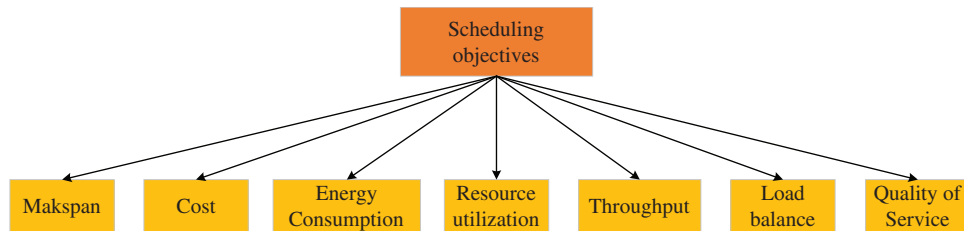
Independent task scheduling [42] means that the tasks have no relationship with each other. The purpose of scheduling is to make these tasks can be allocated to the most suitable virtual machine to satisfy the user's needs under the user's expectation constraints. At the same time, from the perspective of providers, we also hope to maximize the benefits as much as possible. And the workflow scheduling is far more complicated than the independent task scheduling because the workflow tasks have complex or simple connections with each other, so the execution order of the tasks needs to be considered during scheduling. A directed acyclic graph is usually used to represent workflow. Fig. 3 depicts an example of a simple workflow scheduling, where nodes represent tasks and directed edges represent the execution order between tasks. In this section, some task scheduling objectives, constraints and several commonly used scheduling optimization algorithms [43,44] will be introduced, and the two task types of scheduling strategies proposed in the existing references are analyzed and summarized. At the same time, the constraints and the test environments of experiments in different references are analyzed.



**Figure 3:** The example of a simple workflow scheduling

### 3.1 Related Objectives

The multi-cloud environment provides users with abundant computing resources, while also increasing user expectations. In order to enable users to obtain a better service experience and maximize the benefits of providers, it is very important to design a better scheduling scheme under the condition of limited resources. In this section, we illustrate the common scheduling objectives involved in the related multi-cloud scheduling reference as shown in Fig. 4. Since we only express the meaning of the objectives involved in multi-cloud task scheduling and do not discuss the specific calculation methods, these objectives are equally applicable to independent task and workflow scheduling.



**Figure 4:** The objectives of task scheduling in multi-cloud environment

#### (1) Makespan

After users put forward service requirements to cloud resources, they hope that the submitted tasks can be completed as soon as possible. Especially in the case of urgent and real-time tasks, the corresponding processing results must be obtained within the specified time, so the task Makespan is an important objective to be considered in the scheduling scheme. And a smaller objective value is expected.

#### (2) Cost

This objective refers to the total cost required for all tasks to be executed after a user submits a task. Generally, it includes virtual machine running costs, data transmission costs, data storage costs, etc. The lower the cost, the better for the user.

#### (3) Energy Consumption

The cloud server in the cloud platform consumes a lot of power due to long-term operation and also reduces the performance due to the heat of the server. Whether it is huge power consumption or a large number of cooling devices purchased for server cooling, a large amount of energy consumption

is required. To further push carbon neutrality, the indicator of energy consumption should be reduced as much as possible during the scheduling process.

#### (4) Throughput

It refers to the number of tasks completed on the cloud per unit of time. The more tasks performed per unit time, the stronger the execution capability of cloud resources. So the goal is to obtain a maximum value.

#### (5) Load Balance

Cloud providers hope that the tasks in the platform can be distributed on each server as reasonably and evenly as possible. Load refers to the workload of different computing nodes to process data. In the process of task processing, the load balance of computing resources is the expected state when processing tasks. Load balance not only enables tasks to be executed efficiently but also prolongs the overall service life of servers and improves cloud resource utilization.

#### (6) Quality of Service

In general, users have a variety of service requirements when processing tasks. And different users have various requirements, such as safety, cost, time, etc. Since the cloud computing platform mainly provides services for users, the evaluation made by users on the cloud platform is an important measure of its reputation. QoS is determined by the user's initial expectation and the actual degree of completion when submitting the task. For example, the cost constraints, time constraints, and fault tolerance given by the user will affect the QoS value. The larger the QoS value, the better for the user.

#### (7) Resource Utilization

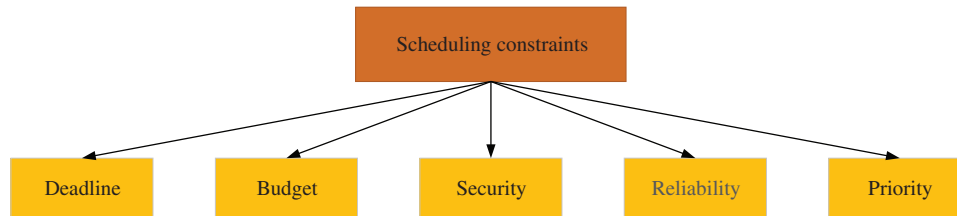
Resource utilization refers to the usage of computing resources during task execution. When the task can make full use of the computing resources allocated by the provider, the overall efficiency of the task will be greatly improved. Therefore, the resource utilization is as large as possible.

### **3.2 Related Constraints**

The constraints of scheduling problems usually come from the requests of users, such as task completion deadlines, budgets, reliability, and specific priorities for task execution. Despite the increased scheduling complexity, users can obtain better scheduling schemes under the premise of satisfying user QoS constraints. In this section, scheduling constraints involved in related multi-cloud scheduling references are introduced as shown in [Fig. 5](#).

- 1) **Deadline:** It refers to the maximum acceptable completion time when a user submits a task for execution on cloud resources, in other words, the cloud provider must complete the task within this period.
- 2) **Budget:** It refers to the maximum cost that the user can pay to complete the task. Therefore, the total cost of the designed scheduling scheme cannot exceed the budget when executing tasks.
- 3) **Security:** It means that the task will not easily lead to data privacy leakage or loss due to external attacks or internal failures during the execution of tasks.
- 4) **Reliability:** It means that when some faults occur during the execution of the task, there is a reasonable solution to make the task continue to execute smoothly without interruption.

- 5) Priority: If the tasks submitted by the user have different importance or sequential dependencies, the tasks will be given different execution orders in the early stage of execution, and the tasks with higher priority will be executed first.



**Figure 5:** The constraints of task scheduling in multi-cloud environment

### 3.3 Scheduling Algorithm

At present, in order to address the complex scheduling problem [45] under the multi-cloud environment, many scheduling algorithms have been designed, which can be roughly divided into two types: classical scheduling algorithm and heuristic scheduling algorithm [46,47].

Classical scheduling algorithms usually design a simple scheduling mechanism to handle various types of tasks, including timing, priority, and other issues. In order to obtain an accurate, feasible, and widely applicable task scheduling scheme, which can simultaneously satisfy multiple QoS objectives (such as waiting time, response time, maximum completion time, efficiency, delay, energy consumption, cost, resource utilization, etc.), multi-objective optimization [48,49] task scheduling algorithm based on heuristic idea emerged.

The intelligent algorithm of heuristic thought usually refers to the algorithm that people construct according to natural phenomenon, social experience, or biological inspiration, and then by summarizing simulation innovation. The idea is that when addressing multi-constraint problems, a feasible solution can be given to satisfy multiple objective optimizations [50] as far as possible. The focus is on multi-objective optimization [51–53], for each instance, the solution obtained now is not the optimal solution for a single objective, but the best solution under the condition of satisfying the requirements of multiple objectives as far as possible.

Next, we will introduce some classical scheduling algorithms and two kinds of intelligent multi-objective optimization [54,55] task scheduling algorithms based on heuristic ideas.

#### 1) Classic scheduling algorithm

The strategy of first come first service (FCFS) [56] is a relatively common real-time scheduling algorithm. The algorithm is relatively simple, and its advantages and disadvantages are more obvious. The advantage is low cost, and the disadvantage of using various scenarios is that the optimal task allocation scheme is not considered, and more tasks may appear in the waiting queue. Max-Min [57], Min-Min algorithm [58], and the greedy algorithm mechanism are optimized for the above problem. Considering the task execution time and resource node execution ability, different types of tasks and resources matched and addressed the long waiting time, but also led to a new problem, namely load imbalance and low resource utilization. The round-robin scheduling algorithm is more inclined to the fairness of scheduling and can address the load balancing problem. However, at the same time, it ignores the objective of execution time and the execution ability of resource nodes, leading to the decline of the overall scheduling efficiency. The basic idea of Minimum Completion Time (MCT) [59] is



to randomly assign existing tasks to the virtual machine that can make it have the earliest completion time, regardless of whether it can be completed the fastest, which may increase the running time of tasks on virtual resources.

## 2) Swarm intelligence optimization scheduling algorithm

By imitating the foraging and behavior of different organisms in nature, many swarm intelligence optimization algorithms [60,61], such as particle swarm [62–64], bacteria foraging [65], pigeons swarm [66,67], bats [68], Firefly algorithm [69], bee colony algorithm [70] are derived. The particle swarm algorithm [71,72] simulates the behavior of bird predation. For each solution, that is, each particle is a bird, and all particles are given corresponding fitness values. The corresponding state of the particles is changed by updating the formula of speed and position. Finally, the optimal solution is obtained by iterating continuously to reach the termination conditions. The ant colony algorithm simulates the behavior of ants using pheromones to forage and updates the position of the ant colony according to pheromones to find the optimal solution. The bat algorithm simulated the echolocation principle in the bat predation and updated the speed and position by adjusting the frequency and pulse loudness, and finally found the optimal solution. However, these swarm intelligence algorithms [73,74] can only rely on a single or a few adaptive value functions to optimize the scheduling problem, which still cannot satisfy multiple problems such as time-load balancing and resource utilization at the same time, and cannot effectively improve the scheduling efficiency.

## 3) Evolutionary optimization algorithm

The evolutionary optimization algorithm [75,76] is based on the evolution of nature and simulates the biological evolution process. Most evolutionary algorithms [77,78] are based on genetic algorithms, involving selection, crossover, and mutation operations. The current mainstream research direction is for the selection strategy. For example, the NSGA-II [79] algorithm mainly designs the Pareto dominant strategy, which has strong convergence ability and is used to select the solution with better performance. On this basis, the NSGA-III algorithm [80,81] further proposes the reference point strategy to address the problem of insufficient solution diversity. And many similar algorithms are mainly used to address multi-objective [82] and many-objective optimization problems [83]. Therefore, in the process of scheduling problem optimization, the multi-objective optimization algorithm [84,85] and many-objective optimization algorithm [86,87] can efficiently address complex constrained problems to maximize scheduling efficiency.

In order to satisfy the massive demand for computing resources in the era of big data under the cloud environment, and also try to reduce the resource energy consumption and computing cost of the cloud data center, the combination of task scheduling and optimization ideas [88,89] is inevitable. Due to the complexity of the scheduling problem [90] and the diversity of algorithms, the scheduling problem has been discussed by more and more scholars. In order to improve scheduling performance [91], scholars at home and abroad diverge their thinking and conduct in-depth research from different perspectives. Various intelligent scheduling strategies are provided in the cloud computing reference. And then, starting from task dependencies, multi-cloud scheduling is analyzed and summarized into two aspects: independent task scheduling and workflow scheduling. Meanwhile, existing multi-cloud scheduling strategies and various optimization algorithms [92,93] they use are studied.

### ***3.4 Independent Task Scheduling Methods***

Frincu et al. [94] considered factors such as deployment, load, runtime cost, and resource reuse during server failures. This paper proposes a scheduling algorithm to optimize high application

availability, low run-time cost, and load balancing nodes and improves the robustness of the scheduling system. The objective of high application availability refers to the ability of the application to make the task execute uninterrupted when the application fails during the execution of the task. The strategy adopted in this paper is to replicate the application components, and the greater the availability of the application, the better the performance. The author compares the proposed algorithm with the classic polling strategy, simulation results verify the effectiveness of the method.

Chen et al. [95] recognized the huge advantages of cloud computing and deployed video streaming media services on the cloud platform to achieve better results. This paper proposes an algorithm for selecting cloud providers and datacenters as video service managers in the multi-cloud environment to evaluate the performance of different video service loads. Considering cost and service quality, and comparing with a single cloud environment, the experimental results show that the multi-cloud environment is more conducive to making the best choice.

In order to ensure the authenticity of the data supplied by the provider, Geethanjali et al. [96] applied the game theory to the real-time task scheduling mechanism. And the corresponding optimization algorithm is designed while improving the authenticity of the data, with makespan and cost as the core objectives. And the simulation results are better than traditional multi-objective optimization scheduling algorithms.

Kang et al. [97] proposed a staged multi-round scheduling method to meet the objectives of high utilization and load balancing of compute nodes under the multi-cloud environment from static and dynamic aspects. The simulation results verify the effectiveness of the proposed method.

Rizvi et al. [98] believed that multi-cloud providers must consider promising predefined SLA to customers. SLA can ensure that the QoS of the customer enables the customer to assert their rights in the event of a service failure. Therefore, this paper proposes an SLA-based resource allocation auction mechanism by using Shortest Job First (SJF) scheduling. The essence of the auction mechanism is to use auctions to obtain available resources when the provider's resources are insufficient. When the resources are sufficient to complete the submitted tasks, the SJF scheduling method is adopted. The total execution time objective involved refers to the time when all tasks are actually completed. Penalty objective refers to what the cloud provider pays when it fails to meet the constraints of user expectations. Both total execution time and penalty objectives are as small as possible. Experiments show that this mechanism can reduce SLA violations and reduce job waiting time.

Miraftabzadeh et al. [99] took into account the heterogeneity of cloud resources and the constraints of diversification in a multi-cloud environment, which intensified the competition among providers. In order to support elastic and inelastic services well, a sorting method combining logarithmic function and sigmoidal function is introduced. A dynamic distributed resource allocation algorithm is proposed to minimize the cost and maximize the benefit. The revenue objective refers to maximizing the provider's revenue by maximizing the use of the available resources in the cloud environment and reducing the provider's management work. The revenue value should be as much high as possible.

To improve the multi-cloud scheduling efficiency, Hao et al. [100] modeled and addressed multi-cloud parallel tasks. The author believes that as the number of virtual machines increases, there will be more parallel tasks, that is, the two are positively correlated. Therefore, a multi-objective parallel task scheduling algorithm is proposed to optimize the waiting time of jobs in different lists. Through comparative experiments under different parameters, compared with other classical algorithms, the proposed method shows the best performance by the indicators of waiting time and response time.

By analyzing the big data of content distribution network (CDN) operation, Wang et al. [101] adopted the long-term resource deployment algorithm to satisfy the needs of users with the lowest resources and proposed a resource allocation and scheduling optimization strategy supported by the multi-cloud architecture. At the same time, the multi-cloud scaling algorithm can further schedule cloud resources from other cloud providers to address the overload problem of the cloud system. And the results verify the efficiency of the scheduling strategy from the aspects of QoS and cost.

Kang et al. [102] proposed new multi-cloud system architecture for the load scheduling problem in cloud storage resource management, taking into account the complex internal characteristics of computing nodes. At the same time, the author puts forward a dynamic scheduling method, which applied the theory of separable load and the prediction technology of node availability. Simulation experiments show that the total load processing time is reduced by 44.60%.

Panda et al. [103] believed that tasks are allocated to the cloud based on their current load, and they did not consider dividing tasks into preprocessing and processing time. Therefore, the task is further divided into two different stages preprocessing and processing. According to the division method, the author proposed three different multi-cloud scheduling algorithms and conducted simulation experiments with utilization and time as performance indicators. The experiments show the superiority of the proposed algorithm.

Jena et al. [104] used a genetic algorithm to address the scheduling problem in order to satisfy the dynamic changes in user needs and the arrangement and combination of heterogeneous resources. Firstly, this method conducted resource allocation by using a genetic algorithm. And then, the priority is set. That is, the shortest task has the highest priority. By assigning tasks to virtual machines under the multi-cloud, the shortest completion time and maximum customer satisfaction are obtained.

Roy et al. [105] believed that the multi-cloud environment was more suitable for solving large-scale tasks. The author takes the reduction of the maximum completion time and the delay as the optimization objectives. By calculating the execution cost, energy consumption, and required time of the task, a scheduling algorithm combining task priority is proposed. Simulation experiments prove that the algorithm has better performance.

Hubert Shanthan et al. [106] decomposed the multi-cloud scheduling problem into two parts: scheduling and rescheduling. A round-robin scheduling algorithm was adopted in the scheduling stage. The author found that about 15% of the task execution cost was reduced only in the rescheduling phase, so the optimization only focused on the cost problem, and the result optimization was achieved through two-phase scheduling.

Thirumalaiselvan et al. [107] believed that when the ratio of virtual machines to the number of tasks was different, the requirements for scheduling policies were also different, so the scheduling problem was considered from the perspective of computing resources and the number of tasks. For the three keywords of load, priority, and rate, the author proposed three different scheduling algorithms, and optimized energy consumption, Makespan, and delay at the same time. The delay refers to various delays generated by cloud computing resources when processing tasks submitted by users. For example, various delays are caused by tasks in the process of transmitting data, waiting for processing, and non-essential transaction processing. The smaller the delay, the higher the task processing efficiency. Compared with existing methods, the results show that the proposed algorithm can achieve higher-performance multi-cloud computing.

For some urgent tasks in a heterogeneous multi-cloud environment, Zhang et al. [108] tried to reduce the delay faced during the task execution process as much as possible. In order to reduce

the delay of task execution, different tasks in a multi-cloud datacenter will compete for edge device resources as much as possible. In this paper, a reinforcement learning framework is introduced to deal with resource competition in different datacenters. Simulation results show that the mechanism can make full use of computing resources to reduce the task delay.

Panda et al. [109] believed that a single cloud environment cannot satisfy the resource capacity problem during peak demand periods, so the multi-cloud environment is introduced in the article. The aware-allocation algorithm is proposed by improving the Min-Min and Max-Min algorithms to improve task scheduling performance. And it includes three steps, namely matching, allocation and scheduling. Finally, the author measures the performance of the algorithm from three aspects: throughput, total time, and utilization. Simulation results show the effectiveness of the algorithm.

Due to the delay in response time when files between datacenters in cloud resources are accessed by each other, Li et al. [110] proposed a data copy-aware scheduling algorithm for this situation. Among them, Bayesian network technology is used to balance system load and improve data access speed. Combine transmission calculation with transmission data, and realize resource matching based on node locality. Compared with the benchmark algorithm, the proposed algorithm has better performance.

Shanthan et al. [111] planned to introduce a multi-cloud environment for task scheduling in order to address problems such as vendor lock-in, resource unavailability, and delay. In this paper, a priority intensive meta task scheduling algorithm is proposed, the task completion time is optimized, and the priority is introduced for task filtering, in which the optimization background is offline task scheduling.

Jing et al. [112] considered the problem of massive loss of resource costs caused by the low efficiency of data-intensive applications under the multi-cloud environment. In order to reduce cost consumption, the author introduced a cloud-independent system, so that computing resources can be placed as close as possible to the execution data, and a corresponding scheduling algorithm is designed to minimize costs. During the experiment, the effectiveness of the system model is evaluated by data transfer efficiency and cost. And data transfer time refers to the inevitable need for data transfer during the execution of tasks due to bandwidth between clouds and data transfer across clouds. If the data transfer efficiency is low, it may lead to an increase in the cost of the overall model and a decrease in inefficiency. Therefore, the greater the data transfer efficiency, the better the performance. Experimental results show that cloud-independent systems and cost-aware scheduling algorithms can improve the data transmission rate and save budget costs.

Considering data privacy, such as data privacy of work, different electricity prices of private clouds, and different billing strategies/cycles of public clouds, Pasdar et al. [113] proposed a multi-cloud scheduling recommendation (MCSR) framework based on an artificial neural network to optimize cost efficiency. Using real Facebook workload data, the experiment demonstrated that the MCSR framework can efficiently schedule tasks.

Karaja et al. [114], aiming at the heterogeneity of cloud resources and considering that there should be budget constraints in task scheduling, proposed a dynamic multi-cloud task scheduling algorithm with budget constraints and conducted experiments by synthesizing datasets. Under the premise of budget constraints, the simulation results show that the maximum completion time performance of the algorithm is better.

Karaja et al. [115] proposed a hybrid bi-level dynamic bag-of-tasks scheduling optimization model (HB-DBOTSP) in view of the scheduling complexity caused by resource heterogeneity in the multi-cloud environment. This model takes into account budget constraints, and the objectives are time and execution cost that ensures load balancing among cloud resources. The experimental results verify that the HB-DBOTSP model can receive better solutions.

In order to enable users to obtain better services, Mohanraj et al. [116] established a multi-objective model under the premise of considering Makespan and cost, as well as task waiting time, delay, and throughput. After the task is scheduled for the virtual machine, it will be stored in the task queue of the virtual machine. Then the service program in the deployed virtual machine will execute the tasks in the queue in sequence according to the principle of first come first serve. Therefore, the waiting time of a task refers to the time that the task needs to wait for execution in the to-be-executed queue after it is allocated to computing resources. The shorter the waiting time, the higher the task execution efficiency. And the author proposed a corresponding algorithm for the designed multi-objective model, a multi-swarm optimization algorithm based on the traditional swarm intelligence optimization idea. Simulation experiments illustrate that the performance of the optimization model in terms of throughput and resource utilization has been improved, but with the increase in the number of tasks, the model efficiency will decrease.

In order to address the problem of trust constraints, Zhu et al. [117] proposed a new multi-cloud scheduling method—matching the multi-round allocation method, which is used to optimize all security and reliability constraints. This method first considers the impact of priority on the task sequence and satisfies safety and reliability as much as possible. At the same time, in order to improve scheduling performance, a multi-round allocation strategy is used to further minimize the time objective. The author performed simulations in CloudSim and compared them with other bio-inspired algorithms, verifying that the algorithm is more effective in reducing completion time and saving costs.

Selvapandian et al. [118] took into account that multi-cloud task scheduling requires enabling massive virtualized resources. So, they demonstrated a set of non-dominated solutions using experiments that can be applied in real-world and manufacturing work processes, and these solutions do not dominate each other. A multi-level task scheduling policy estimation model is proposed for task planning calculation in the heterogeneous multi-cloud environment. Compared with other multi-objective optimization algorithms, simulation results show that the proposed algorithm has lower costs and less time.

Hubert Shanthan et al. [119] improved the particle swarm optimization (PSO) algorithm to optimize multi-cloud multi-objective scheduling problems. It uses inertial weights and PSO speed to get a non-dominated scheduling scheme, combining local search and global search methods. The simulation experiment proves that the algorithm obtains the best performance in terms of optimizing time and task completion rate.

The task scheduling problem in the existing multi-cloud environment cannot be comprehensively considered from the perspectives of providers and users. After fully analyzing the factors that should be included in the scheduling process, Cai et al. [120] established a many-objective model that considers the needs of both providers and users. An intelligent scheduling algorithm is designed to solve the model, focusing on convergence in the early stage and in the later stage. The simulation results verify the effectiveness of the algorithm and model.

In order to obtain higher satisfaction for users, and at the same time, the benefits of providers are greater. Su et al. [121] combined the edge cloud framework with multi-cloud, designed an interaction

mechanism based on a game model between users and providers, and proposed a corresponding algorithm to decide which computing resources to use for the tasks submitted by users. Simulation results show that under the new framework, this method can meet the expectation of users and make providers benefit well.

**Table 1** summarizes the related strategies for independent task scheduling in the above-mentioned references in a timeline, and further lists the motivation and optimization objective. Finally, the types of scheduling strategies are given. And **Fig. 6** shows the proportion of various objectives involved in the multi-cloud independent task scheduling references.

**Table 1:** Summary of research reference on multi-cloud task scheduling strategy

Ref.	Motivation	Optimization objective	Strategy type
[94]	Achieve application high-availability and fault-tolerance	Load balance, cost, application availability	Improved GA
[95]	Huge storage capacity for video streaming service	QoS (Distance between users and servers and existing virtual resources) and cost	Dynamic algorithm based on changes in the number of users
[96]	Ensure authenticity of provider's information	Makespan and cost	Game theory mechanism
[97]	Achieve high utilization and load balancing among compute nodes	Load balance	Multi-round scheduling in stages
[98]	Enable customers to assert their rights with SLAs	Total execution time, Waiting Time, Penalty	Auction mechanism and Shortest Job First (SJF) scheduling
[99]	In order to maximize the use of the resource pool	Cost and revenue	Combining the sorting method of sigmoidal and logarithmic utility functions
[100]	Modeling and solving parallel tasks scheduling	Waiting time, Makespan	Multi-objective ZERO-ONE scheduling
[101]	Meet the needs of users with the lowest resources	QoS (Usability, Availability, Responsiveness, Distance) and cost	based on Auto Regressive Integrated Moving Average (ARIMA) model to forecast requests
[102]	The load scheduling problem	Load balance	Dynamic strategy based on the divisible load theory and node availability prediction techniques

(Continued)

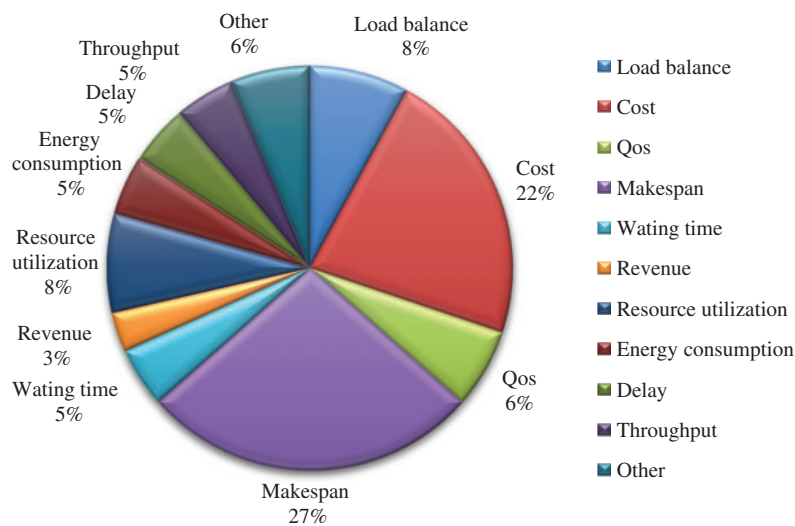
**Table 1 (continued)**

Ref.	Motivation	Optimization objective	Strategy type
[103]	Consider dividing tasks into preprocessing and processing time	Makespan and resource utilization	Scheduling algorithm based on task partitioning
[104]	Dynamic changes for requirements and resources	Makespan and QoS (Expected time, estimated cost, resource waiting time)	GA and shortest task first strategy
[105]	Handling large-scale task scheduling	Makespan, cost and energy consumption	A priority-based hybrid task scheduling policy
[106]	Dwindle the execution costs of the VMs	Makespan, cost, resource utilization	Round-robin scheduling algorithm and rescheduling
[107]	The impact of quantity (tasks and virtual machines) on scheduling problems	Makespan, delay, energy consumption	Scheduling strategy based on load, priority and rate
[108]	To reduce delay when dealing with urgent tasks	Delay	Multi-agent reinforcement learning (MARL) framework
[109]	Multi-cloud scheduling performance	Makespan, throughput and resource utilization	Improved algorithm-based Min-Min and Max-Min
[110]	To reduce response delay between multi-cloud	Makespan and load balance	Replica replication strategy
[111]	Solve the scheduling problem of independent batch tasks in a static environment	Makespan	Priority is introduced for task filtering
[112]	Reduce costs and speed up the operation of data-intensive applications between clouds	Data transfer efficiency, cost	Cloud-independent PIVOT system and cost-aware scheduling
[113]	Optimize cost efficiency explicitly taking into account data privacy	Cost	Simulated annealing (SA) and a recommender using artificial neural networks (ANNs)
[114]	Task scheduling under budget constraints	Makespan and cost	Undergoes variation mechanisms using a local search
[115]	Dynamic scheduling to balancing load	Makespan and cost	Bi-level optimization based on Tabu Search algorithm and Genetic Algorithm

(Continued)

**Table 1 (continued)**

Ref.	Motivation	Optimization objective	Strategy type
[116]	Enhanced Quality of Services for a multi-cloud environment	Throughput, delay, Makespan, waiting time, resource utilization	Multi-swarm optimization algorithm
[117]	Trust constraints problem in a heterogeneous multi-cloud environment	Makespan and cost	Matching and multi-round allocation scheduling algorithm
[118]	Enables virtual resource scheduling to be more efficient	Cost and Makespan	Multilevel task scheduling policy(First in and First out (FIFO) + Continuous Connected Settlement (CLS) + Round Robin scheduling)
[119]	Improved maximum completion time and completion rate for tasks	Makespan and cost	Improved particle swarm optimization (PSO)
[120]	Improve scheduling efficiency consider both users and providers	Makespan, cost, throughput, energy consumption, resource utilization, and load balance	Many-objective optimization algorithm
[121]	Consider the constraints of different tasks and maximize user satisfaction	QoS (delay, energy consumption and cost) and revenue	The Stackelberg game



**Figure 6:** The proportion of objectives involved in the independent task scheduling references



Table 1 sorts out the existing references on typical multi-cloud independent task scheduling in chronological order. Due to the heterogeneous advantages of the multi-cloud environment, it can be concluded that users can be provided with diversified virtual resources so that both users and providers can put forward more requirements in the process of task execution. For example, from the perspective of users and providers, references [116,120] designed the task scheduling model with five and six objectives respectively, although the core objectives of most of the references are still time and cost, such as references [117–119]. In order to obtain a better scheduling scheme in the task scheduling process, the scheduling algorithm is also constantly improving, which can be roughly divided into heuristic scheduling algorithm, traditional scheduling algorithm, and hybrid scheduling algorithm combining the advantages of the two. Such as references [94,119] to improve the GA algorithm and the PSO algorithm, the references [98,109] adopted the traditional scheduling algorithms, and the references [104,115] use the hybrid scheduling algorithm. In order to obtain better scheduling result, some scholars have also introduced edge computing and fog computing frameworks into the multi-cloud environments to further expand computing resources, such as references [108,121]. At the same time, the security and reliability issues caused by massive virtual resources in multi-cloud platforms have also been paid attention to by scholars. For example, references [94,117] considered the security issues of multi-cloud task scheduling and the reliability of task execution. What’s more, neural network technology [113] has also been applied to the task scheduling process. Fig. 6 shows the percentage of different objectives in the multi-cloud independent task scheduling references. It can be seen that cost and makespan are the main concerns of most scheduling references. Secondly, in order to make both users and providers obtain better benefits, the references also pay more attention to the three objectives of load balance, resource utilization and QoS.

Table 2 mainly analyzes and summarizes the scheduling constraints and test environments involved in the multi-cloud independent task scheduling references. Fig. 7 shows the percentage of different constraints in the independent task scheduling references, which can more intuitively reflect that deadline and budget constraints are the most critical in multi-cloud independent task scheduling. At the same time, for the smooth execution of task scheduling, some references also consider reliability and security constraints. For some user-specific requirements, task priority constraints occasionally need to be considered in the scheduling process.

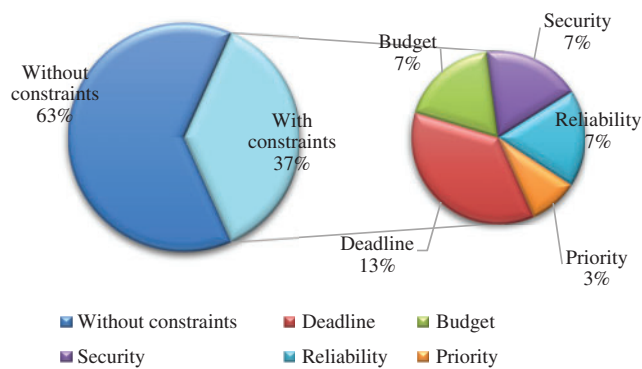
**Table 2:** Constraints and test environments covered in the independent task scheduling references

Ref.	Constraint					Testing environment
	Deadline	Budget	Security	Reliability	Priority	
[94]				✓		Real environment
[95]						Simulation
[96]	✓					Simulation (Cloudsim)
[97]						Simulation (MATLAB)
[98]	✓					Simulation (CloudSim)
[99]						Simulation
[100]						Simulation
[101]						Simulation (OpenStack)
[102]						Simulation
[103]						Simulation (MATLAB)

(Continued)

**Table 2 (continued)**

Ref.	Constraint					Testing environment
	Deadline	Budget	Security	Reliability	Priority	
[104]						Simulation (MATLAB)
[105]	✓					Simulation (MATLAB)
[106]						Simulation (CloudSim)
[107]						Simulation (Netbeans)
[108]						Real environment
[109]						Simulation (MATLAB)
[110]						Simulation (Java)
[111]					✓	Simulation (CloudSim)
[112]						Real environment
[113]	✓		✓			Simulation
[114]		✓				Simulation
[115]		✓				Simulation
[116]						Simulation (NetBeans)
[117]			✓	✓		Simulation (CloudSim)
[118]						Simulation
[119]						Simulation (CloudSim)
[120]						Simulation (MATLAB)
[121]						Simulation



**Figure 7:** The proportion of constraints involved in the independent task scheduling references

### 3.5 Workflow Scheduling Methods

In the heterogeneous multi-cloud environment, Panda et al. [122] proposed new scheduling algorithms, which aim to minimize completion time while maximizing average cloud utilization. These algorithms include one-stage scheduling and two-stage scheduling. They used various benchmarks and synthetic data sets to conduct rigorous experiments on the proposed method, and the effectiveness is verified.

Taking into account the massive amount of data in multi-cloud workflow scheduling and the complex dependencies between data. Sooezi et al. [123] applied the communication idea to multi-cloud workflow scheduling and designed a new algorithm with user deadlines as the constraint. The algorithm changes the definition of part of the critical path (PCP), and simulation experiments prove that the algorithm optimizes the cost of the task.

The dynamic multi-cloud workflow requires various cloud service algorithms to facilitate the completion of workflow task scheduling. Nandhakumar et al. [124] summarized and reviewed the cloud workflow scheduling strategies proposed in some existing references. The algorithms are introduced from the perspectives of heuristics and meta-heuristics respectively so that readers have a better understanding of these existing algorithms.

Lin et al. [125] applied partial of the critical path algorithm to the workflow and designed a scientific multi-cloud workflow scheduling algorithm. The algorithm can minimize the cost of the task under the deadline constraint required by the user. Simulation experiments prove the advantages of this algorithm.

Maheshwari et al. [126] proposed a new scheduling strategy to minimize the cost of data transmission based on the impact of virtual resources on the execution performance of multi-cloud workflow tasks. The author used the proposed multi-site multi-cloud scheduling strategy to achieve lower cost and effectiveness of resource utilization solutions. The experimental results verify that the algorithm has greater advantages in terms of network throughput and task cost.

Gupta et al. [127] further studied the influence of transmission time on multi-cloud workflow scheduling. The author divides the workflow scheduling into two stages according to the priority of the task and designs a two-stage workflow scheduling algorithm, which selects the virtual machine by calculating the priority of the task. The author takes the maximum completion time and average cloud utilization as the objectives, and simulates the standard scientific workflow. The experimental results show that the scheme has good effects.

Suri et al. [128] took into account that multi-cloud task scheduling needs to reasonably assign tasks to the most appropriate resources. In this paper, the tasks are grouped according to their execution time and then sorted in ascending order according to the execution time. The authors argue that the shortest tasks should be executed first, which reduces the largest completion time. The average turnaround time of the model designed in this paper refers to the sum of the average execution time and the waiting time of the task, and the completion time refers to the total time it takes to complete the execution of different grouped tasks. Therefore, the smaller the average turnaround time and the completion time, the better the performance. Compared with other sorting methods, the proposed model receives quality solutions.

Panda et al. [129] designed four scheduling algorithms based on normalization technology for the heterogeneous multi-cloud environment. It should be noted that the four algorithms designed are suitable for online scheduling in different situations, and there is no comparison between them. The author uses synthetic datasets and benchmark datasets to simulate the algorithm. Simulation results show that the algorithm has better performance in terms of time and utilization indicators.

Hu et al. [130] further improved the quality of customer service under multi-cloud workflow scheduling. The author believes that reliability is a key indicator of service quality. Therefore, they designed an algorithm that is superior to traditional particle swarms to reduce the maximum completion time and cost. The corresponding coding strategy also considers the task execution

position and the task sequence of data transmission. A simulation experiment based on a real scientific workflow model shows that the proposed algorithm is greatly improved.

Li et al. [131] aimed at the problems in managing and executing big data scientific workflows to process stream data sets in the multi-cloud environment. To satisfy user constraints while improving cloud throughput, a new heuristic algorithm is designed: a scheduling algorithm that identifies global bottlenecks and maximizes throughput. Simulation experiments prove the effectiveness of the algorithm.

Mohammadi et al. [132] designed a new method to determine the most stable solution of the weighted sum in order to obtain a better and more stable multi-cloud multi-objective workflow scheduling solution, taking the maximum completion time and the cost of the task as the optimization objectives. And the weighted minimum-maximum as a post-optimum analysis. Simulation experiments show that the algorithm performs better.

Chen et al. [133] considered that large-scale scientific workflows have complex task dependencies, leading to large execution costs. In order to better implement schedules, the authors proposed an online multi-cloud adaptive scientific workflow scheduling algorithm. A large number of simulation experiments show that the algorithm reduces the execution cost of the workflow and improves the resource utilization of the cloud environment under the constraints of some hardware devices.

Farid et al. [134] considered that the tasks under the multi-cloud workflow are intricate and dense, and proposed an optimization algorithm that is superior to traditional particle swarms. The algorithm optimizes the task completion time and resources utilization rate for the transmission sequence of tasks. The resource utilization objective in this paper is expressed by the ratio of the virtual machine processing power required by the task to the actual allocation. And the resource utilization is as large as possible. For objectives such as utilization rate and cost required for task execution, simulation experiments show that compared with the particle swarm algorithm, the improved algorithm proposed by the author is better.

Tang et al. [135] constructed a fault-tolerant workflow scheduling framework for multi-cloud systems considering the existing multi-billing mechanism, virtual resource heterogeneity, and system reliability. Weibull distribution is used to analyze the reliability and risk rate of tasks and repeat tasks with high-risk rates to increase the reliability of tasks. At the same time, a multi-cloud scheduling algorithm combined with fault-tolerant ideas is designed. This algorithm reduces the total completion time and increases the reliability and cost of workflow task execution. Simulation experiments prove the advantages of this algorithm.

Ulabedin et al. [136] pointed out that the datacenter is facing huge data transmission. And the movement of data will lead to a longer execution time for multi-cloud workflows and increase the cost of task completion. In order to overcome the difficulty of data transmission, the author introduces the critical path to workflow scheduling. The data movement objective refers to the amount of movement of datasets between different datacenters or within the same datacenter. Fewer movements can reduce the execution cost and time for tasks. Simulation experiments prove that the author's proposed method optimizes the cost and maximum completion time, and reduces the amount of overall data movement.

Wang et al. [137] realized that there is a certain connection between workflows, and these connections will lead to complex or simple data communication between tasks. In order to reduce the loss caused by data communication, this paper introduces the clustering coefficient to reasonably slice the tasks in the workflow, and dynamically allocate the tasks to the appropriate cloud provider to reduce the time and cost increase caused by data communication.

Mohammadzadeh et al. [138] analyzed the respective advantages of the seagull and grasshopper optimization algorithms and combined the advantages of the two to construct a hybrid multi-objective optimization algorithm. This paper uses this algorithm to optimize multi-cloud workflow scheduling problems with the objective of time, energy consumption, cost, and throughput. Multiple scientific workflow datasets are used for testing on the simulation experiment platform, and the experimental results show that the algorithm has advantages in multiple indicators.

Sujana et al. [139] believed that some inexperienced users are in a fuzzy state when using a multi-cloud environment and thus introduced a fuzzy decision-making model. Based on the three objectives of time, cost, and trust value, this paper defines different fuzzy membership functions and makes collective decisions. Finally, an intelligent cloud agent based on fuzzy logic is proposed to solve the workflow scheduling problem in the multi-cloud environment. The trust objective refers to the trust degree of the task to the computing resource, that is, the probability of selecting the computing resource to execute the task. In this paper, the user's trust degree for different resources is determined by the recommendation trust degree of other users and self-historical evaluation of different resources. The larger the objective value, the easier it is to be selected.

Chakravarthi et al. [140] considered that the cloud environment has the characteristics of resource heterogeneity and dynamic change. Also, to ensure that the workflow in the multi-cloud environment can be executed smoothly within the expectations of users, a highly reliable workflow scheduling scheme based on user time, cost, and other budgets is proposed. Reliability refers to selecting the most reliable computing resources for tasks, thereby reducing the possibility of task execution failures due to various failures. The strategy used in this paper is to select computing resources for tasks by first calculating the probability of correct execution of tasks on different resources. The strategy first calculates the probability of tasks being executed correctly on different resources, and finally selects the most reliable resource under the premise of user expectation constraints.

Table 3 summarizes the multi-cloud workflow scheduling strategies in the above references, and further lists the motivation, optimization objectives. Finally, the type of scheduling strategy is given.

**Table 3:** Summary of research reference on the multi-cloud workflow scheduling strategy

Ref.	Motivation	Optimization objective	Strategy type
[122]	Minimize completion time under cloudy conditions while maximizing average cloud utilization	Makespan, resource utilization	Independent-phase scheduling and two-phase scheduling
[123]	Minimize the cost of workflow execution while meeting a user defined deadline	Makespan and cost	Extended Partial Critical Paths (PCP)
[124]	Review workflow multi-cloud scheduling algorithms	—	—
[125]	To better satisfy the users' quality of service requirements	Makespan and cost	Multi-Cloud Partial Critical Paths (MCPCP)
[126]	Balance workloads between sites and minimize data execution time	Makespan, resource utilization	Multi-site performance prediction

(Continued)

**Table 3 (continued)**

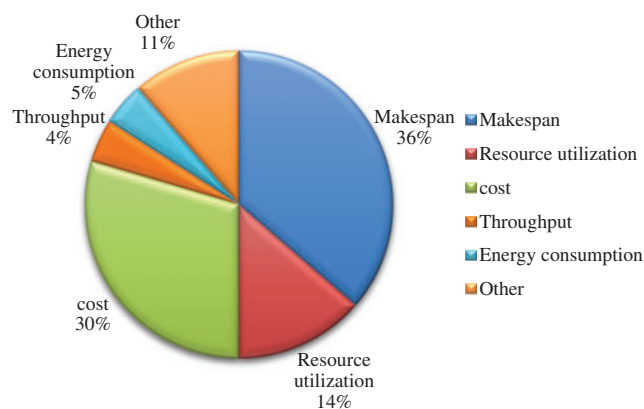
Ref.	Motivation	Optimization objective	Strategy type
[127]	Makes an exclusive mapping between tasks and virtual machines for better performance	Makespan, resource utilization	Priority scheduling based on task node transmission time
[128]	Allocate tasks to the most appropriate resources	Waiting time, average turnaround time, completion time and makespan	Shortest Job First Strategy (SJF)
[129]	Balance multiple cloud scheduling workloads	Makespan and resource utilization	Scheduling algorithm based on normalization technology
[130]	Satisfy reliability constraints	Makespan, cost	Improved Particle Swarm Optimization (PSO)
[131]	Manage and execute big data scientific workflows in multi-cloud environment to process streaming datasets	Throughput, cost	PCP + Prioritization of virtual machines
[132]	Obtain a stable solution that meets both the minimum cost and the maximum completion time	Cost and makespan	Integer linear programming (ILP)
[133]	Handle task scheduling with deadlines for hardware devices	Cost, resource utilization	Adaptive resource allocation and consolidation
[134]	Workflow scheduling under Reliability constraints	Makespan, cost, resource utilization	PSO + fuzzy logic
[135]	Improve the scientific applications execution reliability	Cost, Makespan	Weibull distribution and Task cb-level sorting
[136]	Reduce the impact of data movement on workflow execution time and budget costs	Makespan, cost, energy consumption and data movement	Critical path data locality (R-PCP) technology
[137]	Reduce the data communication loss of workflow	Makespan and cost	Workflow slicing based on clustering coefficient
[138]	Generating pareto optimal solutions in multi-cloud work applications	Makespan, energy consumption, cost, and throughput	Combining the strengths of seagull and grasshopper optimization algorithms

(Continued)

**Table 3 (continued)**

Ref.	Motivation	Optimization objective	Strategy type
[139]	Satisfy the user’s fuzzy needs	Makespan, cost and trust.	Fuzzy membership function for each objective
[140]	Tasks can be reliably executed under budget constraints	Makespan, cost	Min-max normalization, Grouping workflow tasks

Table 3 sorts out the existing typical multi-cloud workflow scheduling reference and sorts them by time. Due to the complex dependencies among workflow tasks, not only the complexity of multi-cloud resources, but also the execution order of tasks needs to be considered in the task scheduling process. And a large amount of data transmission time is required during the execution process. Therefore, in order to improve the efficiency of workflow execution, the references [127,137] designed the scheduling algorithm from the perspective of transmission time. At the same time, it can be concluded that the scheduling strategy of the partial critical path is often used in the scheduling process of workflow. For example, references [125,131,136] adopted the improved version of the partial critical path. The types of optimization algorithms used in the workflow scheduling process include heuristic algorithms, traditional scheduling algorithms, and numerical algorithms. As references [130,134,138] use heuristic algorithms, the reference [140] used the traditional scheduling algorithm, and the reference [134] uses the integer linear programming method. In addition, the references [134,135,140] further considered the security and reliability issues that may arise due to the complexity of resources in the multi-cloud environment and the complex dependencies of workflows. Fig. 8 shows the percentage of different objectives in the multi-cloud workflow scheduling references. Obviously, makespan and cost objectives are also mainly considered in multi-cloud workflow scheduling. At the same time, in order to make full use of cloud resources, resource utilization objective has also attracted much attention.



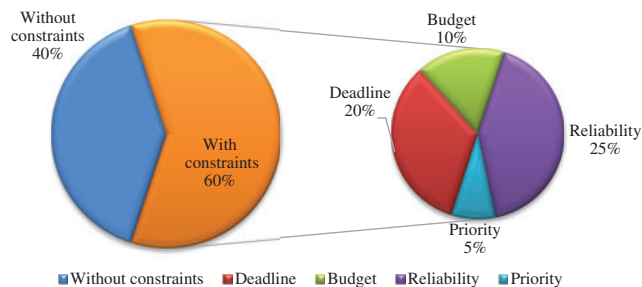
**Figure 8:** The proportion of objectives involved in the workflow scheduling references

Table 4 mainly analyzes and summarizes the scheduling constraints and test environments involved in the multi-cloud workflow scheduling references. Fig. 9 depicts the different proportions of constraints in the multi-cloud workflow scheduling reference. It can be intuitively reflected that most

of the reference on workflow scheduling considers constraints. Deadline and reliability constraints are the focus, and budget constraints also occupy a large proportion in workflow scheduling.

**Table 4:** Scheduling constraints and test environments covered in the workflow scheduling references

Ref.	Constraint					Testing environment
	Deadline	Budget	Security	Reliability	Priority	
[122]						Simulation (MATLAB)
[123]	✓					Simulation
[125]	✓					Simulation
[126]						Real environment
[127]						Simulation (MATLAB)
[128]						Simulation (MATLAB)
[129]						Simulation (MATLAB)
[130]				✓		Simulation (Python)
[131]		✓				Simulation (Java)
[132]						Simulation (CPLEX Optimization Studio)
[133]	✓					Simulation (Python)
[134]				✓		Simulation (Workflowsim)
[135]				✓		Simulation (CloudSim)
[136]	✓	✓				Simulation (Workflowsim)
[137]						Simulation
[138]						Simulation (CloudSim + Workflowsim)
[139]			✓	✓		Simulation (CloudSim)
[140]	✓			✓		Simulation (CloudSim)

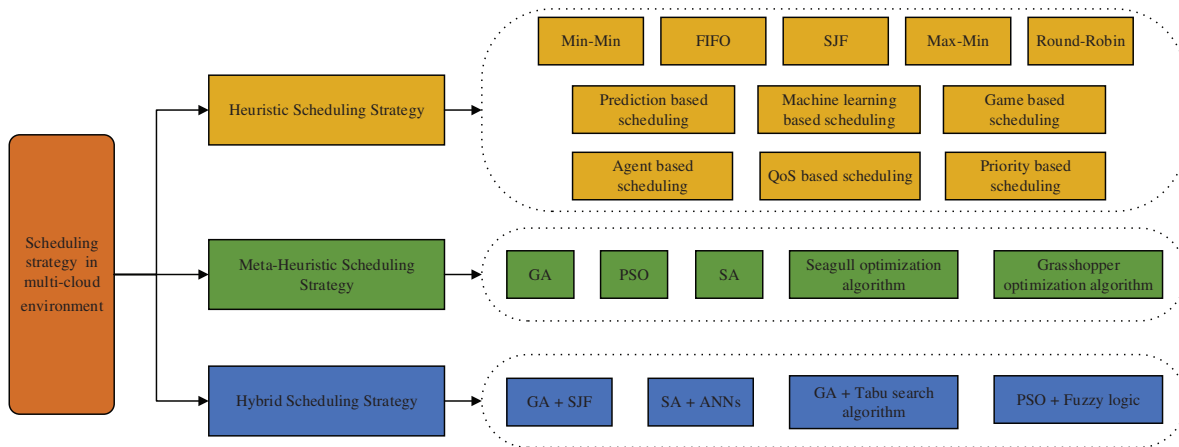


**Figure 9:** The proportion of constraints involved in the workflow scheduling references



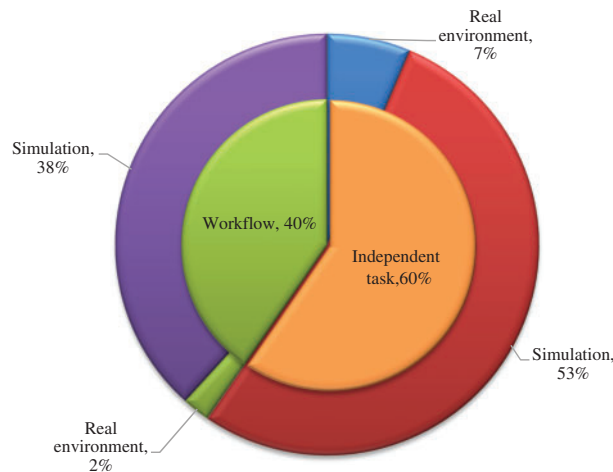
### 3.6 Summary and Classification

Tables 1 and 3 provide an analysis of the strategies used in different multi-cloud task scheduling references. By summarizing the above scheduling strategies, we mainly divide them into heuristic scheduling strategies, meta-heuristic scheduling strategies and hybrid scheduling strategies. The detailed classification and summary can be shown in Fig. 10.



**Figure 10:** Classification of scheduling strategy in multi-cloud computing

Tables 2 and 4 summarize the test environments of different references, and Fig. 11 depicts the proportion of different test environments in the multi-cloud independent task and workflow scheduling references. Obviously, most of the experiments are carried out in the simulation environment. Since the experiment in the real cloud environment is hard to achieve and will consume huge costs, the simulation environment provided by various simulation tools can easily realize various functions to satisfy all kinds of needs for cloud environment experiments.



**Figure 11:** Classification of scheduling strategy in multi-cloud computing

Table 5 categorizes the above references according to journals and conferences and sorts out the journal names (conference names), years, and their publishers. It includes 28 journal articles and 19 conference articles.

**Table 5:** Classification and summary of references

	Ref.	Journal/Conference	Year	Publisher
	[95]	Tsinghua Science and Technology	2013	Tsing Hua University
	[122]	Journal of Supercomputing	2015	Springer
	[126]	Future Generation Computer Systems	2016	Elsevier
	[100]	KSII Transactions on Internet and Information Systems	2017	Korea Society of Internet Information
	[102]	Journal of Parallel and Distributed Computing		Academic Press Inc.
	[130]	Journal of Network and Computer Applications		
Journal	[129]	Information Systems Frontiers	2018	Springer
	[103,104]	Arabian Journal for Science and Engineering		
	[105]	Advances in Intelligent Systems and Computing		
	[110,132]	Journal of Supercomputing		
	[107]	Cluster Computing-The Journal of Networks Software Tools and Applications	2019	Springer
	[109]	Information Systems Frontiers		
	[133,134]	IEEE ACCESS	2020	IEEE
	[116]	Soft Computing		
	[118]	DISTRIBUTED AND PARALLEL DATABASES		Springer
	[138]	Journal of Ambient Intelligence and Humanized Computing		
	[136]	Journal of Supercomputing	2021	
	[117]	IEEE-CAA Journal of Automatica Sinica		
	[120]	IEEE Internet of Things Journal		IEEE
	[135]	IEEE Transactions on Cloud Computing		
	[137]	Journal of Tongji University		Tongji University
	[121]	Computer Networks		Elsevier
	[139]	EAI/Springer Innovations in Communication and Computing	2022	Springer
	[140]	Cluster Computing		

(Continued)

**Table 5 (continued)**

	Ref.	Journal/Conference	Year	Publisher
	[94]	UCC 2011	2011	IEEE
	[96]	ICRTIT 2014	2014	IEEE
	[97]	UCC 2014		
	[123]	2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)	2015	IEEE
	[124]	ICACCS 2015		
	[125]	IPDPSW 2015		
Conference	[98]	ICCCS 2016	2016	CRC Press/Balkema
	[99]	SoSE 2016		IEEE
	[127]	ICCCA 2016	2017	Springer
	[128]	ICICCT 2017		
	[101]	2017 IEEE International Conference on Services Computing (SCC)	2018	IOP Publishing
	[106]	NCCI 2018		
	[131]	16th IEEE International Symposium on Parallel and Distributed Processing with Applications	2019	IOP Publishing
	[111]	NCCI 2019		
	[108]	GLOBECOM 2019	2020	IEEE
	[112]	NOMS 2020–2020 IEEE/IFIP Network Operations and Management Symposium		
[114]	2020 International Multi-Conference on: Organization of Knowledge and Advanced Technologies (OCTA)	2021	Springer	
[113]	ICONIP 2020			
[115]	HIS 2020			
	[119]	ICACCS 2021	2021	IEEE

In order to facilitate readers to have a more intuitive view of the source of the reference mentioned above. We classify the literatures analyzed in Table 5 according to their publishers. Fig. 12 shows that most of the multi-cloud scheduling reference is published in IEEE and springer publications. At the same time, Fig. 13 analyzes the number of journals and conference papers published each year from 2011 to 2022.

The proportion of references from different publishers

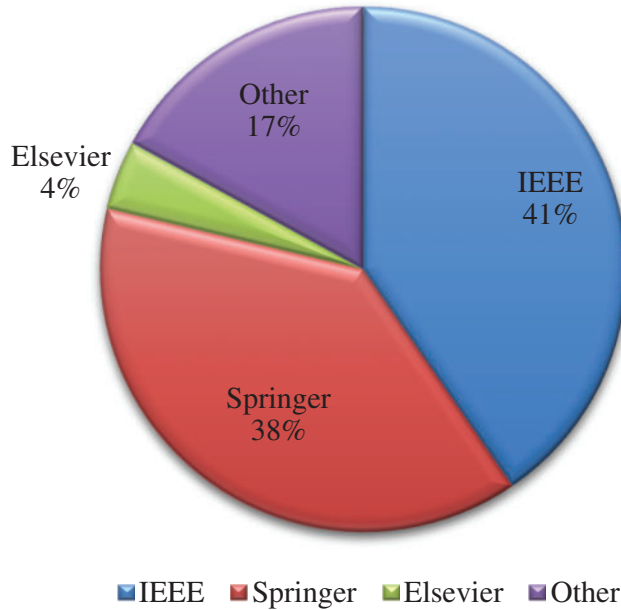


Figure 12: The proportion of references from different publishers

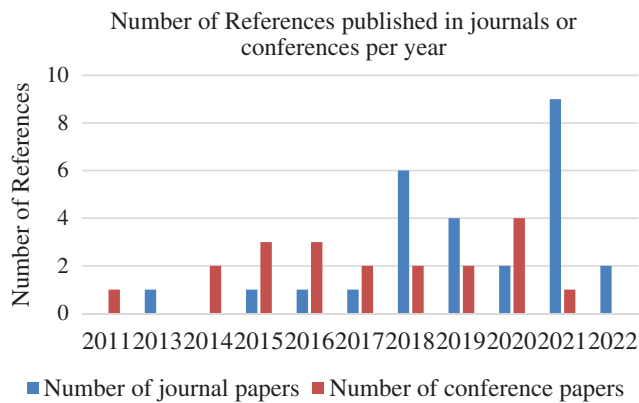
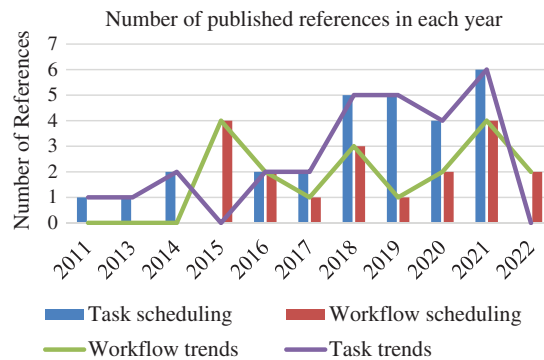


Figure 13: Number of journal and conference papers published annually on multi-cloud task scheduling

Fig. 14 depicts the trend of independent task and workflow scheduling related references published under the multi-cloud environment in recent years. The figure shows that task scheduling under the multi-cloud environment has received extensive attention in recent years, and in the early days, scholars mainly focused on the independent scheduling problem. In recent years, the references on workflow scheduling have gradually increased, indicating that the complex dependencies between tasks have been paid more and more attention. However, in terms of overall quantity, the research references on multi-cloud workflow scheduling are still scarce. Therefore, the follow-up research on workflow scheduling can be further in-depth.



**Figure 14:** Number of references per year on multi-cloud workflows and independent task scheduling

#### 4 Conclusion and Future Work

This paper makes a detailed review and summary of the independent tasks and workflow scheduling references in the multi-cloud environment based on the dependencies between tasks. Firstly, the characteristics and advantages of the multi-cloud environment are discussed, and the challenges faced by multi-cloud task scheduling are analyzed. In addition, various algorithm types, core objectives and constraints of multi-cloud scheduling are summarized, and then the existing multi-cloud scheduling references are analyzed in detail, including research motivations, scheduling strategies, constraints, test environment and optimization objectives. Finally, the research trend of the latest and representative research references about multi-cloud task scheduling in recent years is analyzed. On this basis, the direction for the next more in-depth research is proposed.

- 1) The rapid increase in the amount of data submitted by users will inevitably bring greater security risks to the operation of multi-cloud scheduling, and will also reduce the probability of tasks being reliably executed. How to ensure data security and task execution reliability in the modeling process is still an important research topic.
- 2) There are many scheduling methods designed to improve scheduling performance. However, the real-time and dynamic characteristics of task scheduling are more suitable for actual scheduling.
- 3) Optimization problems pay more attention to knowledge transfer and experience learning, so extracting the similarity of tasks in the scheduling process and performing knowledge interaction can effectively improve the scheduling efficiency, which will be a meaningful research topic.

**Funding Statement:** This work is supported by Science and Technology Development Foundation of the Central Guiding Local under Grant No. YDZJSX2021A038, the National Natural Science Foundation of China under Grant No. 61806138, China University Industry-University-Research Collaborative Innovation Fund (Future Network Innovation Research and Application Project) under Grant No. 2021FNA04014.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

1. Khabbaz, M., Assi, C. M. (2018). Modelling and analysis of a novel deadline-aware scheduling scheme for cloud computing data centers. *IEEE Transactions on Cloud Computing*, 6(1), 141–155. DOI 10.1109/TCC.2015.2481429.
2. Zhang, Y. M., Lan, X. L., Ren, J., Cai, L. (2020). Efficient computing resource sharing for mobile edge-cloud computing networks. *IEEE/ACM Transactions on Networking*, 28(3), 1227–1240. DOI 10.1109/TNET.2020.2979807.
3. Cui, Z. H., Zhang, Z. X., Hu, Z. M., Geng, S. J., Chen, J. J. (2021). A many-objective optimization based intelligent high performance data processing model for cyber-physical-social systems. *IEEE Transactions on Network Science and Engineering*. DOI 10.1109/TNSE.2021.3073911.
4. Cai, X. J., Geng, S. J., Zhang, J. B., Wu, D., Cui, Z. H. et al. (2021). A sharding scheme-based many-objective optimization algorithm for enhancing security in blockchain-enabled industrial internet of things. *IEEE Transactions on Industrial Informatics*, 17(11), 7650–7658. DOI 10.1109/TII.2021.3051607.
5. Yadav, A. M., Tripathi, K. N., Sharma, S. C. (2022). An enhanced multi-objective fireworks algorithm for task scheduling in fog computing environment. *Cluster Computing*, 25, 983–998. DOI 10.1007/s10586-021-03481-3.
6. Yadav, A. M., Tripathi, K. N., Sharma, S. C. (2022). A bi-objective task scheduling approach in fog computing using hybrid fireworks algorithm. *The Journal of Supercomputing*, 78, 4236–4260. DOI 10.1007/s11227-021-04018-6.
7. Zhang, J. W., Li, T., Ying, Z. B., Ma, J. F. (2022). Trust-based secure multi-cloud collaboration framework in cloud-fog-assisted IoT. *IEEE Transactions on Cloud Computing*. DOI 10.1109/TCC.2022.3147226.
8. Wang, Y. T., Chen, I. R., Wang, D. C. (2015). A survey of mobile cloud computing applications: Perspectives and challenges. *Wireless Personal Communications*, 80(4), 1607–1623. DOI 10.1007/s11277-014-2102-7.
9. Cui, Z. H., Jing, X. C., Zhao, P., Zhang, W. S., Chen, J. J. (2021). A new subspace clustering strategy for AI-based data analysis in IoT system. *IEEE Internet of Things Journal*, 8(16), 12540–12549. DOI 10.1109/JIOT.2021.3056578.
10. Cai, X. J., Lan, Y., Zhang, Z. X., Wen, J., Cui, Z. H. et al. (2021). A many-objective optimization based federal deep generation model for enhancing data processing capability in IoT. *IEEE Transactions on Industrial Informatics*. DOI 10.1109/TII.2021.3093715.
11. Xiang, S., Xing, L. N., Wang, L., Zhou, Y. Q., Peng, G. S. (2021). Enhanced pigeon inspired optimisation approach for agile earth observation satellite scheduling. *International Journal of Bio-Inspired Computation*, 17(3), 131–141. DOI 10.1504/IJBIC.2021.114863.
12. Luo, Y., Li, W. F., Yang, W. C., Fortino, G. (2021). A real-time edge scheduling and adjustment framework for highly customizable factories. *IEEE Transactions on Industrial Informatics*, 17(8), 5625–5634. DOI 10.1109/TII.2020.3044698.
13. Shang, L., Shang, Y. Z., Hu, L. Z., Li, J. L. (2020). Performance of genetic algorithms with different selection operators for solving short-term optimized reservoir scheduling problem. *Soft Computing*, 24(9), 6771–6785. DOI 10.1007/s00500-019-04313-8.
14. Shen, Y. D., Yu, L. Q., Li, J. P. (2022). Robust electric vehicle routing problem with time windows under demand uncertainty and weight-related energy consumption. *Complex System Modeling and Simulation*, 2(1), 18–34. DOI 10.23919/CSMS.2022.0005.
15. Cechova, P., Kubala, M. (2021). Comparison of three nature inspired molecular docking algorithms. *International Journal of Bio-Inspired Computation*, 17(1), 34–41. DOI 10.1504/IJBIC.2021.113362.
16. Wang, L., Pan, Z. X., Wang, J. J. (2021). A review of reinforcement learning based intelligent optimization for manufacturing scheduling. *Complex System Modeling and Simulation*, 1(4), 257–270. DOI 10.23919/CSMS.2021.0027.

17. Zhang, P. Y., Zhou, M. C. (2018). Dynamic cloud task scheduling based on a two-stage strategy. *IEEE Transactions on Automation Science and Engineering*, 15(2), 772–783. DOI 10.1109/TASE.2017.2693688.
18. Li, X. Y., Ma, H. D., Yao, W. B., Gui, X. L. (2018). Data-driven and feedback-enhanced trust computing pattern for large-scale multi-cloud collaborative services. *IEEE Transactions on Services Computing*, 11(4), 671–684. DOI 10.1109/TSC.4629386.
19. Masdari, M., Zangakani, M. (2020). Efficient task and workflow scheduling in inter-cloud environments: Challenges and opportunities. *Journal of Supercomputing*, 76(1), 499–535. DOI 10.1007/s11227-019-03038-7.
20. Wu, X. L., Cao, Z., Wu, S. M. (2021). Real-time hybrid flow shop scheduling approach in smart manufacturing environment. *Complex System Modeling and Simulation*, 1(4), 335–350. DOI 10.23919/CSMS.2021.0024.
21. An, J., Xu, L. Y., Fan, Z., Wang, K. F., Deng, Q. et al. (2021). PSO-based optimal online operation strategy for multiple chillers energy conservation. *International Journal of Bio-Inspired Computation*, 18(4), 229–238. DOI 10.1504/IJBIC.2021.119999.
22. Gao, D., Wang, G. G., Pedrycz, W. (2020). Solving fuzzy job-shop scheduling problem using DE algorithm improved by a selection mechanism. *IEEE Transactions on Fuzzy Systems*, 28(12), 3265–3275. DOI 10.1109/TFUZZ.91.
23. Parvande, S., Boroomand, M., Boroumand, F., Soltani, P. (2021). A modified single and multi-objective bacteria foraging optimisation for the solution of quadratic assignment problem. *International Journal of Bio-Inspired Computation*, 17(1), 1–13. DOI 10.1504/IJBIC.2021.113354.
24. Han, X., Han, Y. Y., Chen, Q. D., Li, J. Q., Sang, H. Y. et al. (2021). Distributed flow shop scheduling with sequence-dependent setup times using an improved iterated greedy algorithm. *Complex System Modeling and Simulation*, 1(3), 198–217. DOI 10.23919/CSMS.2021.0018.
25. Ma, X. T., Zhao, J. H., Gong, Y. (2021). Joint scheduling and resource allocation for efficiency-oriented distributed learning over vehicle platooning networks. *IEEE Transactions on Vehicular Technology*, 70(10), 10894–10908. DOI 10.1109/TVT.2021.3107465.
26. Konda, R. S., Panwar, K. L., Panigrahi, K. B., Kumar, R., Gupta, V. (2021). Binary fireworks algorithm application for optimal schedule of electric vehicle reserve in traditional and restructured electricity markets. *International Journal of Bio-Inspired Computation*, 18(1), 38–48. DOI 10.1504/IJBIC.2021.117430.
27. Zhao, X. J., Su, J. H., Cai, J. H., Yang, H. F., Xi, T. T. (2021). Vehicle anomalous trajectory detection algorithm based on road network partition. *Applied Intelligence*, 52(8), 8820–8838. DOI 10.1007/s10489-021-02867-5.
28. Pen, Q., Wu, H. S., Xue, R. S. (2021). Review of dynamic task allocation methods for UAV swarms oriented to ground targets. *Complex System Modeling and Simulation*, 1(3), 163–175. DOI 10.23919/CSMS.2021.0022.
29. Wang, J., Na, Z. Y., Liu, X. (2021). Collaborative design of multi-UAV trajectory and resource scheduling for 6G-enabled Internet of Things. *IEEE Internet of Things Journal*, 8(20), 15096–15106. DOI 10.1109/JIOT.2020.3031622.
30. Tong, B. D., Chen, L., Duan, H. B. (2021). A path planning method for UAVs based on multi-objective pigeon-inspired optimisation and differential evolution. *International Journal of Bio-Inspired Computation*, 17(2), 105–112. DOI 10.1504/IJBIC.2021.114079.
31. Rajakumari, K., Kumar, M. V., Verma, G., Balu, S., Sharma, D. K. et al. (2022). Fuzzy based ant colony optimization scheduling in cloud computing. *Computer Systems Science and Engineering*, 40(2), 581–592. DOI 10.32604/csse.2022.019175.
32. Xu, J. L., Zhang, Z. X., Hu, Z. M., Du, L., Cai, X. J. (2021). A many-objective optimized task allocation scheduling model in cloud computing. *Applied Intelligence*, 51(6), 3293–3310. DOI 10.1007/s10489-020-01887-x.

33. Xu, M., Zhang, M. Q., Cai, X. J., Zhang, G. Y. (2021). Adaptive neighbourhood size adjustment in MOEA/D-DRA. *International Journal of Bio-Inspired Computation*, 17(1), 14–23. DOI 10.1504/IJBIC.2021.113336.
34. Li, M., Wang, G. G. (2022). A review of green shop scheduling problem. *Information Sciences*, 589, 478–496. DOI 10.1016/j.ins.2021.12.122.
35. Shi, T., Ma, H., Chen, G., Hartmann, S. (2020). Location-aware and budget-constrained service deployment for composite applications in multi-cloud environment. *IEEE Transactions on Parallel and Distributed Systems*, 31(8), 1954–1969. DOI 10.1109/TPDS.71.
36. Pang, B. B., Hao, F., Yang, Y. X., Park, D. S. (2020). An efficient approach for multi-user multi-cloud service composition in human-land sustainable computational systems. *Journal of Supercomputing*, 76(7), 5442–5459. DOI 10.1007/s11227-019-03140-w.
37. Singh, S., Chana, I. (2016). A survey on resource scheduling in cloud computing: Issues and challenges. *Journal of Grid Computing*, 14(2), 217–264. DOI 10.1007/s10723-015-9359-2.
38. Liu, G. X., Shen, H. Y., Wang, H. Y. (2017). An economical and SLO-guaranteed cloud storage service across multiple cloud service providers. *IEEE Transactions on Parallel and Distributed Systems*, 28(9), 2440–2453. DOI 10.1109/TPDS.2017.2675422.
39. Douik, A., Dahrouj, H., Al-Naffouri, T. Y., Alouini, M. S. (2018). Distributed hybrid scheduling in multi-cloud networks using conflict graphs. *IEEE Transactions on Communications*, 66(1), 209–224. DOI 10.1109/TCOMM.2017.2749573.
40. Dhingra, S., Madda, R. B., Patan, R., Jiao, P., Barri, K. et al. (2021). Internet of things-based fog and cloud computing technology for smart traffic monitoring. *Internet of Things*, 14, 100175. DOI 10.1016/j.iot.2020.100175.
41. Zhang, Z. X., Zhao, M. K., Wang, H., Cui, Z. H., Zhang, W. S. (2022). An efficient interval many-objective evolutionary algorithm for cloud task scheduling problem under uncertainty. *Information Sciences*, 583, 56–72. DOI 10.1016/j.ins.2021.11.027.
42. Deb, S., Tammi, K., Gao, X. Z., Kalita, K., Mahanta, P. et al. (2021). A robust two-stage planning model for the charging station placement problem considering road traffic uncertainty. *IEEE Transactions on Intelligent Transportation Systems*, 23(7), 6571–6585. DOI 10.1109/TITS.2021.3058419.
43. Zhao, J., Mo, H. W. (2021). Magnetotactic bacteria optimisation algorithm with self-regulation interaction energy. *International Journal of Bio-Inspired Computation*, 18(3), 189–198. DOI 10.1504/IJBIC.2021.119195.
44. Yi, J. H., Xing, L. N., Wang, G. G., Dong, J., Vasilakos, A. V. et al. (2020). Behavior of crossover operators in NSGA-III for large-scale optimization problems. *Information Sciences*, 509, 470–487. DOI 10.1016/j.ins.2018.10.005.
45. Pradeep, K., Jacob, T. P. (2018). A hybrid approach for task scheduling using the cuckoo and harmony search in cloud computing environment. *Wireless Personal Communications*, 101(4), 2287–2311. DOI 10.1007/s11277-018-5816-0.
46. Nedjah, N., Mourelle, L. D. M., Morais, R. G. (2021). Inspiration-wise swarm intelligence meta-heuristics for continuous optimisation: A survey-part III. *International Journal of Bio-Inspired Computation*, 17(4), 199–214. DOI 10.1504/IJBIC.2021.116578.
47. Cai, X. J., Geng, S. J., Wu, D., Chen, J. J. (2021). Unified integration of many-objective optimization algorithm based on temporary offspring for software defects prediction. *Swarm and Evolutionary Computation*, 63, 100871. DOI 10.1016/j.swevo.2021.100871.
48. Zou, J., Sun, R. Q., Yang, S. X., Zheng, J. H. (2021). A dual-population algorithm based on alternative evolution and degeneration for solving constrained multi-objective optimization problems. *Information Sciences*, 579, 89–102. DOI 10.1016/j.ins.2021.07.078.



49. Cui, Z. H., Wen, J., Lan, Y., Zhang, Z. X., Cai, J. H. (2022). Communication-efficient federated recommendation model based on many-objective evolutionary algorithm. *Expert Systems with Applications*, 201, 116963. DOI 10.1016/j.eswa.2022.116963.
50. Balaji, S., Vikram, S. T., Kanagasabapathy, G. (2021). Jumping particle swarm optimisation method for solving minimum weight vertex cover problem. *International Journal of Bio-Inspired Computation*, 18(3), 143–152. DOI 10.1504/IJBIC.2021.119198.
51. Cai, X. J., Cao, Y. H., Ren, Y. Q., Cui, Z. H., Zhang, W. S. (2021). Multi-objective evolutionary 3D face reconstruction based on improved encoder-decoder network. *Information Sciences*, 581, 233–248. DOI 10.1016/j.ins.2021.09.024.
52. Cai, X. J., Zhang, J. J., Ning, Z. H., Cui, Z. H., Chen, J. J. (2021). A many-objective multistage optimization-based fuzzy decision-making model for coal production prediction. *IEEE Transactions on Fuzzy Systems*, 29(12), 3665–3675. DOI 10.1109/TFUZZ.2021.3089230.
53. Penalzoza, M. G. M., Montes, E. M., Reyes, A. M., Aguirre, H. E. (2021). Distance-based immune generalised differential evolution algorithm for dynamic multi-objective optimisation. *International Journal of Bio-Inspired Computation*, 18(2), 69–81. DOI 10.1504/IJBIC.2021.118091.
54. Nasir, M., Das, S., Maity, D., Sengupta, S., Halder, U. et al. (2012). A dynamic neighborhood learning based particle swarm optimizer for global numerical optimization. *Information Sciences*, 209, 16–36. DOI 10.1016/j.ins.2012.04.028.
55. Rather, S. A., Bala, S. P. (2021). Application of constriction coefficient-based particle swarm optimisation and gravitational search algorithm for solving practical engineering design problems. *International Journal of Bio-Inspired Computation*, 17(4), 246–259. DOI 10.1504/IJBIC.2021.116617.
56. Wang, D., Liao, F. X. (2021). Analysis of first-come-first-served mechanisms in one-way car-sharing services. *Transportation Research Part B-Methodological*, 147, 22–41. DOI 10.1016/j.trb.2021.03.006.
57. Kamalam, G. K., Sentamilselvan, K. (2022). SLA-based group tasks Max-min (GTMax-min) algorithm for task scheduling in multi-cloud environments. *EAI/Springer Innovations in Communication and Computing*, pp. 105–127. DOI 10.1007/978-3-030-74402-1\_6.
58. Ezzatti, P., Pedemonte, M., Martin, A. (2013). An efficient implementation of the Min-min heuristic. *Computers & Operations Research*, 40(11), 2670–2676. DOI 10.1016/j.cor.2013.05.014.
59. Walter, R. (2013). Comparing the minimum completion times of two longest-first scheduling-heuristics. *Central European Journal of Operations Research*, 21(1), 125–139. DOI 10.1007/s10100-011-0217-4.
60. Cui, Z. H., Zhang, M. Q., Wang, H., Cai, X. J., Zhang, W. S. et al. (2020). Hybrid many-objective cuckoo search algorithm with lévy and exponential distributions. *Memetic Computing*, 12(3), 251–265. DOI 10.1007/s12293-020-00308-3.
61. Li, H. F., Wang, D. J., Zhou, M. C., Fan, Y. S., Xia, Y. Q. (2022). Multi-swarm co-evolution based hybrid intelligent optimization for bi-objective multi-workflow scheduling in the cloud. *IEEE Transactions on Parallel and Distributed System*, 33(9), 2183–2197. DOI 10.1109/TPDS.2021.3122428.
62. Mallick, S., Kar, R., Mandal, D. (2022). Optimal design of second generation current conveyor using craziness-based particle swarm optimisation. *International Journal of Bio-Inspired Computation*, 19(2), 87–96. DOI 10.1504/IJBIC.2022.121234.
63. Cui, Z. H., Zhang, J. J., Wu, D., Cai, X. J., Wang, H. et al. (2020). Hybrid many-objective particle swarm optimization algorithm for green coal production problem. *Information Sciences*, 518, 256–271. DOI 10.1016/j.ins.2020.01.018.
64. Jena, R. K. (2015). Multi objective task scheduling in cloud environment using nested PSO framework. *Procedia Computer Science*, 57, 1219–1227. DOI 10.1016/j.procs.2015.07.419.
65. Srichandan, S., Kumar, T. A., Bibhudatta, S. (2018). Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm. *Future Computing and Informatics Journal*, 3(2), 210–230. DOI 10.1016/j.fcij.2018.03.004.

66. Cui, Z. H., Zhang, J. J., Wang, Y. C., Cao, Y., Cai, X. J. et al. (2019). A pigeon-inspired optimization algorithm for many-objective optimization problems. *Science China Information Sciences*, 62(7), 070212. DOI 10.1007/s11432-018-9729-5.
67. Huo, M. Z., Deng, Y. M., Duan, H. B. (2021). Cauchy-Gaussian pigeon-inspired optimisation for electromagnetic inverse problem. *International Journal of Bio-Inspired Computation*, 17(3), 182–188. DOI 10.1504/IJBIC.2021.114875.
68. Cai, X. J., Zhang, J. J., Liang, H., Wang, L., Wu, Q. D. (2019). An ensemble Bat algorithm for large-scale optimization. *International Journal of Machine Learning and Cybernetics*, 10(11), 3099–3113. DOI 10.1007/s13042-019-01002-8.
69. Chen, W. P., Ye, J., Wu, R. X., Liu, G. M., Kang, P. (2020). Firefly algorithm based on intelligent single particle learning. *International Journal of Computing Science and Mathematics*, 12(4), 309–326. DOI 10.1504/IJCSM.2020.112674.
70. Xiao, S. Y., Wang, W. J., Wang, H., Huang, Z. K. (2022). A new multi-objective artificial bee colony algorithm based on reference point and opposition. *International Journal of Bio-Inspired Computation*, 19(1), 18–28. DOI 10.1504/IJBIC.2022.120732.
71. Lin, Q. Z., Liu, S. B., Zhu, Q. L., Tang, C. Y., Song, R. Z. et al. (2018). Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems. *IEEE Transactions on Evolutionary Computation*, 22(1), 32–46. DOI 10.1109/TEVC.2016.2631279.
72. Mapetu, J. P. B., Chen, Z., Kong, L. (2019). Low-time complexity and low-cost binary particle swarm optimization algorithm for task scheduling and load balancing in cloud computing. *Applied Intelligence*, 49(9), 3308–3330. DOI 10.1007/s10489-019-01448-x.
73. Zeng, T., Wang, W. J., Wang, H., Cui, Z. H., Wang, F. et al. (2021). Artificial bee colony based on adaptive search strategy and random grouping mechanism. *Expert Systems with Applications*, 192, 116332. DOI 10.1016/j.eswa.2021.116332.
74. Zedadra, O., Guerrieri, A., Jouandeu, N., Spezzano, G., Seridi, H. et al. (2018). Swarm intelligence-based algorithms within IoT-based systems: A review. *Journal of Parallel and Distributed Computing*, 112, 173–187. DOI 10.1016/j.jpdc.2018.08.007.
75. Bai, T. T., Wang, D. B., Zain, A., Masroor, S. (2022). Formation control of multiple UAVs via pigeon inspired optimisation. *International Journal of Bio-Inspired Computation*, 19(3), 135–146. DOI 10.1504/IJBIC.2022.123106.
76. Gao, X. Z., Nalluri, M. S. R., Kannan, K., Sinharoy, D. (2020). Multi-objective optimization of feature selection using hybrid cat swarm optimization. *Science China Technological Sciences*, 64(3), 508–520. DOI 10.1007/s11431-019-1607-7.
77. Verma, J., Sobhanayak, S., Sharma, S., Turuk, A. K., Sahoo, B. (2017). Bacteria foraging based task scheduling algorithm in cloud computing environment. *2017 International Conference on Computing, Communication and Automation (ICCCA)*, pp. 777–782. Greater Noida.
78. Cui, Z. H., Zhao, P., Hu, Z. M., Cai, X. J., Zhang, W. S. et al. (2021). An improved matrix factorization based model for many-objective optimization recommendation. *Information Sciences*, 579, 1–14. DOI 10.1016/j.ins.2021.07.077.
79. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197. DOI 10.1109/4235.996017.
80. Cui, Z. H., Chang, Y., Zhang, J. J., Cai, X. J., Zhang, W. S. (2019). Improved NSGA-III with selection-and-elimination operator. *Swarm and Evolutionary Computation*, 49, 23–33. DOI 10.1016/j.swevo.2019.05.011.
81. Hou, Y., Wu, N. Q., Li, Z. W., Zhang, Y. X., Qu, T. et al. (2020). Many-objective optimization for scheduling of crude oil operations based on NSGA-III with consideration of energy efficiency. *Swarm and Evolutionary Computation*, 57, 100714. DOI 10.1016/j.swevo.2020.100714.

82. Cui, Z. H., Zhao, Y. R., Cao, Y., Cai, X. J., Zhang, W. S. et al. (2021). Malicious code detection under 5G HetNets based on multi-objective RBM model. *IEEE Network*, 35(2), 82–87. DOI 10.1109/MNET.011.2000331.
83. Jiang, C. B., Li, R. L., Chen, J., Li, S. F., Chen, T. G. et al. (2021). Modelling the green supply chain of hotels based on front-back stage decoupling: Perspective of ant colony labour division. *International Journal of Bio-Inspired Computation*, 18(3), 176–188. DOI 10.1504/IJBIC.2021.119201.
84. Zhang, W. Q., Hou, W. L., Li, C., Yang, W. D., Gen, M. (2021). Multidirection update-based multiobjective particle swarm optimization for mixed no-idle flow-shop scheduling problem. *Complex System Modeling and Simulation*, 1(3), 176–197. DOI 10.23919/CSMS.2021.0017.
85. Cai, X. J., Hu, Z. M., Zhao, P., Zhang, W. S., Chen, J. J. (2020). A hybrid recommendation system with many-objective evolutionary algorithm. *Expert Systems with Applications*, 159, 113648. DOI 10.1016/j.eswa.2020.113648.
86. Cao, Y., Zhou, L., Xue, F. (2021). An improved NSGA-II with dimension perturbation and density estimation for multi-objective DV-hop localisation algorithm. *International Journal of Bio-Inspired Computation*, 17(2), 121–130. DOI 10.1504/IJBIC.2021.114081.
87. Sobhanayak, S., Mendes, S., Jaiswal, K. (2022). Bi-objective task scheduling in cloud data center using whale optimization algorithm. *Lecture Notes on Data Engineering and Communications Technologies*, 106, 347–360. DOI 10.1007/978-981-16-8403-6.
88. Cui, Z. H., Zhao, L. H., Zeng, Y. Q., Ren, Y. Q., Zhang, W. S. et al. (2021). A novel PIO algorithm with multiple selection strategies for many-objective optimization problems. *Complex System Modeling and Simulation*, 1(4), 291–307. DOI 10.23919/CSMS.2021.0023.
89. Shen, Y. K., Deng, Y. M. (2021). Pigeon-inspired optimisation algorithm with hierarchical topology and receding horizon control for multi-UAV formation. *International Journal of Bio-Inspired Computation*, 18(4), 239–249. DOI 10.1504/IJBIC.2021.119949.
90. He, H., Xu, G. Q., Pang, S. C., Zhao, Z. H. (2016). AMTS: Adaptive multi-objective task scheduling strategy in cloud computing. *China Communications*, 13(4), 162–171. DOI 10.1109/CC.2016.7464133.
91. Chen, X., Long, D. (2019). Task scheduling of cloud computing using integrated particle swarm algorithm and ant colony algorithm. *Cluster Computing*, 22(2), 2761–2769. DOI 10.1007/s10586-017-1479-y.
92. Kim, J. S., Ahn, C. W. (2021). Expediting population diversification in evolutionary computation with quantum algorithm. *International Journal of Bio-Inspired Computation*, 17(1), 63–73. DOI 10.1504/IJBIC.2021.113356.
93. Sanaj, M. S., Prathap, P. M. J. (2019). Nature inspired chaotic squirrel search algorithm (CSSA) for multi objective task scheduling in an IAAS cloud computing atmosphere. *Engineering Science and Technology, an International Journal*, 23(4), 891–902. DOI 10.1016/j.jestch.2019.11.002.
94. Frincu, M. E., Craciun, C. (2011). Multi-objective meta-heuristics for scheduling applications with high availability requirements and cost constraints in multi-cloud environments. *4th IEEE/ACM International Conference on Cloud and Utility Computing*, pp. 267–274. Melbourne, Australia.
95. Chen, W., Cao, J. W., Wan, Y. X. (2013). QoS-Aware virtual machine scheduling for video streaming services in multi-cloud. *Tsinghua Science and Technology*, 18(3), 308–317. DOI 10.1109/TST.2013.6522589.
96. Geethanjali, M., Sujana, J. A. J., Revathi, T. (2014). Ensuring truthfulness for scheduling multi-objective real time tasks in multi cloud environments. *2014 4th International Conference on Recent Trends in Information Technology*, Chennai, India. DOI 10.1109/ICRTIT.2014.6996183.
97. Kang, S., Veeravalli, B., Aung, K. M. M. (2014). Scheduling multiple divisible loads in a multi-cloud system. *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*, pp. 371–378. London, UK.

98. Rizvi, N., Pranav, P., Rajv, B., Paul, S. (2016). Auction mechanism using sjf scheduling for sla based resource provisioning in a multi-cloud environment. *Communication and Computing Systems-Proceedings of the International Conference on Communication and Computing Systems*, Gurgaon, India. DOI 10.1201/9781315364094-38.
99. Miraftabzadeh, S. A., Rad, P., Jamshidi, M. (2016). Efficient distributed algorithm for scheduling workload-aware jobs on multi-clouds. *2016 11th Systems of Systems Engineering Conference*, Kongsberg, Norway. DOI 10.1109/SYSESE.2016.7542955.
100. Hao, Y. S., Xia, M. D., Wen, N., Hou, R. T., Deng, H. et al. (2017). Parallel task scheduling under multi-clouds. *KSII Transactions on Internet and Information Systems*, 11(1), 39–60.
101. Wang, C., Lu, Z., Wu, Z., Wu, J., Huang, S. (2017). Optimizing multi-cloud CDN deployment and scheduling strategies using Big data analysis. *2017 IEEE 14th International Conference on Services Computing*, pp. 273–280. Honolulu, HI, United States.
102. Kang, S., Veeravalli, B., Aung, K. M. M. (2018). Dynamic scheduling strategy with efficient node availability prediction for handling divisible loads in multi-cloud systems. *Journal of Parallel and Distributed Computing*, 113, 1–16. DOI 10.1016/j.jpdc.2017.10.006.
103. Panda, S. K., Pande, S. K., Das, S. (2018). Task partitioning scheduling algorithms for heterogeneous multi-cloud environment. *Arabian Journal for Science and Engineering*, 43(2), 913–933. DOI 10.1007/s13369-017-2798-2.
104. Jena, T., Mohanty, J. R. (2018). GA-based customer-conscious resource allocation and task scheduling in multi-cloud computing. *Arabian Journal for Science and Engineering*, 43(8), 4115–4130. DOI 10.1007/s13369-017-2766-x.
105. Roy, A., Midya, S., Hazra, D., Majumder, K., Phadikar, S. (2018). A hybrid task scheduling algorithm for efficient task management in multi-cloud environment. *Advances in Intelligent Systems and Computing*, 811, 47–57. DOI 10.1007/978-981-13-1544-2.
106. Hubert Shanthan, B. J., Arockiam, L. (2018). Rate aware meta task scheduling algorithm for multi cloud computing (RAMTSA). *2nd National Conference on Computational Intelligence*, pp. 012001. Bangalore, India.
107. Thirumalaiselvan, C., Venkatachalam, V. (2019). A strategic performance of virtual task scheduling in multi cloud environment. *Cluster Computing-The Journal of Networks Software Tools and Applications*, 22, 9589–9597. DOI 10.1007/s10586-017-1268-7.
108. Zhang, Y. T., Di, B. Y., Zheng, Z. J., Lin, J. L., Song, L. Y. (2019). Joint data offloading and resource allocation for multi-cloud heterogeneous mobile edge computing using multi-agent reinforcement learning. *2019 IEEE Global Communications Conference*, pp. 1–6. Piscataway, NJ, USA.
109. Panda, S. K., Gupta, I., Jana, P. K. (2019). Task scheduling algorithms for multi-cloud systems: Allocation-aware approach. *Information Systems Frontiers*, 21(2), 241–259. DOI 10.1007/s10796-017-9742-6.
110. Li, C. L., Zhang, J., Tang, H. L. (2019). Replica-aware task scheduling and load balanced cache placement for delay reduction in multi-cloud environment. *Journal of Supercomputing*, 75(5), 2805–2836. DOI 10.1007/s11227-018-2695-9.
111. Shanthan, H. B. J., Arockiam, L., Donald, A. C., Vinoth Kumar, A. D., Stephen, R. (2019). Priority intensed meta task scheduling algorithm for multi cloud environment (PIMTSA). *3rd National Conference on Computational Intelligence*, Bangalore, Karnataka, India. DOI 10.1088/1742-6596/1427/1/012007.
112. Jing, F., Ferriter, K., Castillo, C. (2020). A cloud-agnostic framework to enable cost-aware scheduling of applications in a multi-cloud environment. *2020 IEEE/IFIP Network Operations and Management Symposium*, Budapest, Hungary. DOI 10.1109/NOMS47738.2020.9110325.
113. Pasdar, A., Hassanzadeh, T., Lee, Y. C., Mans, B. (2020). ANN-assisted multi-cloud scheduling recommender. *Communications in Computer and Information Science*, 1332, 737–745. DOI 10.1007/978-3-030-63820-7.

114. Karaja, M., Ennigrou, M., Said, L. B. (2020). Budget-constrained dynamic Bag-of-tasks scheduling algorithm for heterogeneous multi-cloud environment. *2020 International Multi-Conference on Organization of Knowledge and Advanced Technologies*, pp. 1–6. Tunis, Tunisia.
115. Karaja, M., Ennigrou, M., Said, L. B. (2020). Solving dynamic Bag-of-tasks scheduling problem in heterogeneous multi-cloud environment using hybrid bi-level optimization model. *20th International Conference on Hybrid Intelligent Systems*, pp. 171–180. Bhopal, India.
116. Mohanraj, T., Santhosh, R. (2021). Multi-swarm optimization model for multi-cloud scheduling for enhanced quality of services. *Soft Computing*. DOI 10.1007/s00500-021-06184-4.
117. Zhu, Q. H., Tang, H., Huang, J. J., Hou, Y. (2021). Task scheduling for multi-cloud computing subject to security and reliability constraints. *IEEE/CAA Journal of Automatica Sinica*, 8(4), 848–865. DOI 10.1109/JAS.6570654.
118. Selvapandian, D., Santhosh, R. (2021). Dynamic multilevel scheduling strategy (MSS) mechanism for commercial multi-cloud surroundings. *Distributed and Parallel Databases*. DOI 10.1007/s10619-021-07349-8.
119. Hubert Shanthan, B. J., Cecil Donald, A., Dalvin Vinoth Kumar, A., Stephen, R., Arockiam, L. (2021). TCAMTSA: Time and cost based scheduling algorithm for multi cloud systems. *2021 7th International Conference on Advanced Computing and Communication Systems*, pp. 1922–1926. Coimbatore, India.
120. Cai, X. J., Geng, S. J., Wu, D., Cai, J. H., Chen, J. J. (2021). A multicloud-model-based many-objective intelligent algorithm for efficient task scheduling in Internet of Things. *IEEE Internet of Things Journal*, 8(2), 9645–9653. DOI 10.1109/JIOT.2020.3040019.
121. Su, Y., Fan, W. H., Liu, Y. A., Wu, F. (2021). Game-based distributed pricing and task offloading in multi-cloud and multi-edge environments. *Computer Networks*, 200, 108523. DOI 10.1016/j.comnet.2021.108523.
122. Panda, S. K., Jana, P. K. (2015). Efficient task scheduling algorithms for heterogeneous multi-cloud environment. *Journal of Supercomputing*, 71(4), 1505–1533. DOI 10.1007/s11227-014-1376-6.
123. Sooezi, N., Abrishami, S., Lotfian, M. (2015). Scheduling data-driven workflows in multi-cloud environment. *IEEE 7th International Conference on Cloud Computing Technology and Science*, pp. 163–167. Vancouver, BC, Canada.
124. Nandhakumar, C., Ranjithprabhu, K. (2015). Heuristic and meta-heuristic workflow scheduling algorithms in multi-cloud environments—A survey. *2015-Proceedings of the 2nd International Conference on Advanced Computing and Communication Systems*, Coimbatore, India. DOI 10.1109/ICACCS.2015.7324053.
125. Lin, B., Guo, W., Chen, G., Xiong, N., Li, R. (2015). Cost-driven scheduling for deadline-constrained workflow on multi-clouds. *2015 IEEE 29th International Parallel and Distributed Processing Symposium Workshops*, pp. 1191–1198. Hyderabad, India.
126. Maheshwari, K., Jung, E. S., Meng, J., Morozov, V., Vishwanath, V. et al. (2016). Workflow performance improvement using modelbased scheduling over multiple clusters and clouds. *Future Generation Computer Systems*, 54, 206–218. DOI 10.1016/j.future.2015.03.017.
127. Gupta, I., Kumar, M. S., Jana, P. K. (2017). Transfer time-aware workflow scheduling for multi-cloud environment. *2016 IEEE International Conference on Computing, Communication and Automation*, pp. 732–737. Greater Noida, India.
128. Suri, P. K., Rani, S. (2017). Design of task scheduling model for cloud applications in multi cloud environment. *Communications in Computer and Information Science*, 750, 11–24. DOI 10.1007/978-981-10-6544-6.
129. Panda, S. K., Jana, P. K. (2018). Normalization-based task scheduling algorithms for heterogeneous multi-cloud environment. *Information Systems Frontiers*, 20(2), 373–399. DOI 10.1007/s10796-016-9683-5.
130. Hu, H. Y., Li, Z. J., Hu, H., Chen, J., Ge, J. D. et al. (2018). Multi-objective scheduling for scientific workflow in multicloud environment. *Journal of Network and Computer Applications*, 114, 108–122. DOI 10.1016/j.jnca.2018.03.028.

131. Li, R., Wu, C. Q., Hou, A., Wang, Y. Q., Gao, T. Y. et al. (2018). On scheduling of high-throughput scientific workflows under budget constraints in multi-cloud environments. *16th IEEE International Symposium on Parallel and Distributed Processing with Applications*, pp. 1087–1094. Melbourne, VIC, Australia.
132. Mohammadi, S., PourKarimi, L., Pedram, H. (2019). Integer linear programming-based multi-objective scheduling for scientific workflows in multi-cloud environments. *Journal of Supercomputing*, 75(10), 6683–6709. DOI 10.1007/s11227-019-02877-8.
133. Chen, Z. Y., Lin, K., Lin, B., Chen, X., Zheng, X. H. et al. (2020). Adaptive resource allocation and consolidation for scientific workflow scheduling in multi-cloud environments. *IEEE ACCESS*, 8, 190173–190183. DOI 10.1109/Access.6287639.
134. Farid, M., Latip, R., Hussin, M., Hamid, N. A. W. A. (2020). Scheduling scientific workflow using multi-objective algorithm with fuzzy resource utilization in multi-cloud environment. *IEEE Access*, 8, 24309–24322. DOI 10.1109/Access.6287639.
135. Tang, X. Y. (2021). Reliability-aware cost-efficient scientific workflows scheduling strategy on multi-cloud systems. *IEEE Transactions on Cloud Computing*. DOI 10.1109/TCC.2021.3057422.
136. Ulabedin, Z., Nazir, B. (2021). Replication and data management-based workflow scheduling algorithm for multi-cloud data centre platform. *The Journal of Supercomputing*, 77(10), 10743–10772. DOI 10.1007/s11227-020-03541-2.
137. Wang, P. W., Lei, Y. H., Zhao, Y. Y., Zhang, Z. H. (2021). Clustering coefficient-based workflow slicing and multi-cloud scheduling. *Journal of Tongji University*, 49(8), 1192–1201.
138. Mohammadzadeh, A., Masdari, M. (2021). Scientific workflow scheduling in multi-cloud computing using a hybrid multi-objective optimization algorithm. *Journal of Ambient Intelligence and Humanized Computing*, 1–21. DOI 10.1007/s12652-021-03482-5.
139. Sujana, J. A. J., Raj, R. V., Revathi, T. (2022). Fuzzy-based workflow scheduling in multi-cloud environment. *EAI/Springer Innovations in Communication and Computing*, pp. 201–215. DOI 10.1007/978-3-030-74402-1\_11.
140. Chakravarthi, K. K., Neelakantan, P., Shyamala, L., Vaidehi, V. (2022). Reliable budget aware workflow scheduling strategy on multi-cloud environment. *Cluster Computing*, 25(2), 1189–1205. DOI 10.1007/s10586-021-03464-4.