Journal of King Saud University – Computer and Information Sciences 34 (2022) 2309–2331

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

A review on job scheduling technique in cloud computing and priority rule based intelligent framework

Saydul Akbar Murad [a,1], Abu Jafar Md Muzahid [a], Zafril Rizal M Azmi [a], Md Imdadul Hoque [b], Md Kowsher [c]

[a] Faculty of Computing, College of Computing & Applied Sciences, Universiti Malaysia Pahang, 26600 Pekan, Pahang, Malaysia
[b] University of Bremen, Bremen 28359, Germany
[c] Stevens Institute of Technology, NJ 07030, United States

ARTICLE INFO

Article history:
Received 29 January 2022
Revised 5 March 2022
Accepted 28 March 2022
Available online 13 April 2022

Keywords:
Cloud computing
Job scheduling
Taxonomy
Conceptual framework
Resource allocation
Open research issue

ABSTRACT

In recent years, the concept of cloud computing has been gaining traction to provide dynamically increasing access to shared computing resources (software and hardware) via the internet. It's not secret that cloud computing's ability to supply mission-critical services has made job scheduling a hot subject in the industry right now. Cloud resources may be wasted, or in-service performance may suffer because of under-utilization or over-utilization, respectively, due to poor scheduling. Various strategies from the literature are examined in this research in order to give procedures for the planning and performance of Job Scheduling techniques (JST) in cloud computing. To begin, we look at and tabulate the existing JST that is linked to cloud and grid computing. The present successes are then thoroughly reviewed, difficulties and flows are recognized, and intelligent solutions are devised to take advantage of the proposed taxonomy. To bridge the gaps between present investigations, this paper also seeks to provide readers with a conceptual framework, where we proposed an effective job scheduling technique in cloud computing. These findings are intended to provide academics and policymakers with information about the advantages of a more efficient cloud computing setup. In cloud computing, fair job scheduling is most important. We proposed a priority-based scheduling technique to ensure fair job scheduling. Finally, the open research questions raised in this article will create a path for the implementation of an effective job scheduling strategy.

Contents

E-mail address: zafril@ump.edu.my (Z.R.M Azmi)
[1] ORCID: 0000-0002-9015-1448.

☆ The authors would like to thank the Ministry of Higher Education for providing financial support under Fundamental Research Grant Scheme (FRGS) No. FRGS/1/2019/ICT03/UMP/02/2 (University reference RDU1901194).
Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2022.03.027
1319-1578/

1. Introduction

Now, Cloud Computing has grown in popularity as a medium for scientific applications. To facilitate scientific study, cloud computing aims to share large-scale resources and equipment in the areas of processing, storage, information, and expertise with other researchers. Cloud computing's job scheduling algorithms are among the most difficult theoretical problems to solve Ghanbari and Othman (2012). Cloud computing uses a scheduler (broker) to figure out how to best allocate a limited number of resources to incoming activities and applications in order to achieve a variety of scheduling goals (e.g., monetary cost, computational cost, makespan, availability, reliability, response time, resource utilization, energy consumption, etc.) Lee (1996); Allahverdi et al. (2008). One of the most notable uses of contemporary scheduling has been the allocation of distributed computing systems of limited resources to jobs submitted by Internet users since their establishment in 1980. In the last few years, a new technology called "cluster systems" has emerged, which combines several separate computers into a single unit. Grid systems were developed in response to the weakness of cluster systems, which only utilize local resources, by gathering together all heterogeneous resources accessible in geographically distant places Weinhardt et al. (2009). Cloud computing is a relatively new technology which makes use of the advantages of both clustered and grid-based systems.

Due to the huge solution space, many scheduling issues that are NP-hard or NP-completely take a long time to implement an optimum or sub-optimal solution in the shortest time. Due to the limited resources in modern computer systems, there is no polynomial time-scheduling technique which could be used to improve the constrained resources scheduling. Using a simple example from Taillard (1990), we can see that just about 0.02 percent of the possible solutions use between 1 and 1.01 times the time required to find the ideal answer. Finding the best answer to a complex problem is quite challenging, as this example illustrates. As a result, most scholars have been motivated to look for a quick but effective solution to these kinds of scheduling challenges. The two most basic forms of scheduling methods are static and dynamic scheduling strategies. However, because cloud settings are inherently dynamic, additional dynamic algorithms must be incorporated into the cloud scheduling process to achieve outstanding results in this field. Static algorithms, on the other hand, are only utilized when the workloads vary just slightly. As a result, adopting deterministic ways to tackle the job scheduling problem is unfeasible in this circumstance Allahverdi (2015). Nondeterministic meta-heuristic algorithms have been offered as a way to considerably address this challenge in a polynomial amount of time.

Consumers and producers of cloud services can benefit from a variety of advantages because to dynamic work scheduling approaches and virtualization technology. Resource (task) scheduling that is effective not only minimize resource consumption (increasing the resource used), but also assures that new jobs are completed as promptly as possible (the minimizes of makespan). Job scheduling has become most important due to the possibility of a scarcity of cloud resources as a result of the continual increase in workloads at cloud datacenters. This has resulted in a significant increase in the importance of task scheduling. As a result, more study into the still-developing topic of cloud job scheduling is required to push for things like more effective mapping of incoming job to available resources and improved criteria for measuring how efficiently a service is provided. Scheduling algorithms can be used to optimize a variety of quality of service (QoS) parameters, for example resource use and utilization, task rejection ratio, energy consumption, and other constraints, by determining the optimal set of resources available to carry out incoming tasks (underutilization and over utilization). The primary objective of a

scheduling approach is to find the most efficient use of the available resources (SLA).

1.1. Motivation for conducting the research

As the purpose of cloud computing is to maximize the usage of virtual machines (VMs) while minimizing data center operational expenses, resource scheduling is crucial. This leads to an increase in the quality of service (QoS) metrics in cloud computing. To accomplish the aims of both cloud service providers and users, resource scheduling manages a huge number of user insistence and distributes them all to the most applicable virtual machines. We searched the literature for scheduling algorithms and determined that just a few well-known surveys have been revealed in the cloud computing Raghava and Singh (2014); Milani and Navimipour (2016); Thakur and Goraya (2017); Ghomi et al. (2017). These questionnaires show how this suggested algorithm works in its most basic form. Example, Randles et al. (2010) assess several load balancing solutions based on one crucial performance indicator, throughput, while the methodology ignores other factors like flowchart, taxonomy, and other survey characteristics. Despite the fact that Raghava and Singh (2014) presented a brief overview of the existing scheduling techniques based on QoS criteria, all of the surveys described above focus on only a few elements (QoS parameters, year-wise analysis, state-of-art).

Further, none of the existing surveys is complete, and none of the existing surveys considers all of the QoS characteristics at the same time Ghomi et al. (2017) and Thakur and Goraya (2017) improve the survey methodology and considers more QoS metrics, besides taxonomy, a visual representation, and flow chart. In Kumar et al. (2019), the authors conducted an excellent review on cloud computing, in which they included only the work scheduling algorithms and did not mention any obstacles that they encountered during their research, which was a mistake.Houssein et al. (2021) did another survey in which they present a taxonomy as well as a thorough discussion of the study subject. They did not provide any framework for successful job scheduling in this section, which we added in our research. Research in the field of job scheduling, despite this, its development is still at an early phase. As a result, we seek a comprehensive survey that will assist us in expanding and integrating study findings into resource scheduling on a continuous basis. This work represents a complete and systematic analysis of job scheduling strategies, as well as an assessment of present and future research challenges originating from the employment of cutting-edge scheduling approaches. We have done our best to apply our experience in this work.

1.2. Contribution of this paper

The purpose of this research is to explore and critique existing cloud scheduling methodologies, as well as the performance matrices used in the job scheduling process. The results of this survey will be beneficial in developing new job scheduling algorithms or strategies in the future. The following are concrete examples of the contents of contributions to this paper:

1. In this paper, we investigate and assess various well known existing heuristic, meta-heuristics, hybrid, and training-based job scheduling algorithms in the cloud and all resource scheduling techniques what is used during job scheduling.
2. A complete study is carried out using the current research flow and expert opinions to identify, segment, and classify the work scheduling strategy associated to cloud computing. In this area, we've created a unique taxonomy for cloud-based task scheduling solutions.

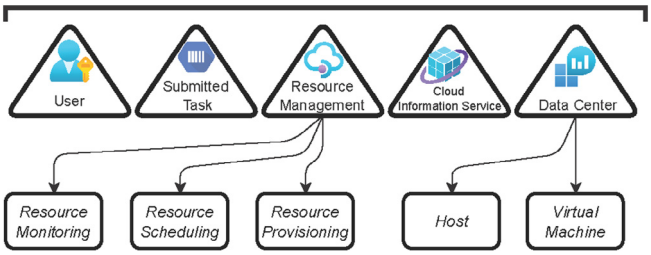

Fig. 1. Workflow of JST in cloud computing.

3. This survey provides a Job Scheduling Framework, which is a route for further research and development in cloud computing. The suggested framework is structured into four sections, each of which is directly related to effective job scheduling.
4. According to the findings of this study, a novel job scheduling algorithm based on priorities has been proposed, and it will be used to ensure that jobs are distributed fairly throughout the CPU. Furthermore, it will improve the performance of scheduling matrices such as the flow time, makespan time, and so on.
5. This paper provides a thorough understanding of the resource allocation system, as well as the advantages and drawbacks of all of the strategies that have been implemented.
6. Explicitly defining the advantages and disadvantages of meta-heuristic, heuristic, hybrid, and training-based job scheduling algorithms. Additionally, various cloud-based simulation tools are shown and contrasted.
7. Open research questions are identified and addressed in order to develop a way for future study on existing works and probable future research areas.

The paper allows readers to discover these subjects because it provides a complete overview of specific topics crucial to the creation of the conceptual framework. Some areas of the page are frequently embellished. Section III goes into detail about the difficulties and issues associated with cloud-based job scheduling, while Section II provides a high-level summary of job scheduling strategy Section IV represents a cloud-based Job Scheduling Taxonomy, and Section V represents a comprehensive conceptual framework for Job Scheduling Technique. In VI, priceless future research signals are synchronizing to provide future challenges to acknowledged researchers. Finally, Section VII brings the study to a close by exposing the contribution of the article.

2. Overview of JST in cloud computing

Typically, each consumer utilizes a cloud platform on a computer or smartphone to request a certain service via a browser Internet connection. As a result, innovative job scheduling algorithms can effectively balance workload across system hosts. The scheduling of jobs is an important aspect of resource management (RR) and job scheduling is a method of determining how jobs is conducted in the system, including the authentic mapping of resource components and the time at which they should be completed Mansouri and Javidi (2020). Fig. 1 shows a job overflow demonstrating job scheduling in cloud computing. In a Cloud computing context, work scheduling essentially entails mapping jobs to available ingredient resources. Prior to scheduling, we must first provision resources. The term "provisioning" refers to the formally assigned resources for the execution of any application. Scheduling aids in the optimization of resources based on the needs of the user. Currently, numerous researchers from various nations are attempting to enhance job scheduling. By inventing new algo-

rithms based on priority criteria, the author of Lee (1996) focuses on improving job scheduling.

2.1. User

Several well-known companies around the world have already made the transition to the cloud. The reason for this is that in the current world, everything is now going to be in online, and individuals will want to store all of their data in the cloud. Any form of organization, a collection of people, or a single person can be a user. A large number of companies exist to meet the needs of users. The user submits their request to the cloud service provider, who then allocates resources based on the request.

2.2. Submitted task

Job scheduling is a technique for allocating certain jobs to specific resources at specific times. The job-scheduling problem is a major and difficult topic in cloud computing. Job scheduling in cloud computing is primarily concerned to improve the efficient utilization of resources such as bandwidth, memory, and completion time. An effective work scheduling approach should aim for a lower response time so that submitted jobs are completed in the shortest amount of time possible and there is no in-time where resources are transferred Patel and Bhoi (2013). All tasks submitted by the user are saved in the state of submitted task. Those are unprocessed tasks. The final task is chosen later based on resource availability and authentication of this task.

2.3. Resource management

The purpose of RR in cloud computing is to ensure high resource availability, time variant service model fulfilment, resource sharing, and resource usage efficiency and reliability. The term RR in the field of cloud computing refers to a process that provides cloud customers with QoS assurances while also efficiently managing the above-mentioned resources Kumar (2018); Parikh et al. (2017).

According to our definition, research management entails the provisioning, scheduling, and monitoring of resources, as shown in Fig. 1. Physical resources like as disc space, CPU cores, and network bandwidth are managed by these processes. This resource has to be divided and shared into all virtual machines (VM) that may run a variety of tasks Mohamaddiah et al. (2014). Now we'll go over every aspect of the resource management system.

Resource Provisioning: Visualized resources can be allocated to users using the resource provisioning approach. A cloud service provider generates and distributes virtual machines (VMs) in response to a user's request. It is also the responsibility of resource provisioning to meet user demands based on QoS specifications, service level agreements (SLAs), and matching resources to incoming workloads.

As a general principle, Resource provisioning purpose is to identify and prioritize which resources are most appropriate for upcoming application requests (demands) to keep the number of resources required to service the application to a minimum (maximum throughput and low execution time). Resource provisioning links forthcoming requests to running virtual machines, guaranteeing that the customer obtains services in the shortest possible time and at the cheapest price, while the service provider makes the most money possible without sacrificing SLA compliance Javadi et al. (2012).

Resource Scheduling: Quality of service (QoS) variables are used to determine which activities would be prioritized for execution. For task execution, Scheduling uses heuristic or meta-heuristic approaches to pick the optimal virtual machines and ensure that QoS constraints are fulfilled. On-demand scheduling is a resource scheduling strategy in which a cloud service provider delivers resources to a random task fast. It is feasible to run multiple processes on a single VM, but this method has a problem with uneven workload allocation, leading to performance deterioration and the possibility of over-provisioning Singh and Chana (2016). It's possible to hold VM for a period, however this can lead to an under-provisioning problem. Over-provisioning and under provisioning increase service costs because of the excessive use of resources and time they cause. We need a resource provisioning algorithm that can assess and arrange impending workloads efficiently to deal with circumstances like these.

Identifying the better cloud resources for incoming end user applications (jobs) in order to maximize main performance metrics and the resource utilization ratio is the primary purpose of scheduling Singh and Chana (2016). Various performance matrices are available in cloud computing, for example execution cost, response time, makespan time and dependability. Resource scheduling (RS) in cloud computing has become a major difficulty because of the dynamism, heterogeneity, and dispersion of resources. These difficulties cannot be addressed by current scheduling solutions. We require a scheduling technique that can distributes diverse workloads among cloud resources (VMs) based on their capability to avoid overload and under-load.

Resource Monitoring: The key to achieving resource utilization with high-performance management is resource monitoring in cloud computing. Cloud resource monitoring is a method that allows cloud providers to regulate and maintain their software and infrastructure while also providing effective cost and output to their customers. After collecting the information from the Host and VM, Resource Monitor provides the task scheduler on the status of the tasks assigned to the various VMs and profiles each VM at a predetermined frequency Mehta et al. (2017).

2.4. Cloud information service (CIS)

In job scheduling, the cloud information service is crucial. Its role is to act as a liaison between the datacenter and the datacenter Broker. CIS is a type of cloud storage that contains the resources that are available in the cloud. When a datacenter is constructed, it must first be registered with CIS. The broker then attempts to get the resources that have been registered with CIS. Make a connection with the datacenter once the broker can read the data from CIS.

2.5. Datacenter

A datacenter (also known as a server farm) is a centralized storage, administration, and distribution facility for data and information. A datacenter is often a structure that houses computer systems and other components including telecommunications and storage Stryer (2010). Cloud datacenters are designed to meet very particular infrastructure requirements in order to serve cloud customers efficiently. Naturally, datacenters are a significant component of cloud computing, with a particular emphasis on reliable networks, content, and service, to name a few. A datacenter must have two required components: virtual machines (VMs) and hosts (as defined below).

VM Allocation: A virtual machine (VM) is a computer that operates in the same way as any other physical computer, such as a smartphone, laptop, or server. It features a RAM, CPU, and discs for storing your files, additional capability of connecting to the internet, if necessary, as well. Virtual machines (VMs) are often thought of as virtual computers or software-defined computers running on physical servers, with no physical components other than code, whereas the physical components of your computer (known as hardware) are present and palpable Yao et al. (2013).

Fig. 2. The obstacles and issues encountered during effective job scheduling are depicted in this diagram.

In cloud computing the datacenter detect and allocate the appropriate physical resources for each requested VM. The number of VMs, their configurations, and the connectivity requirements can all be defined in a user's request. A user may not be aware of the communication requirements between virtual machines (VMs) in advance.

Host: In a datacenter, hosts play an important role. The cloudlet data is stored on Host. The RAM is provided by the host during the job scheduling process. The bandwidth is provided by the host as well. The processing component is a component of the host. The MIPS is identified by the processing element (Million Instruction per second). The VM processes the element based on the MIPS. The service provider can provide a number of hosts based on consumer demand.

3. Challenges & issues of JST in cloud computing

In order to navigate effective job scheduling, rigorous evaluations on a variety of VM, datacenter, host, and datacenter brokers will be required. These issues point to the need to investigate the core causes of job scheduling system failures and identify the events that lead to possible failures. Obviously, policymakers and researchers rely on extensive reviews to determine the best methods. The evolution and use of job scheduling technology are likely to be hampered by a number of obstacles. The Fig. 2 describe some of the significant challenges that could stymie technology adoption before and after it reaches full maturity.

3.1. Workload fluctuations

The CPU, network resources, and storage in cloud data centers are generally virtualized. In comparison to traditional data centers, these virtualized resources use less energy. Users are provided with virtual machines (VMs), which are a sort of virtualized environment. These virtual machines are used to handle high-volume, high-variability workloads. As the demand for application rises, the loads may fluctuate dramatically and increasingly. Predictable and unexpected workloads are two different types of workloads.

3.2. Identical and diversified workload

In the cloud, there are two distinct sorts of workloads. One type of workload is homogenous, which is defined as having the same configurations as other workloads. CPUs, RAM, storage, and even execution time are all considered here. If cloud systems are constructed correctly, they can handle both categories of workloads Selvi et al. (2014).

3.3. Interactional workloads and batch workloads

Various performance indicators are available in cloud computing, for example execution cost, response time, dependability and makespan time. RS in cloud computing has become a major difficulty because of the heterogeneity, dispersion of resources, and dynamism. These difficulties cannot be addressed by current scheduling solutions. To avoid overload and underload, we need a scheduling approach that can distributes different workloads in total cloud resources (VMs) based on their capacity.

3.4. Reduction of cost and best resource utilization

The two most important considerations in cloud resource allocation are cost savings and maximum utilization. Customers should be entrusted with the obligation of continuing the service through a dependable cloud system. In order for this to work, the service provider must be able to supply low-cost services to its customers. There are many ways to accomplish this, including using effective methods for monitoring resource consumption and reducing user expenses.

3.5. Managing high availability for long-term jobs

The duration of cloud-based jobs might range from a few minutes to a few hours. This necessitates the availability of resources for work without interruption or failure. Consequently, any failure or unavailability must be dealt with in order to shift jobs to available resources. Users must be unaware of any downtime because the strategies must implement the procedure quickly.

3.6. Granularity of scheduling is increased compared to traditional scheduling

When it comes to cloud computing, the scheduling challenge has gone from simple task scheduling in traditional cloud systems with limited data transfers to heavy VM migrations VM resource scheduling Kaur et al. (2017).

3.7. VM migration

When dealing with insufficient cloud resources, VM migration is one options that can be used. Virtual machines (VMs) can be moved between hosts to make room for more resources. It's an element of maintaining hardware virtualization systems, and it's something that virtualization service providers consider.

3.8. Uncertainty

The current cloud scheduling solutions are based on deterministic modelling with previous knowledge of jobs and resources. However, in cloud computing, this is not possible because the tasks received for computation are very unexpected in nature, and the service provider is uninformed of the amount of data and computation that must be managed. Furthermore, virtualization technology isolates the cloud service provider (CSP) and service users from the specifics of the resources available, posing further hurdles to the performance of the service provider and service users. Because of the unpredictability around metrics such as the quantity of computing resources available, their speed and capabilities, bandwidth changes, and resource availability, service providers and Con-

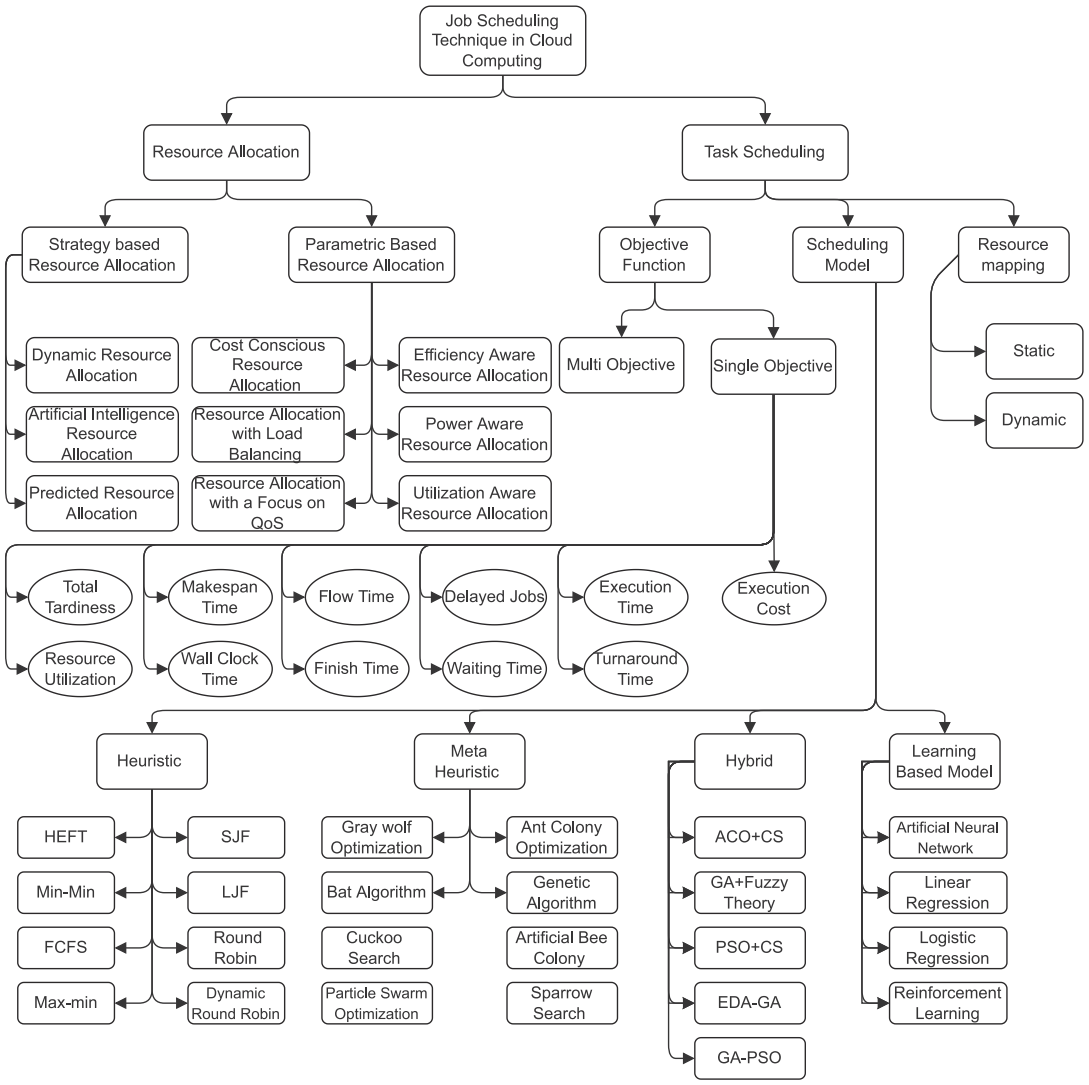

Fig. 3. Taxonomy of JST.

sumers must be much more concerned with the preservation of basic levels of service quality (QoS). For this, academics are tackling the problem of minimizing uncertainty by forecasting task execution times and queue waiting times in order to increase efficiency and resource usage.

3.9. Energy-efficient allocation

Data centers on the cloud are enormous, requiring a lot of processing and computer resources to run. The carbon footprint of these data centers is expected to be massive Khan et al. (2020). As a result, strategies for allocating resources in an energy-efficient manner must be considered.

3.10. Scheduling tasks in parallel

The task's make span would be extended if tasks were computed in parallel. Independent and dependent tasks are the two sorts of tasks. In the meantime, separate tasks can be conducted on numerous VM. However, because dependent tasks involve communication concerns, it must be done with caution.

3.11. Cloud network

Cloud data centers constructed inside their own domain have a diversity of resource distribution techniques and tactics at their disposal. For dispersed clouds, traditional resource allocation algorithms couldn't keep up with newer ways for allocating resources. Distributed cloud concerns include communication delays, networked system virtualization optimal resource allocation, and so on. Virtualized network resource mapping (VNE) is a challenging task Papagianni et al. (2013).

4. Taxonomy of JST in cloud computing

Job scheduling and Resource allocation (RA) are the two main types of work scheduling algorithms used in the cloud. Furthermore, divide these groups into subgroups and offer thorough classifications as illustrated in Fig. 3. The goal of this research is to lay

Table 1
Strategy Based Resource Allocation.

Ref.	Used Algorithm	Technique	Resource Allocation Type	Advantage	Disadvantage	Tool
Wang and Su (2015)	DHRA and Min-Min.	Depending on the amount of processing power and storage required.	Dynamic RA	Improve the quality of your work	In comparison to the conventional algorithm	CloudSim
Shang (2021)	Fussy clustering, HEFT and FIFO.	workflow and resource clustering.	Dynamic RA	improves the resource utilization and load balance	Reduce the average speed compared to other algorithms	CloudSim
Praveenchandar and Tamilarasi (2021)	FCFS and Round Robin.	Prediction and dynamic resource table updating technique used.	Dynamic RA	Improve job scheduling and power management	Compare only the two common static algorithms.	CloudSim
Abedi and Pourkiani (2020)	Artificial intelligence (AI) based task distribution algorithm (AITDA) and ANN.	Using a smart broker, interaction between cloud and fog servers.	Artificial intelligence RA	Minimize Internet traffic and response time.	The number of performance metrics is very little.	MATLAB
Geetha and Parthasarathy (2021)	ACO and ANN-GA	Maintaining incoming job request.	Artificial intelligence RA	Fault tolerance has been minimized	Robustness of algorithms is increased	MATLAB
Vinothiyalakshmi and Anitha (2021)	Combinatorial Double Auction Resource Allocation (CDARA) and Credibility-based Multiattribute Combinative Double Auction (CMCDA).	Use credibility for resource provisioning	Dynamic RA	minimizes the complexity of obtaining resources for job completion.	Risk analysis and service level agreement is not clear here.	Cloud-Auctio
Gu et al. (2017)	Latest reservation online (LRO) mechanism	Virtual machines can be assigned to users in real time using this system.	Predicted RA	Improve the performance	Only one virtual machine (VM) is focused on a time.	CloudSim
Dabbagh et al. (2015)	A framework for energy-aware resource provisioning.	Estimates the number of future virtual machine requests	Predicted RA	Preserve data center power	Concentrate solely on cloud service providers	MATLAB
Xiong and Xu (2014)	particle swarm optimization (PSO), MBFD and MBFH	The optimum balance of resource usage and energy consumption	Artificial intelligence RA	Minimize energy	Algorithms compared to traditional ones	CloudSim
Oddi et al. (2013)	Algorithm for multi-cloud resource allocation using Markov decision processes (MDP)	Management of multi- cloud resources	Dynamic RA	increased revenue and greater performance	Practically, this has not been implemented.	MATLAB
Pu et al. (2020)	online VM allocation and pricing (OVMAP) and Vickery-Clarke-Groves (VCG-VMAP)	online technique to solve the VM pricing and allocation issue	Predicted RA	Enhance the overall performance	Not all future demand can be predicted by a mechanism.	CPLEX 12
Wang et al. (2014)	Schema for distributing resources that conserves energy and makes use of predictions (ECRASP)	Distribute the incoming task to the PM with the lowest workload.	Predicted RA	Enhance the overall	performance Practically, this has not been implemented.	Eclipse
Manzoor et al. (2020)	Ant Colony Optimization (ACO), Simulated Annealing (SA) and Genetical Algorithm (GA)	Resource allocation	Artificial intelligence AR	better performance and has minimum response time	Depend on the grid system	CloudSim
Mousavi et al. (2017)	teaching–learning-based optimization algorithm (TLBO) and grey wolf's optimization algorithm (GW)	Proposed new algorithm for RA	Dynamic RA	Improvement of local optimization and increasing the accuracy	Number of comparison algorithms is limited.	MATLAB
Hu et al. (2013)	Ant colony optimization-based allocation algorithm (ACO)	Predicts the capability of resource nodes.	Dynamic RA	High performance and reduced response times	The grid-based algorithm is compared.	GridSim
Xu et al. (2018)	FF, BF, FFD, BFD, and DRAM	IoT application in fog computing.	Dynamic RA	Increase the number of fog services for RA	Service migration and data transmission cost degradation.	CloudSim
Chien et al. (2019)	Long short-term memory (LSTM), GA-based resource allocation algorithm (GARAA) and (LSTM + GARAA)	combines cloud computing and edge computing	Artificial intelligence RA	High resource utilization and low power usage	Resource allocation becomes quite complicated	Anaconda
Li and Li (2013)	The methodology for IaaS providers, cloud consumers, and SaaS providers to allocate resources (RASP)	composition of SaaS and IaaS, and its joint optimization	Artificial intelligence RA	Increase usage of resources	The success rate isn't higher.	CloudSim
Goutam and Yadav (2015)	Algorithm for fault tolerance and Cloud min–min algorithm.	Dynamic resource provi- sioning	Predicted RA	Improve resource use and fault tolerance.	Based on high priority, allocation is made.	CloudSim
Ali et al. (2013)	Artificial Neural Networks were created by Cartesian Genetic Programming (CGPAN N)	the ability to foresee customer needs in data cen- ters	Dynamic RA	Enhanced Capabili- ties	Focus solely on the computer's processing power.	MATLAB

(*continued on next page*)

Table 1 (*continued*)

Ref.	Used Algorithm	Technique	Resource Allocation Type	Advantage	Disadvantage	Tool
Zhang et al. (2011a)	A system for allocating resources in real time	Accomplish a response time and service rate that meet the quality of QoS	Dynamic RA	In the dynamic cloud, the proposed framework performs better.	Practically, nothing has been done.	MATLAB.
Samriya and Kumar (2022)	Spider Monkey Optimization (SMO)	social behaviour of spider monkey.	Dynamic RA	response time, makespan, energy consumption.	Makespan Time is higher.	Cloudsim

the groundwork for future research into the scheduling technique utilized in cloud computing.

4.1. Allocation of resources

As end-users are able to access resources from everywhere at any time, one of the difficulties with cloud computing is RA. Soap/Restful web APIs, which connect requests for storage or computation to virtualized ICT resources, are the only way to get at the resources accessible in the cloud (such as servers, elastic IP, blob storage, and so on) Kiruthiga and Akila (2021). Because cloud data centers have a high artificiality of resources, the cloud computing paradigm may provide elastic resource allocation on demand. However, such a high level of Artificiality could result in wasteful resource allocation Hameed et al. (2016). Resource allocation is divided into two divisions depending on technique behavior and environment, as detailed below: strategy-based RA and parametric-based RA.

4.1.1. Strategy based RA

Strategy-based RA is the technique that is used during the process of RA. Based on the approach's environment and behavior, we may split the technique into three categories: artificial intelligence RA, dynamic RA, and predicted RA. Table 1 demonstrate a study and analysis of strategy-based resource allocation, as well as their benefits and drawbacks. The following are the details of the above-mentioned classification.

Dynamic RA: In cloud computing, the capacity to meet the changing expectations of cloud consumers is considered as a difficult subject. In order to handle and satisfy these unpredictably high expectations, dynamic RA is used to adapt to the needs of users across a variety of workloads and environments. Also, give a guarantee of QoS in order to avoid SLA aggression Jayanthi (2014). Saraswathi et al. (2015) provide a novel approach to completing high-priority assignments. This method does not consider the most recent VMs that have been created for the new task. A high-priority activity is carried out in the VM, resulting in the suspension of lower-priority operations. A dynamically hierarchical resource allocation approach (DHRA) has been proposed to address the issue of producing enormous quantities of information's while allocating resources. Cloud computing's proposed solution is able to satisfy large-scale application service demand while also enhancing system security. The DHRA's effectiveness and practicality are demonstrated through evaluation and testing, and communication traffic and messages are reduced Wang and Su (2015). Ali et al. (2013) provide an IaaS performance management architecture that explains the primary OpenStack-based application. The basic structure consists of a group of managers who allocate resources based on user demands and work together to achieve a management objective. The manager's intentions contain typical components that support a specific management goal. After that, estimate a prototype implementation for the two specific goals of cost and efficiency. The ant colony optimization (ACO) resource allocation technique has been developed by Hu et al. (2013) to allot and transfer IaaS resources in the cloud. To arrive at a set of ideal compute nodes, the new ACO algorithm first projected the capability of possible resource nodes, then looked at some features of actual network quality and response times. Finally, jobs are assigned to the appropriate nodes. Due to significant energy usage, allocating on-demand resources to customers from a single cloud provider is a difficult task. Aside from that, to create adequate cash and meet the needs of the users.Zhang et al. (2011a) employ model predictive control (MPC), which is based on discrete-time optimum control and aids in the discovery of solutions. Furthermore, the building of a proper information model necessitates the application of tight conditions.

Table 2
Parametric Based Resource Allocation.

Ref.	Used Algorithm	Technique	Resource Allocation Type	Advantage	Disadvantage	Tool
Ma et al. (2019)	Normal GA algorithm (GA_N), deadline, cost-aware genetic algorithm (DCGA) and particle swarm optimization (PSO).	The topological structure of a task is used to divide it into multiple levels.	Costeffective RA	Reducing costs while meeting a tight time-frame	Cost of communication among VMs is higher.	LIGO, Montage, and Cybershake
Lee et al. (2014)	Integer quadratic program (IQP) and Column generation (CG)	Virtual machine placement	RA with load balancing.	Power consumption optimization for servers, networks, and migration	Increase complexity of implemented algorithms.	IBM, ILOG and CPLEX
Geetha and Robin (2021)	Dynamic VM placement, VM provisioning and Data center Provisioning.	Creating two layers such as Green Manager Layer (GMLs) and Cloud Manager Layer (CML)	RA with focus on Quality of service (QoS)	Minimal response time	Consideration of performance matrices is little.	CloudSim
Ibnyaich et al. (2021)	Congestion aware resource allocation, energy efficient, and routing protocol (ECRR)	Large-scale devices and gateways are distributed throughout the network using a *meta*-heuristic algorithm and data clustering.	Efficiency Conscious RA	Improve IoT communication by developing an efficient and intelligent protocol.	This is not practical.	NS2
Tarahomi et al. (2021)	Micro-genetic algorithm	Cloud server consolidation and live VM migration are used.	power consumption RA	improved power consumption	Cost will be increased.	CloudSim
Rezvani et al. (2015)	Integer linear programming	Allocation and migration of virtual machines	Utilization Conscious RA	Enhance performance	In comparison to conventional algorithms	Haizea
Kumar and Saxena (2015)	Resource allocation depending on customer demand	The allocation of resources based on compensation	Costeffective RA	Better performance	Priority is used in allocating resources. *	CloudSim
Mohana (2015)	a support vector machine (SVM), An artificial neural network, and a PB-PPSO optimization (ANN)	Optimal resource allocation	Costeffective RA	High total and average profit reaction time efficiency	Efforts are focused on learning the rules of engagement for new users.	CloudSim
Liu et al. (2016)	Method for allocating resources with a multiquality of service load balance (MQLB-RAM)	Allocation of virtual resources using a resource division approach and advance reservation.	RA with load balancing.	Improve productivity while keeping expenses to a minimum.	The results don't show that the load is evenly spread out.	CloudSim
Zhang et al. (2011b)	A statistically driven technique to load balancing (SLB)	Allocates resources for load balancing while the VM is initializing.	RA with load balancing	Real-time load balancing	The emphasis is on time, and no other resources are discussed.	XenServer
Horri et al. (2014)	SLA-aware algorithm	VM consolidation	RA with focus on Quality of service (QoS)	Reduce VM migration, SLAV, and total data sent.	VM loads are to occur at the same time.	CloudSim
Katyal and Mishra (2014)	The min–min and max–min algorithms form the foundation of the selective algorithm.	Resource allocation and scheduling	RA with focus on Quality of service (QoS)	Reduce the makespan to increase throughput.	FCFS is only compared to other algorithms	CloudSim
Pradhan et al. (2016)	Modified round Robin algorithm	Time Quantum	Efficiency Conscious RA	Improve the output.	Concentrate solely on cloud users.	MATLAB
Xu and Yu (2014)	Game theory FUGA algorithm	Multi-resource allocation	Efficiency Conscious RA	Increase the efficiency of fair allocation.	In comparison to conventional algorithms.	Google workload
Geetha and Robin (2021)	Cloud Manager Layer (CMLs) and Green Manager Layer (GML).	CML choose the suitable resources among all and GML picks the best one.	power consumption RA	Reduce average response time and power consumption	Maintenance cost will be increase.	CloudSim
Dashti and Rahmani (2016)	First-fit algorithm (FF) and Best-fit algorithm (BF), Particle swarm Optimization, Power Aware Best Fit Decreasing (PABFD)	VM placement	power consumption RA	improve dynamic resource allocation	Compare with traditional algorithm	CloudSim
Patel et al. (2016)	Cuckoo search algorithm	Server utilization	Utilization Aware RA	Enhance dependability and efficiency.	The tests make no provision for data security or storage.	MATLAB
Pillai and Rao (2014)	open coalition request formation algorithm, task allocation algorithm coalition dissolving algorithm.	Underutilization of resources	Utilization Aware RA	Avoid integer programming's complexity and improve performance.	Each job is limited to a single type of request.	GroudSim
Jain and Sharma (2022)	Deadline constrained time–cost effective salp swarm algorithm (DTC-SSA)	Resources are provided to request based on QoS requirements	Costeffective RA	Improve resource utilization throughput	reliability, availability are not consider.	Cloudsim

Artificial intelligence RA: Artificial intelligence (AI), a cloud computing subject, is at the foundation of intelligent resource allocation algorithms that work in the same way as humans. A RA approach that takes into account aspects of automated and intelligent systems, where systems inspired by nature and based on operational research as well as elements of machine learning and neural networks as well as systems using agents and expert systems is included here Endo et al. (2011). In IaaS cloud computing, artificial intelligence improves precision and accuracy by reducing the risk of mistakes and failures.

Cloud data center energy efficiency and performance can be improved by allocating virtual machines with multiple resources. It contributes to a decrease in the consumption of data center power. VM resource allocation that takes advantage of numerous resources can be made more energy efficient by using the Particle Swarm Optimization technique. Only processing and storage considerations are taken into account in these solutions Xiong and Xu (2014). It is possible to provide QoS for IoT-based delay-sensitive applications using both cloud and fog computing, but neither can do it on their own. The importance of fog and cloud server compatibility cannot be overstated. In Abedi and Pourkiani (2020), they discuss an AI-based task distribution algorithm (AITDA) that targets to minimize Internet traffic and response time by distributing jobs between cloud and fog servers. In the current state of the art, Ant Colony Optimization (ACO), that can satisfy the criteria of cloud computing environment, is utilized to optimize, and manage resources. The suggested technique estimates the necessary bandwidth and anticipates available resources in advance Manzoor et al. (2020). In addition, it forecasts the response time and the quality of the network. The optimization of optimal resource allocation for infrastructure as a service (IaaS) and software as a service (SaaS) in cloud computing is provided by Li and Li (2013). When compared to another current algorithms, experimental results describe that the exposed joint optimization technique for effective resource allocation works better.

Predicted RA: When it comes to allocating resources in the cloud, cloud computing professionals focus on predicting future user demand, influencing resource requirements, and automatically distributing resources. There are a number of objectives that can be achieved by utilizing projected resource allocation. In IaaS cloud computing, resource allocation is critical, and this is a must Patel and Dahiya (2015).

To accurately anticipate workload and save energy in cloud centers, an adaptable, effective, and simple framework is necessary to be used. Machine learning classification and stochastic theory are combined to forecast cloud resources. A shortcoming of other approaches is that they require frequent models, which is not the case with our method. It can also be used in cloud data centers to make energy-conscious resource management decisions. The efficiency of the proposed approach is determined using Google data traces Dabbagh et al. (2015). There is also an ECRASP (energy conserving resource allocation strategy with prediction) proposed by Wang et al. (2014). The system is able to make intelligent decisions based on estimates of job arrival trends and other features of future demand. Numerical findings show that the suggested approach reduces energy consumption more than existing RA algorithms. Multiple types of resources, such as virtual machines, CPUs, and storage, are considered while developing an auction-based online (AO) algorithm for cloud Virtual machine (VM) allocation and pricing. The suggested online strategy invokes resource availability, selection, and progress updates to react to the cloud user's request. Pricing for cloud clients is also calculated according on the quantity of resources they use. Faster response times, maximum income, and incentive compatibility are all significant considerations when using cloud-based online services, as shown by the simulation findings Pu et al. (2020). Furthermore, Fault tolerance techniques for advanced resource reservation are provided by considering service deployment for varying SLA Goutam and Yadav (2015). To begin with, it checks for local resource availability and assigns resources to users if they are available or free.

4.1.2. Parametric based resource allocation

The six forms of Parametric-based resource allocation are: cost-effective RA, Efficiency Conscious RA, RA with load balancing, power consumption RA, utilization aware RA, and RA with focus on Quality of service (QoS).

Cloud service providers' profits and income, as well as user spending and resource prices, are all considered in cost-effective RA. The goal of efficiency-conscious RA is to reduce execution and reaction times, increase bandwidth, and priorities jobs to optimize performance. Cognitively RA to multiple users in multiple data centers is a major focus in load balancing. Power-aware RA aims to cut down on the amount of energy and heat used in data centers by utilizing green computing. When it comes to cloud services, QoS aware RA is all about making them more reliable and less prone to SLA violations and other issues such as service interruptions and outages. Utilization-aware RA, as the term suggests, aims to make better use of cloud resources by allocating resources based on usage. Analyses of parametric resource allocation and its benefits and drawbacks are shown in Table 2.

Cost-Effective RA: Cost-conscious RA is an important topic in cloud computing; it is responsible for providing services at a low cost, as defined by the cloud. Cloud providers are in charge of efficiently distributing services to meet the needs of users. In exchange, they expect more profit and income from increased resource usage, whereas cloud consumers expect to receive services at a low cost with good performance Zhang et al. (2010). In this situation, cloud computing relies heavily on efficient resource allocation systems or procedures. Using a market driven auction mechanism, Kumar and Saxena (2015) propose a demand-based biased resource allocation method. It employs a payment plan depending on the buyer's service preferences. You may use this strategy to allocate resources in two ways, one that ensures the winner pays less than what they bid, and one that ensures they don't pay more than what they bid if their bids reflect their ability to pay. Second, a market-driven auction method that ensures the profit and reliability of the service supplier. Meeting service quality objectives while assigning resources to activities is one of the most difficult challenges. In Ma et al. (2019), A deadline and cost sensitive scheduling technique for the infrastructure as a service (laS) paradigm is provided, which lowers a workflow's execution costs under time limitations. Because of the volatility in VM performance and acquisition latency, there is no connection between activities at the same level, thus they divide jobs into multiple levels based on the topological structure. Additionally, the proposed allocation method is compared to the well-known offline VCG auction mechanism, with the findings indicating an advantage in service provider revenues, cloud user fees, and optimal resource use. A new method dubbed PBPPSO (position-balanced parallel particle swarm optimization) is provided for cloud RA Mohana (2015). PBPPSO's primary purpose is to figure out how to maximize resources for a set of tasks in the shortest amount of time and at the lowest cost.

RA with load balancing: For better performance and efficient use of resources, a realistic approach to load balancing data centers or virtual machines (VMs) entails systematically assigning resources and sharing workloads Aslam and Shah (2015). When allocating resources, care must be taken to make sure that they are always readily available for consumers to use, and that they can handle high and low loads equally well Goswami and De Sarkar (2013).

For cloud computing, a good resource allocation algorithm can improve bandwidth, load balancing, delay, and dependability. Goswami and De Sarkar (2013) propose a resource allocation-based multi-QoS load balancing resource allocation method (MQLB-RAM). It combines the demands of users and the services provided by suppliers, assigns VMs to PMs, and links the job to a certain sensor. It also examines the gravity of every index value in order to match demand with resources, maximize resource utilizations, achieve proper load balancing, and lower costs. In addition, power consumption as a function of time has been extensively studied using temporal load aware optimization. There were two separate solutions for migration, one focused on server and/or network power utilization, and the other on VM migration, which was not the ideal strategy. In the context of heterogeneous workloads and servers, the authors present an integrated technique for optimizing server power consumption, network communications, and migration costs via VM placement Vakilinia (2018). The result of optimization is an integer quadratic programmed (IQP) with linear/quadratic constraints in the number of Virtual machines allotted to a job on a server. Furthermore, Zhang, et al. (2011b) explain, the distribution of virtual machines to cloud customers is a combination of mutual prediction and cloud resource allocation. Virtual machines are being used for load balancing, but statistics-based load balance (SLB) is allocating resources on behalf of those virtual machines. The SLB approach has two components: the first is an online statistical study of virtual machine (VM) performance to forecast resource need, and the second is a load-balancing algorithm that selects the most appropriate host based on host prediction and historical load data.

RA with focus on Quality of service (QoS): Resources must be allocated in accordance with QoS in the cloud. Availability, fault tolerance, recovery time, dependability, throughput, and service level agreements are all key terms in this discussion of resource distribution in the cloud based on the QoS needs of both cloud providers and customers Abdelmaboud et al. (2015). When allocating resources, the QoS must take into account increasing failure rates, a lack of resources, inefficient use of resources, and SLA aggressiveness Ardagna et al. (2014).

For assigning on-demand resources to cloud end customers, this system Katyal and Mishra (2014) was chosen as a recommendation. Using the traditional scheduling algorithm's principles of max–min and min–min, the proposed method assigns resources to users based on their scheduling needs. This approach, also known as the max–min algorithm, is chosen because it uses less computer power and employs heuristics. The CloudSim simulator is used, and resources are allocated on a first-come, first-served (FCFS) basis in the proposed method. In a similar vein, Lee et al. (2014) describe a reliable method for allocating virtual machines to physical machines that works in conjunction with the best-fit strategy. Each host node's processing and storage characteristics are taken into account while constructing a performance analysis methodology to facilitate the VM migration. In the proposed resource allocation system, virtual machines are distributed to the best node for providing the service while taking into account the user's requirements as well as the high and low composition of each node. Findings from an experiment reveal users may meet their needs in real-time thanks to the suggested framework's increased use of resources without losing the time needed to allocate resources. The data center's power usage allows for capabilities such as web-based monitoring, real-time virtual machine mobility, and virtual machine allocation advancement. This Geetha and Robin (2021) research focuses on a RA strategy for cloud users that cannot affect QoS by utilizing two layers, the Green Manager Layer (GML) and Cloud Manager Layer (CMLs). The CML oversees selecting appropriate resources from among all accessible resources, and GML selects the better one.

Efficiency Conscious RA: Cloud computing customers' satisfaction is directly related to how efficiently resources are allocated, and this directly affects performance. As a result, cloud customers benefit from increased bandwidth, faster application execution times, better prioritization, and faster response times when allocating resources in the cloud Wood and Alsawy (2018).

Pradhan et al. (2016) propose a hybrid round robin strategy for meeting cloud user expectations while minimizing response time. The distinction between dynamic and set time quantum computations is also proven to improve cloud computing RA. Quantum time is believed to be a fundamental component of the RR algorithm. Additionally, addressing user requests for real-time dynamic adjustments is particularly challenging. The meta-heuristic ant colony algorithm is thought to be able to tackle these kinds of problems, however it has sluggish convergence and parameter selection concerns. Wi-Fi, ZigBee, Bluetooth, WiMAX, 4G, and LTE are examples of wireless technologies that have emerged as a way to make communication in Smart City activities. However, because many of them use coexistence, unlicensed interference, and band difficulties are becoming more prevalent. As a result, smart cities leverage IoT to fix the problem. Based on hybrid optimization techniques, this Ibnyaich et al. (2021) study finds the difficulties of both RA and routing to present congestion aware, an energy efficient RA and the routing protocol (ECRR) for IoT networks. As a result, Xu and Yu (2014) look at the topic of cloud computing RA. Resource utilization is studied on a virtual machine (VM) level. Additionally, a recommended allocation FUGA algorithm encourages efficient RA for cloud customers by assisting in the optimal utilization of resources for every physical server. To solve the problem of RA, the FUGA method is used to represent a vast finite game with accurate information.

Power consumption RA: Algorithms for allocating resources that consider energy consumption and heat generation in data centers have shown to be successful. To save money and reduce energy consumption, cloud providers and data centers must reduce the amount of heat they generate. A surge in the number of servers, high demands, a tremendous load, and the waste of idle power are factors that contribute to energy and heat inefficiency in data center. By minimizing the amount of heat and electricity used in data centers, green computing has the potential to improve resource allocation and utilization Pandi and Somasundaram (2016); Singh (2015).

Dashti and Rahmani (2016) employ the PSO algorithm to dynamically migrate virtual machines in order to update resource allocation and obtain additional benefits in the data center. They may be able to assure faster response times and improved QoS by putting forth a novel heuristic technique for dynamic resource re-allocation that balances the overburdened cloud providers (SLA). Similarly, under load and power, linked cloud providers can achieve greater energy efficiency and power savings. Furthermore, one of the more advantageous strategies is to use power-aware methodologies to decide where to assign VMs in datacenter physical resources. For power aware VM allocation system, virtualization is being used as a potential solution. Because it's an NP-complete problem, the author turns to evolutionary approaches to solve the VM allocation dilemma. This study Tarahomi et al. (2021) presents a useful micro-genetic algorithm for selecting appropriate VM destinations among physical hosts. However, this research Geetha and Robin (2021b), focuses on a resource allocation strategy for cloud users that does not compromise QoS by utilising two layers, the Cloud Manager Layer (CMLs) and the Green Manager Layer (GML). The CML is in charge of selecting appropri-

Table 3
Objective function of Task Scheduling.

Ref.	Makespa Time	n Flow Time	Waiting Time	Executio Time	n Delay Time	Finish Time	Turnaround Time	Wall clock Time	Resource Utilization	Total Tardiness	Execution Cost
Alemnesh (2020)	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗
Al-Maamari and Omara (2015)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Pratap and Zaidi(2018)	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗
Alhaidari and Balharith (2021)	✗	✗	✓	✗	✗	✗	✓	✗	✗	✗	✗
Cui et al.(2017)	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗
Rjoub et al.(2020)	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✓
Goyal et al.(2020)	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗
Gond and Singh (2018)	✓	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗
Alemnesh (2020)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Rjoub and Bentahar (2017)	✓	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗
Sels et al. (2012)	✗	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗
Alkayal et al.(2016)	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗
Eldesokey et al.(2021)	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗
Ebadifard and Babamir (2018)	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗
Holladay et al.(2017)	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✗
Gomathi et al.(2018)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Chen and Long (2019)	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓
Kumar and Venkatesan (2019)	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗
Al-Maamari and Omara (2015)	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗
Manasrah and Ba Ali (2018)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Navimipour (2015)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
Al-Maamari and Omara (2015)	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗
Hassan et al. (2015)	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
Mansouri and Javidi (2020)	✓	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗
Gąsior and Seredyński (2019)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Nazir et al. (2018)	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✓
Ananth and Chandrasekaran (2015)	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗

ate resources from among all accessible resources, and the GML selects the best one.

Utilization Aware RA: Resource usage plays a big role in cloud computing's success. Despite the fact that cloud providers' data centers always have a limited number of resources, they make every effort to maximize their use by allocating resources wisely Li and Wu (2014). An difficulty that arises when a large number of people use the cloud is how to meet all of their needs while maximizing use of all resources Tchendji et al. (2016).

Pillai and Rao (2014) show how game theory's uncertainty standards can be used to predict the establishment of associations between machines in the cloud. This proposed approach has the advantage of avoiding the intricacies of integer programming by describing the coalition building optimization problem. In addition, the resource allocation system aims to reduce resource waste, reduce job allocation time, and increase user satisfaction. Virtual machine placement on real machines, particularly for complex reservation models, has been identified as a problem. After that, provide a solution based on integer linear programming in order to tackle certain community scenarios of the issue (ILP). A last step entails running the software on a Haizea simulator, which links simulation values to the Haizea greedy algorithm and a variety of heuristic strategies Rezvani et al. (2015) further enhance cloud computing storage systems' security, Patel et al. (2016) discover and increase resource use. Server and user authentication are selected using the Cuckoo search strategy. Resource reliability and efficiency are improved as a result of its use. As demonstrated in MATLAB, a proposed technique is superior to GA and SLPSO.

4.2. Task scheduling

Depending on the quality of service (QoS) standards, the purpose of work scheduling varies from application to application. Because of this, there have been numerous investigations on the topic of task planning. Fig. 3 depicts a new, rigorous taxonomy for better understanding cloud computing task scheduling approaches. It is based on a variety of significant methods used in the literature. In terms of the objective function, scheduling model, and resource mapping, these methods are classified into three primary groupings in this section. The objective function portion is additionally separated into two sub-sections, each of which is explored in detail. The scheduling model is also broken down into four sections. Resource mapping is divided into two parts based on this.

4.2.1. Objective function

An optimization model must be constructed that finds the best optimal solution while still meeting the objectives, as there is constantly some trade-off into the optimization objectives. It is possible to compare the efficacy of different solutions using single-objective optimization. To do so directly in Multi-Objective Optimization Problems is not possible (MOPs). Pareto dominance relation techniques are frequently employed by MOPs to build a comparison model that replaces a single optimal solution with an array of possibilities, which allows for a wide range of trade-offs between objectives. For the sake of evaluating performance, just one of the numerous Pareto optimal solutions offered in MOPs

need be selected Houssein et al. (2021). By highlighting their most relevant aspects, we present an outline of the mechanisms in certain selected studies based on single and multi-objective optimization approaches. Single-objective and multi-objective scenarios are summarized in Table 3.

Single Objective: When it comes to job scheduling in cloud computing, most current solutions only consider the CPU and memory requirements, leaving the makespan requirement out entirely. For getting better performance in Cloud environments, Navimipour (2015) proposes a novel bee colony algorithm for scheduling activities on service providers. The findings showed that the suggested approach has a faster makespan time than other algorithms. Furthermore, three heuristic methods for job scheduling in the cloud environment have been compared in this research AlMaamari and Omara (2015). PSO algorithm, genetic algorithm, and modified PSO algorithm are three methodologies for effective work scheduling. The purpose of all three algorithms is to develop an ideal plan that minimizes the makespan time.

Optimizing computationally demanding models of real-world systems can be difficult, especially when a single model evaluation requires a large amount of wall clock time. Using both synchronous and entirely asynchronous particle updates, this work Holladay et al. (2017)investigates the effect of model run time variance on the behavior of PSO. The findings show that, in most circumstances, asynchronous updates save a large amount of time while having no meaningful impact on the likelihood of discovering a solution.

Multi Objective: Optimizing task scheduling in a distributed heterogeneous computing system to improve cloud resource consumption while maintaining service quality is an NP-hard optimization problem. With the propose of minimizing job execution/transfer cost and time, Ramezani et al. (2014) established a multi-objective model with antithetical objective functions.

The Cloudsim toolkit is used to extend the Jswarm package into a multi-objective framework (MO-Jswarm). The presented optimization model attains the better balance solution and the maximum QoS when compared to existing job scheduling techniques. However, job priority, or the sort of work performed energy consumption are not considered.

Cloud computing requires job scheduling in order to maximize performance and manage resources effectively. Cloud computing environment and optimization job scheduling schemes are introduced in this research by Cui et al. (2017). They concentrate on large-scale cloud computing systems and efficient job scheduling using VM resources and SLAs. A gateway, a job scheduler, and a pool of resources are only some of the system's components. Because of the constraints on resources and deadlines, they came up with an innovative way to lower Average Waiting Time (AWT) and the Makespan utilizing reinforcement learning and parallel multi-age parallel technologies.

Makespan Time: A makespan is the whole time it takes to finish a workflow. It's the total time it takes for all workflow processes to complete Belgacem and Beghdad-Bey (2021). In the domain of computer science and information technology, job scheduling is a combinative optimization problem in which the best jobs are assigned to the best resources at a given point in time. Rajuet al. (2013) proposed a hybrid method in this research that put together the benefits of Cuckoo search and ACO. With the use of a hybrid algorithm, the makespan (or completion time) can be lowered.

Wall Clock Time: For the purposes of practical computing, wall clock time is the duration of time it takes a program to run or accomplish its assigned tasks, which is often measured in seconds Murad et al. (2021). In multitasking mode, the wall clock times for each program are configured independently and are depending on how the computer's CPU distributes resources between the running apps. In order to improve wall clock time, the author investigates the impact of model run time variance on PSO behavior utilizing both synchronous and entirely asynchronous particle updates in Holladay et al. (2017).

Resource Utilization: The usage of resources in a datacenter is a significant part of minimizing energy consumption. To make efficient use of resources, cloud computing employs a variety of ways Surendran and Tamilvizhi (2018). For more reliable resource allocation in clouds (SSO), Eldesokey et al. (2021) suggested a hybrid swarm optimization (HSO) technique, which is a combination of salp swarm optimization and PSO. HSO's main goal is to allot the tasks to all available resources in such a way that computation costs and execution time are decreased. Multilayer logistic regression (MLR) is a technique for detecting overloaded virtual machines (VMs) so that tasks can be assigned to them based on their workload capacity.

Flow Time: Flow time is the total time required to go from one operation to the next, including any time spent waiting for machines or work orders to arrive, any time lost due to machine breakdown, and any time lost due to process delays or component shortages. According to Gond and Singh (2018), the author concentrated on a compare of two algorithms, SJF-MMBF and PSO, on metrics such as completion time variance, total flow time, RPD total flow time, RPD makespan, and so on, in order to determine which algorithm was superior.

Waiting Time: The amount of time a process spends in the ready state before the CPU responds is referred to as waiting time. All cloudlets must wait in the main queue before they can be assigned a processor. Alemnesh (2020) suggested approach, Time Optimized Hybrid Job Scheduling Algorithm for Cloud Computing, is intended to minimize the amount of time that users must wait. The method operates on the Shortest Job First (SJF) and Round Robin (RR) principles, but with a dynamic quantum time component to it. Furthermore, this work Alhaidari and Balharith (2021) proposed a unique technique named the dynamic roundrobin heuristic algorithm (DRRHA), which is dependent on the average of the time quantum and used the RR algorithm while tweaking its time quantum in a continuous basis.

Finish Time: It is the finish time of a process that determines when the process has completed its execution and has exited the system. In order to complete the project more quickly, An method based on the optimization of bee colonies is proposed in this paper Mousavinasab et al. (2011). Algorithms employ artificial bees to properly plan workloads across grids of computers and other devices. Using the algorithm proposed in the grid computing environment. According to this Kumar and Venkatesan (2019) study, an efficient Hybrid Genetic Algorithm–Ant Colony Optimization (HGA–ACO)-based algorithm for allocating work to address the massive amounts of requests from cloud users can be found. In the proposed HGA–ACO, reaction time, completion time, and throughput are taken into account to determine the appropriate job allocation mechanism.

Delay Time: Scheduling time encompasses both the time it takes to move a task from one part of the system to another and the time it takes to send a task result back from one part of the system to another. Reduce the size of tasks or task outcomes if scheduler delays are excessive. Using a bee colony optimization technique, the authors in this study Mousavinasab et al. (2011) suggest a new task scheduling algorithm. The system uses artificial bees to distribute tasks to the grid's resources in an efficient manner. The maximum delay of jobs can be decreased by using the proposed approach in grid computing environments.

Execution Time: The amount of time it takes a job to complete a task is referred to as execution time. The nature of the work environment and the results required dictate the scheduling of a job. According to this study Goyal et al. (2020), the authors tested four

Table 4
Summary of the heuristic algorithms with their findings and weakness.

Ref.	Name of Algorithm	Technique	Backfilling	Findings	Weakness	Tool
Ilyushkin and Epema (2018)	HEFT	Fair Workflow Prioritization (FWP) policy	Yes	Reduces the variability of the job process	extremely high system uti- lizations	DGSim
Singh (2021)	Min-Min	Managing the load balance of resources and increasing the task processing time.	No	Reduce Makespan Time	Number of parameters is only one.	CloudSim
Pratap and Zaidi (2018)	FCFS	Time shared policy or Space shared policy of VM	No	Comparison between RR, FCFS and SJF	Number of jobs is only 10.	CloudSim
Kodli and Terdal (2021)	Max- Min	Using load balancing	No	Minimize makespan time	Reduce the flow time	CloudSim
Kopanski and Rzadca (2021)	SJF	plan-based scheduling	Yes	lack of burst buffer reservations	computationally intensive	Batsim
Alworafi et al. (2019)	LJF	Hybrid ShortestLongest Job First (HSLJF)	No	Response time is fast, resource utilization is high, and throughput is high.	Cost and load imbalance are not taken into consideration.	CloudSim
Panetta et al. (2010)	Round Robin	In a cyclic fashion, assign tasks.	No	Reduce the time it takes for a response and the associated expenditures.	Algorithms cannot reduce the makespan time without pre-emption because of the risk of resource overload.	CloudSim
Farooq et al. (2017)	Dynamic Round Robin	Dynamic Time Quantum	No	Reduce running time of an algorithm	Increase the Makespan time	CloudSim
Khalili and Babamir (2015)	FCFS	PSO with LDIW Meta-heuristic (swarm)	No	Makespan is short and resource use is high.	Reliability is low, and throughput is low.	CloudSim
Mondal et al. (2015)	SJF	Prioritize the shortest task for execution.	Yes	SJF significantly reduces execution and turnaround times as compared to FCFS and RR.	In the SJF algorithm, there is a risk of famine and load imbalance.	CloudSim
Benny and Wirawan (2022)	Round Robin	Burst Time.	No	Higher response time.	No comperison with others algorithms.	Real environment

job scheduling algorithms in a cloud environment by changing the number of cloudlets and work duration and analyzing the total execution time for each algorithm. Multi-objective PSO (MOPSO) is an algorithm developed by Alkayal et al. (2016) that uses a new ranking technique. The approach results in a 20% decrease in execution time.

Execution Cost: Cost optimization is one of the most important issues in the cloud. When it comes to infrastructure, the interests of cloud service providers (CSPs) and end users are at opposition. There are various pricing options for Amazon EC2 services, such as on-demand resource model, advanced resource reserve, and spot instances. There are pros and disadvantages to any policy. As a result, cloud computing relies heavily on effective resource scheduling Kumar et al. (2019). With the use of an ant colony-based algorithm, which proposes the parameter determination into an algorithm that may reduce the execution cost of the algorithm, it can be done Chen and Long (2019). The integrated algorithm is able to maintain a specified concentration of particles in the fitness level while ensuring that the population is diverse.

Turnaround Time: Turnaround time refers to the amount of time it takes from the point at which a process is ready to the point at which it is complete. CPU scheduling strategies produce various turnaround times for the same set of activities. A hybrid job scheduling technique is introduced in this paper Alemnesh (2020) in order to increase efficiency and performance in a heterogeneous cloud computing environment while also reducing turnaround time. For cloud computing environments, the approach developed is termed Time Optimized Hybrid Scheduling Algorithm. It uses a dynamic quantum time method based on the SJF and RR algorithms. The dynamic round-robin heuristic algorithm (DRRHA) is introduced in this study Alhaidari and Balharith (2021) by using the RR algorithm and dynamically modifying its time quantum depending on the average of the time quantum. the DRRHA considerably surpassed its competitors in terms of the turnaround time, response times, and average waiting time measured by the CloudSim Plus tool in comparison with many other algorithmic studies.

Total Tardiness: Delays in operations are measured by tardiness, while early completion is measured by earliness, in the scheduling context. Operation sequences may be dependent on each other and equipment availability. To reduce the tardiness, in this study Sels et al. (2012), various priority criteria for the job shop scheduling (JSS) problem are compared and validated under various objective functions. To schedule workshop difficulties under two flow time-related and three tardiness-related objectives, 30 priority rules from the literature were applied.

4.2.2. Scheduling model

Heuristic, metaheuristic, hybrid, and training-based scheduling systems are all examples of scheduling systems. Scheduling schemes based on heuristics are the most frequent. The scheduling strategy is further classified in Fig. 3, as shown in more detail. The purpose of this study is to lay the groundwork for future research in cloud computing by developing a foundation for the job scheduling technique that will be used.

Heuristic Algorithms: Heuristic algorithms are problem-specific, with strong performance for some domains but poor performance for others. Heuristic algorithms, in general, provide a precise solution for a specific domain of problem in an indefinite period, but they are incapable of solving difficult optimization problems. Heuristic methods for managing workflows and individual activities or apps in the cloud have been created in large numbers. Many heuristic algorithms have been addressed, and we've divided them into according to the key word of the article, for example Heterogeneous Earliest Finish Time (HEFT) Ilyushkin and Epema (2018), min-min Singh (2021), FCFS Pratap and Zaidi (2018); Khalili and Babamir (2015), max–min Kodli and Terdal (2021), SJF Kopanski and Rzadca (2021); Mondal, Nandi and Sarddar (2015), LJF Alworafi et al. (2019), Round Robin Panetta et al. (2010), Dynamic round robin Farooq et al. (2017). By deploy-

Table 5
The findings and weaknesses of the meta-heuristic algorithms are presented in this table.

Ref.	Name of Algorithm	Technique	Backfilling	Findings	Weakness	Tool
Yuvaraj et al. (2021)	Gray wolf optimization (GWO)	Characteristics of serverless computing	No	Improve the process of task allocation	Manage of runtime resource requirements is difficult.	CloudSim
Zheng and Wang (2021)	Bat algorithm (BA)	The conjugate gradient method and mean square error	No	Minimize imbalance and increase output.	Cost is high.	MATLAB
Alkhateeb et al. (2021)	Cuckoo Search (CS)	1. opposition-based learning method 2. combining VNS and Lévy-fight methods.	No	Find better solutions in the search space	Energy consumption is higher.	Real environment
Ramezani et al. (2014)	Particle Swarm Optimization	Transferring jobs from an overloaded VM, instead of migrating	No	High throughput, Low energy and makespan.	Less scalability, as well as low dependability	CloudSim
Meena et al. (2016)	Genetic Algorithm (GA)	Deadline constraint	No	Low cost and only a minor delay in the due date	Low scalability	Real environment
Jena (2017)	Artificial Bee Colony (ABC)	ABC multi-objective algorithm	No	Save time, energy, money, and resources.	The method does not mention a unique compromise approach for time and cost.	CloudSim
Nguyen et al. (2021)	Sparrow Search (SS)	combines a blockchain network with a fog	No	Optimize power consumption, service latency, and monetary cost simultaneously	Number of performances matrices is little	GridSim
Deol et al. (2021)	Ant Colony Optimization (ACO)	Changes have been made to four entities: name, secondary name, aggregator, and data nodes.	No	Optimize allocation and utilization of resources	Only makaspan time is calculated.	CloudSim
Adhikari et al. (2019)	Bat algorithm	LB-RC with BAT	No	Improve resource use, timeliness, and reliability.	Response time and cost aren't taken into account in the method's QoS balance.	CloudSim
Bezdan et al. (2022)	Hybridized bat algorithm	Multi-objective optimization	No	cost reduction and minimize makespan time.	Number of performances is only two.	CloudSim

ing the jobs on a virtual machine utilizing various scheduling strategies, these techniques attempted to improve the various QoS characteristics. Table 4 lists a number of heuristic algorithms, along with their benefits and drawbacks.

HEFT Algorithm: When it's come scheduling full workloads of activities that come over time, Ilyushkin and Epema (2018) created a workflow scheduling employing HEFT algorithms in which they examine the influence of the absence or restricted accuracy of task runtime estimations on slowness. In this article, they look at seven different approaches to managing time: There are several well-known policies for handling (batches of) workloads in literature, including the HEFT policy for a single workflow applied to the online workload condition. Two of the policies are unique workload-oriented policies, such as one that emphasizes fairness. They examine the performance of these policies in homogeneous and heterogeneous distributed systems with varying accuracy of task runtime estimations.

Min-min and Max-min: Efficient user task scheduling is important for managing physical and virtual resources and achieving improved performance in cloud services. Job Scheduling is a popular form of cloud scheduling since it reduces the total time it takes to complete a specific task. Makespan refers to the total amount of time that virtual machines take to accomplish the tasks that have been allocated to them. In this paper Singh (2021), the min-min algorithm is employed to reduce the makespan time. For low time complexity and low cost, Kodli and Terdal (2021) employed max–min. They also did some comparisons between existing algorithms at this point. They achieved a better result than they expected for max–min.

FCFS: The task mapping and scheduling process entails assigning jobs to run on existing resources in a way that maximizes usage and reduces time to completion. The entire time need to complete all jobs is referred to as the makespan. In this study, FCFS is used to decrease the makespan time. In another study published in 2015 Pratap and Zaidi (2018), they conducted a comparison of numerous algorithms. FCFS received a better outcome than the other algorithm.

SJF and LJF: Kopanski and Rzadca (2021) look at how burst buffer reserves affect the overall proficiency of online job scheduling for typical algorithms. EASY backfilling using the Shortest-Job-First (SJF) method. They test the algorithms with I/O side effects in a thorough simulation. Their findings suggest that backfilling without burst buffer reservations can drastically degrade scheduling. Mondal et al. (2015) represent an effective scheduling algorithm based on SJF that can maintain load balance and provide enhanced task scheduling techniques. This would decrease the average response time and increase the number of VMs available to assign new jobs from requesting nodes. The authors of this article Alworafi et al. (2019) developed a hybrid algorithm based on LJF. To begin, the algorithm prioritizes all the tasks that have been submitted. After then, it makes a choice between two jobs, one based on SJF and the other on LSF. Finally, it selects the virtual machine that can do the work in the shortest amount of time.

RR and DRR: Load balancing is essential for jobs submitted to the service provider to appropriately manage the service provider's resources. Load balancing also aids in the optimization of the centralized server's performance. In this article Panetta et al. (2010), the Round Robin algorithm is utilized to improve load balancing. The authors of this paper Farooq et al. (2017) used Dynamic Time Quantum to create an efficient Round Robin algorithm. The purpose of this work is to lower an algorithm's running time while also addressing efficiency limitations such as context changes, average waiting periods, and turnaround times.

Meta-heuristic Algorithms: In recent years, metaheuristic algorithms have become increasingly popular because of their capacity to handle massive and complex computational problems. Meta-heuristic algorithms have a number of useful properties, including:

1. These methods aren't limited to a particular problem.
2. A method that uses meta-heuristics to search the search space for NP-Complete problem solutions is very efficient.
3. The majority of meta-heuristic algorithms are nondeterministic and approximate.

Table 6
Summary of the hybrid algorithms with their findings and weakness.

Ref.	Name of Algorithm	Technique	Backfilling	Findings	Weakness	Tool
Tiwari and Bansal	ACO + CS	Combined the ACO and CS	No	Better result for Makespan time and cost.	Compared with traditional algorithms.	CloudSim
Javanmardi et al. (2014)	GA + Fuzzy Theory	VM MIPS and length of jobs.	No	Improved execution cost and time.	Energy consumption is higher.	CloudSim
Manasrah and Ba Ali (2018)	GA + PSO	Hybrid algorithm using GA and PSO	No	Reduce the makespan and the cost	It's work only for homogeneous workflow.	WorkfowSim, CloudSim
Pang et al. (2019)	EDA + GA	Scheduling work for virtual machines in the most efficient way.	No	Enhance load balancing ability and minimize task completion time	Cost is not considered here.	CloudSim
Al-Maamari and Omara (2015)	PSO + CS	Combine task scheduling algorithm called PSOCS.	No	Reduce the time it takes for something to happen and enhance the utilization ratio.	Resource usage is minimal.	CloudSim
Yu et al. (2018)	GA + PSO	Process planning and scheduling (IPPS) integration.	No	Cost and time are optimized.	Nothing clear about multiobjective IPPS optimization	CloudSim
Ravichandran et al. (2016)	PSO + CS	parallel line job shop scheduling.	No	Minimum makespan value	No information about resource utilization and other parameters.	MATLAB

Metaheuristic algorithms are problem-universal, meaning they may be applied to a large scale of problems and produce acceptable answers. One of the most frequent ways for solving NP-hard optimization issues is metaheuristic methods.

NP-Complete problems can be solved in a short period of time using a variety of meta-heuristic methods on the cloud. Since there are so many possible solutions, task scheduling is an NP-Complete issue, which means it will take a long time to determine the best one. Several heuristic algorithms are examined and evaluated in Table 5, along with the benefits and downsides that come with each.

GWO and CS: Yuvaraj et al. (2021) used the Gray Wolf Optimization (GWO) model to optimize the process of work allocation in their research. Furthermore, they use the Reinforcement Learning (RL) method to optimize GWO, which simultaneously optimizes GWO parameters while also optimizing job assignment. The CS algorithm and simulated annealing (SA) is a new hybrid optimization method. This work Alkhateeb et al. (2021) proposes distinct CSA (DCSA) for solving the JSSP. DCSA modifies CSA in four ways. In the initialization process, it leverages opposition-based learning to generate candidate solutions. Second, it combines Lévy-flight and VNS approaches to improve search space exploration. It employs elite opposition-based learning prior to CSA to avoid local optima. Finally, the CSA's candidate solutions are discretized using the JSSP's smallest position value.

PSO and GA: While continuous VM migration is a useful way for balancing system load, it is still time and cost-intensive, and the migration process consumes a significant amount of memory. Task-based System Load Balancing based on Particle Swarm Optimization (TBSLBPSO) by Ramezani et al. (2014) addresses these issues by moving extra jobs from an overcrowded virtual machine rather than migrating totally. When it comes to job scheduling, the cost is a critical consideration. Meta-heuristic cost effective genetic algorithms are proposed in this Meena et al. (2016) research for minimizing the execution costs of a workflow while ensuring that the deadline is meet. Crossover, population initialization, Genetic algorithms' encoding, and mutations operators are among the innovations we've developed.

ABC, SS and ACO: Reduced processing time for users and lower power usage in the cloud infrastructure are both benefits of efficient task scheduling. Using an artificial bee colony algorithm, this study Jena (2017) aims to improve work scheduling efficiency (TA-ABC). It is proposed that a new algorithm be developed to maximize the cloud's energy, cost, resource usage, and processing time. The Sparrow Search Algorithm (SSA) is used in this paper Nguyen et al. (2021) to make it more transparent and protect against attacks from people who aren't who they seem to be. These experiments compare MO-SSA to other famous algorithms for finding the best way to do something (MO-ALO, NSGAII, and NSGA-III). In this paper Deol et al. (2021), an improvised method called Ant colony optimization is used. This improves the job scheduling abilities of Hadoop and makes sure that resources are used efficiently.

Bat Algorithm: This paper Zheng and Wang (2021) introduces a hybrid multi-objective bat method for improving cloud computing service quality. It also considers the features of resource scheduling optimization techniques and develops a bat method. The bat population is categorized to ensure that the algorithm does not fall into a local minimum. The LB-RC load balancing system is a revolutionary longterm process load balancing technology described in this Adhikari et al. (2019) work (load balancing resource clustering). Optimized resource clustering and cluster centers are obtained using the meta-heuristic Bat-algorithm.

Hybrid Algorithms: To deal with the challenge of resource scheduling in cloud technology, a variety of scheduling techniques can be used, however in this part, we will focus on hybrid scheduling techniques. The adoption of a hybrid approach improves task scheduling in the cloud. In this article, we've covered a wide range of hybrid algorithms under many headings dependent on the article's primary phrase, such as ACO and CS Tiwari and Bansal, GA and Fuzzy theory Gharbia et al. (2014), EDA and GA Pang et al. (2019), GA and PSO Yu et al. (2018), PSO and CS Ravichandran et al. (2016). Table 6 show the research and analysis of a variety of heuristic algorithms, as well as their pros and cons.

ACO + CS and EDA + GA: The goal of constraints in an optimization problem is to maximize or decrease the value of a variable. A function of specific design variables can be used to describe the solution. The goal of this paper Tiwari and Bansal is to use a hybrid ACO to solve the IPPS problem by combining two techniques, including ACO and CS. As the availability of cloud users and demands for cloud computing has expanded in recent decades, academics and businesses have become deeply focused on strategies to improve system load balancing ability and reduce job completion time. This research Pang et al. (2019) proposes an EDA-GA hybrid scheduling method which is based on EDA (estimation of distribution method) and GA in order to achieve the two aforementioned objectives (genetic algorithm).

GA + PSO: In this research Manasrah and Ba Ali (2018), a Hybrid GA-PSO method is suggested for efficiently allocating tasks to resources. The Hybrid GA-PSOs method, used in cloud computing, aims to reduce the cost and time of distributed components while

Table 7
Summary of the learning-based algorithms with their findings and weakness.

Ref.	Name of Algorithm	Technique	Backfilling	Findings	Weakness	Tool
Zhang et al. (2018)	Linear regression and logistic regression	Regression model.	No	Effective resource allocation	It's work only for small scale training set	DAS-2
Li and Hu (2019)	Reinforcement learning	control-theoretic trialand-error learning method	No	minimize the makespan time.	No information about accuracy.	Anaconda
Weckman et al. (2008)	Artificial neural network	Combined of GA and ANN.	No	Improved the makaspan time	Only makaspan is calculated.	Google Collab
Ilyushkin and Epema (2018)	Artificial neural network	Directed Search Optimization (DSO) and 3 layers ANN.	No	Minimum Makespan Time	limited to one DAG	Real Environment
Wang et al. (2011)	Artificial neural network	Thermal-aware workload scheduling based on temperature.	No	Minimize Job response time and max data cen- ter utilization	No comparison between existence algorithms	Real Environment
Cheng et al. (2022)	Deep Reinforcement Learning (DRL)	Deep Q-learning Network (DQN) model.	No	Cost optimization of rented cloud instances	Number performance matrices is only two	Anaconda

Table 8
A resource mapping that has been examined and is based on numerous.

Ref.	Static	Dynamic	Deadline Constraint	Tool
Abd Elaziz et al. (2019)	No	Yes	No	CloudSim
Kumar and Sharma (2018)	No	Yes	Yes	CloudSim
Jain and Gupta (2015)	Yes	No	No	CloudSim
Dubey and Sharma (2021)	No	Yes	Yes	CloudSim
Gao and Huang (2021)	No	Yes	Yes	Real Environment
Bagheri et al. (2021)	Yes	No	No	Netbeans
Dubey et al. (2018)	Yes	No	No	CloudSim

also balancing their load across heterogeneous resources' (integrated process planning and scheduling) is a critical technology for achieving a computer-integrated manufacturing system (CIMS). The new IPPS approach proposed in this Yu et al. (2018) paper involves two steps: dynamic and static. For solving the IPPS issue, a hybrid approach based on PSO, and GA is provided.

PSO + CS: Because the user must pay for a resource based on how much time it is utilized, task scheduling is critical in cloud computing. By maximizing consumption and reducing task execution time, it aids in the equitable distribution of load among system resources. The MDAPSO algorithm, which is a mixture of the Dynamic PSO (DAPSO) and the CS methods, is proposed in this study Al-Maamari and Omara (2015). A line's jobs are handled in a predetermined order. The goal of the project Ravichandran et al. (2016) is to figure out how to distribute workloads to various lines in the most efficient way possible. Parallel line JSS is accomplished using PSO and CS in this study.

Learning based algorithms: Learning-based algorithms have become increasingly popular in recent years. The reasons for this is that those models can make decisions the same way that humans do. Machine learning algorithms and Deep Learning algorithms are two types of learningbased algorithms that are widely available. In this session, we'll go over the methods used for job scheduling in cloud computing, including numerous training-based techniques. With the use of a learning-based approach, we have examined and reviewed numerous task scheduling algorithms for cloud environments. We have also discussed their benefits and drawbacks (shown in Table 7). The selection of the most appropriate optimization method for a certain problem can be quite beneficial in discovering the precise answers to that problem.

Artificial neural Network: A neural network (NN) scheduler is the focus of this paper Weckman et al. (2008) because it is the best way to schedule jobs. Genetic algorithms (GA) are used in this hybrid intelligent system to come up with the best schedules for a known benchmark problem. A neural network is used to store information about how operations should be placed in a sequence. On the benchmark problem, the trained NN was able to do as well as the GA did. However, this article Wang et al. (2011) made use of artificial neural networks (ANN) to make predictions. They conduct their studies using actual data from a data center's general operation. They create a thermal impact matrix that captures the particular interaction between the data center's heating systems in order to clarify the information.

Reinforcement Learning: Scheduling jobs in a cloud data center is essential. For the bin packing problem this research presents the DeepJS, a method for scheduling jobs based on deep reinforcement learning Li and Hu (2019). DeepJS can acquire a fitness calculation straight from experience, which will reduce the makespan (maximize throughput) of a group of jobs.A Deep Reinforcement Learning (DRL) based job scheduler has been developed to address this issue in this work Cheng et al. (2022). Their primary goal is to reduce the amount of time and money spent executing jobs on virtual instances while still providing high quality service (QoS) to the end user.

Linear Regression and Logistic Regression: For cloud computing, resource allocation in auctions is a difficult challenge. On the other hand, the RR difficulty is NP-hard and can't be fixed in polynomial time. Zhang et al. (2018) evaluate the multi-dimensional cloud resource allocation (RA) problem using machine learning (ML) classification and provide two resource allocation prediction techniques that rely on linear and logistic regressions in order to address this difficulty.

4.2.3. Resource mapping

As a result of the current state of the workload and the cloud environment that has been submitted, both static and dynamic mapping of cloud resources to new jobs is required. Resources and workloads, as is widely known, have qualities that are subject to change, and they are also malleable in nature Houssein et al. (2021). The abovementioned allocation techniques are devised and used in order to meet QoS requirements while also minimizing SLA violations. The most prevalent mapping systems, as well as their characteristics, are examined in greater detail in Table 8.

Static: In order to make a scheduling choice before a task begins to execute, static scheduling requires previous information of the tasks. Bagheri et al. (2021) present a dynamic data replication approach for improving the performance of software systems in this study. To choose which file to reproduce and how many copies to make, they weigh in the degree of popularity and the quantity of replicas. How-

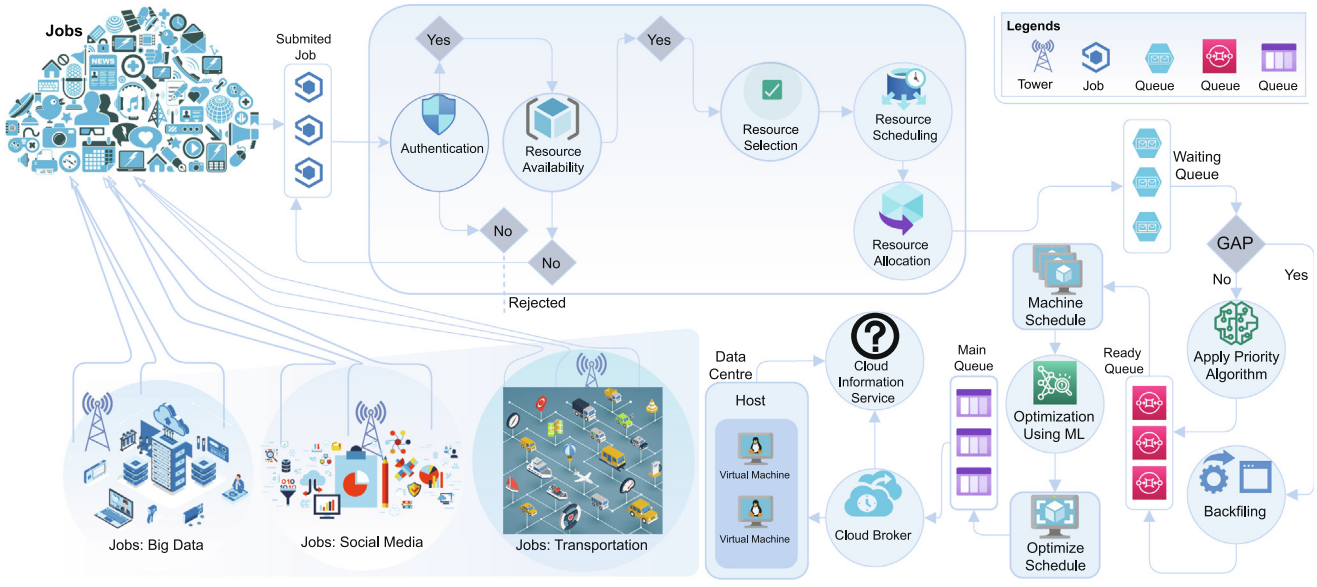

Fig. 4. Proposed Conceptual Framework for JST in Cloud Computing.

ever, in this research Dubey et al. (2018), author present a modified HEFT method that distributes workload to all processors in an efficient and useful manner, hence reducing makespan time of programs.

Dynamic: A new solution for cloud task scheduling difficulties is presented in this research Abd Elaziz et al. (2019), with the goal of reducing the amount of time necessary to schedule a lot of activities on multiple VM while maintaining performance. For the proposed technique, the Differential Evolution (DE) algorithm is utilized to enhance the Moth Search Algorithm (MSA) (DE). Two ideas are merged to imitate the natural behavior of insects, which is to move towards a beam of light. Workstations (physical machines) and unanticipated user demand make it challenging to control energy consumption and maximize profit in the cloud computing environment. With a deadline as a constraint Kumar and Sharma (2018), the works have presented a resource provisioning framework for efficiently processing the jobs, as well as the PSO-COGENT algorithm, a PSO based scheduling technique that not only optimizes execution time and cost but also reduces cloud data center energy consumption.

5. Conceptual framework of JST

Illustration of present Job Scheduling Technique Research has revealed that the various aspects and signs that contribute to good job scheduling are not properly defined, categories, or model in a comprehensive context that can be implemented into applications as is currently the case. Research in cloud computing for job scheduling is more sophisticated and has yet to be discovered, even though it exists. We conducted an in-depth examination of current researcher materials before developing a novel strategy for successful job scheduling in cloud computing environments. Several phases have been defined for our proposed system. These phases include resource allocation, backfilling, optimization (using machine learning), and so on. Fig. 4 depicts a proposed conceptual framework for an effective job scheduling technique in cloud computing.

5.1. Resource allocation

In cloud computing, distribution of resources is the only factor that affects resource management. Using the Internet, the cloud computing environment distributes available resources to the appropriate cloud application. In our approach, we demonstrate that in order to ensure optimal resource allocation, it is necessary to first verify the authenticity of all submitted jobs. Authentication is required because the job request is coming from several places such as social media, transportation. Besides, sometimes hackers try to submit large numbers of requests to bring the server down. If the authentication is successful, the next step is to determine whether or not the resource is available. The cloud service provider will allocate resources to the cloud computing system based on resource availability. If the resources are not attainable, this project will need to be rescheduled, and if the resources are available, this work can proceed to the resource selection stage. During the resource selection phase, the service provider makes available the resources that are required for this particular activity. After receiving the resources, it is scheduled for fixed execution before being transferred to resource allocation.

5.2. Applied algorithms and backfilling

The algorithms are used to create a queue based on the importance of the jobs in the queue. The algorithm can be heuristic, meta-heuristic, hybrid, or training-based in nature. Backfilling is another significant strategy for increasing the efficiency with which the CPU is utilized. Increasing the number of jobs finished before the deadline and maximizing the use of available resources can be accomplished by regularly filling in the gaps with acceptable assignments. Following the receipt of the resource allocation, a new scheduling schedule known as the waiting queue is generated. The scheduler will look for holes in the present resource schedule that may be filled by available resources (machine schedule). For jobs in which there is no gap at all or where the gap does not match the job requirements, the job will be placed in the ready queue and sorted based on the intended PR, which implies that jobs at or near the top of the ready queue will not necessarily be the first jobs to arrive in the Grids. Backfilling, on the other hand, will be used to place the work into a machine schedule if there is more than one Gap that matches the job specifications.

5.3. Optimization using ML technique

The optimization module improves on the initial schedule established by the backfilling approach by merging the meta-heuristics-based technique with the scheduler and so increasing its efficiency.

It is revealed in this phase the technique that was utilized to optimize the backfilling technique utilizing local search in order to successfully harness the advantages of meta-heuristics in order to improve the initial scheduler, which was previously unknown. As machine learning (ML) is becoming increasingly popular, and because the performance of machine learning for optimization is improving, we included the ML technique for optimization in our framework. It is important to optimize the model's hyper parameters to increase its performance. Exhaustive search, gradient descent, and genetic algorithms are three types of strategies that can be used to optimize the hyper parameters of your model. Following the optimization process, a new scheduler is developed, which is referred to as the optimized schedule. Finally, all optimized schedule jobs are queued up in the main queue for processing.

5.4. Job execution on machine

Whenever the main process is complete, each subsequent job is prepared for execution. First a job is received by a cloud broker, the broker verifies the storage of the datacenter from the cloud information service provider (CIS). The cloud-based information storage (CIS) is a type of storage that stores all of the information about the datacenter storage that is available in the public cloud. When a datacenter is established, it must first be registered with the Central Information System (CIS). When the cloud broker receives the information from CIS, the job is assigned to a specific datacenter location. The datacenter is comprised of hosts and virtual machines. The host and virtual machine both include parameters such as bandwidth, storage, RAM, the number of processing elements, the cost of processing, and so on. Eventually, a job is executed on the machine by the operator.

6. Proposed algorithm for priority based job scheduling

Job scheduling in cloud computing is one of the most complex fundamental challenges to solve and the most difficult to implement. Job scheduling's major goal is to accomplish high-performance computing while maintaining the best potential system throughput Foster et al. (2008). When it comes to cloud computing, traditional task scheduling algorithms for example First Come Shortest Job First (SJF) Cobham (1954), First Serve (FCFS), Shortest Remaining Job First (SRJF) Kleinrock (1964) and Round Robin (RR) Coffman Jr and Kleinrock (1968) are not acceptable methods for scheduling because of the huge number of concurrent users. This is because just a few scheduling metrics can be satisfied by basic work scheduling algorithms Ghanbari (2019). With this reason, cloud computing treats users' requirements as a large-scale resource that may be allocated to a large number of operations in order to improve efficiency. As a result, a good cloud-based job scheduling system must meet a number of different requirements. Based on all of the efficiency-related scheduling criteria, we designed a scheduling technique for fair job distribution in the central processing unit (CPU).

In cloud computing, fair and optimized scheduling is an important issue. We proposed an algorithm based on two considerations. All jobs are sorted into two categories based on priority parameters in the first step. Priority is most important because when certain emergent requests like medical services, fire services, etc. come to get services, using the existing algorithms, the request has to wait a long time. To solve these problems, we've come up with a new algorithm that will classify all jobs as either priority or nonpriority. In the second step, we concentrated on optimization. The service provider can divide the jobs using any number, denoted by n, based on the availability of work in both queues (priority and nonpriority). Non-priority jobs will be divided into twice as much as priority jobs. Because, for fair scheduling, our initial goal is to give priority to jobs in the priority queue. Finally, one group of jobs will arrive from the priority queue, while another group will come from the non-priority queue in the final queue. For example, if there are 8 jobs available in both the priority and non-priority queues and the service provider considers n = 2, the priority queue will be divided by 2 and the non-priority queue will be divided by 4. Finally, in the final queue, 4 jobs will come from the priority queue and 2 jobs will arrive from the nonpriority queue.

Fig. 5 depicts a general structure for the proposed technique, which provides a starting point for further discussion. The proposed algorithm is divided into six steps, which are as follows:

Step 1: All submitted jobs will be stored in the initial Queue.

Step 2: The job type will be checked by the search algorithm. If the task is an emergency, such as medical treatment, fair service, or government notification, it will be labeled as a priority job; otherwise, it will be marked as a non-priority job. Priority jobs have a number of 1, while non-priority jobs have a number of 0.

Step 3: Check the priority number to make sure it is correct. If the number is 0, the job will be placed in the nonpriority queue, and if the number is 1, the job will be checked again to see if it is longer than the average job length. If the job length is more than the average job length, it will be placed in the non-priority queue; otherwise, it will be placed in the priority queue.

$$Average\ job\ length = \frac{Total\ length\ of\ submitted\ job}{Total\ number\ of\ job} \quad (1)$$

Step 4: If the non-priority job length is two times greater compared to the priority Queue, then this queue will be divided into

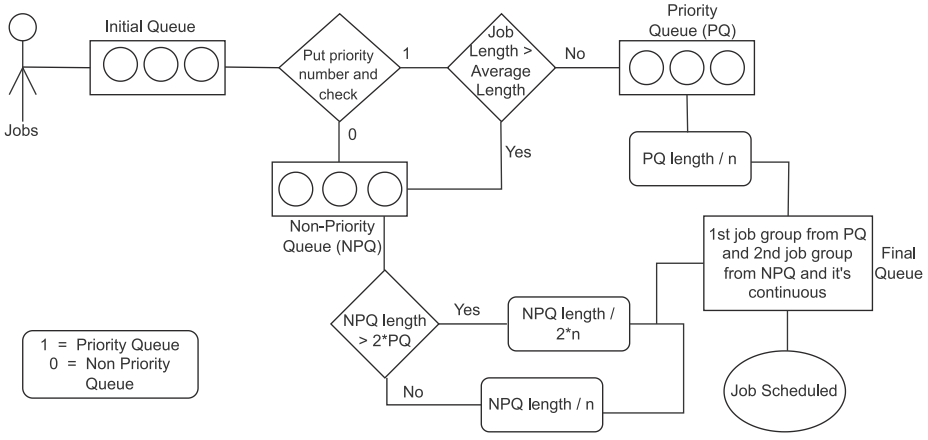

Fig. 5. Diagram of proposed algorithm for priority-based job scheduling.

2*n otherwise this queue will be divided into n. Here n is a real number what will be select by the service provider.

$$Group\ of\ NPQ = \frac{Total\ number\ of\ non\text{-}priority\ jobs}{n\ or\ 2n} \quad (2)$$

Step 5: All the priority queue jobs will be divided by 3 groups.

$$Group\ of\ PQ = \frac{Total\ number\ of\ priority\ jobs}{n} \quad (3)$$

Step 6: In the final queue, the first group will come from the priority queue and the 2nd group will from the nonpriority queue and it will continue until having any group in the priority queue and non-priority queue.

There are no rigorous programming language syntaxes or technical considerations when writing pseudocode, which is an informal way to describe programming. It's a tool for sketching out a program's general structure. The flow of a program is summarized in pseudocode, but the underlying details are left out. Algorithm 1 depicts the pseudocode of proposed algorithm.

7. Open research issues

Cloud computing has made significant progress in the creation of scalable computer infrastructures based on a pay-per-use model. To begin, we will go over several open research issue in cloud computing that have yet to be solved. Resources management techniques have been presented in the past to enhance key performance indicators, but it's still in its early stages because of some main issues such as heterogeneity, diversity and complexity of applications, varying costs and energy consumption, unpredictable end user demands, virtualization, reliability, and scalability as research directions in new approaches to balancing end-user demand and energy efficiency have not yet been developed by researchers.

1. Cloud resource scheduling is one of the most difficult problems to address, as it involves concerns like heterogeneity, uncertainty, and resource dispersion that cannot be solved using standard resource management methods. Cloud applications and services will be more reliable if certain cloud features are prioritized, thus they should be given a lot of attention. The goal of resource scheduling is to ensure that resources are used effectively and efficiently by the applications by matching the correct workloads to the appropriate resources at the appropriate time.
2. Green computing is a difficult environmental issue. Because green computing environment maintenance is believed to be highly complex, it can be both time consuming and costly. This is because the technology behind green IT is very recent and continuously evolving, necessitating significant maintenance efforts. As a result, considerable research into energy-based resource scheduling is required.
3. Virtualization can generate significant revenues by enabling the transfer of Virtual Machines (VMs) to stable workloads around the datacenter. Furthermore, VM migration makes it possible to set up a new data center quickly and robustly. Researchers found that migrating a full operating system and most of its jobs as a single unit avoid many of the problems that come with process-level migration methods. They also looked into the benefits of VM migration. You can't move quickly enough when you find hotspots in the workload and start the migration.
4. Resources can be added or removed dynamically, depending on the needs of the system. The major purpose of a dynamic autonomous resource management procedure in a data center is to avoid squandering resources due to underutilization. A breach in the service level agreement (SLA) between the client and provider might come from excessively long response times caused by over-utilization.
5. Since only a few scheduling algorithms have taken these considerations into account, future research must include failure handling and job migration feathers to increase system efficiency.

Algorithm 1
Priority Based Job Scheduling Technique

```
JobType = Total number job request types;
EmergencyType = Total number of the emergency
  job set by service provider;
for i = 0 to JobType do
    for j = 0 EmergencyType do
        if EmergencyType = JobType then
            priority number 1;
        end
        non-priority number 0
    end
end
JobNumber = Total number of jobs in the initial
  queue;
for i = 0 to JobNumber do
    if prioritynumber1 and JobLength <
      AverageLength then
        store job in priority queue;
    end
    store in non-priority queue;
end
PriorityCount = 0;
for i = PriorityCount to priority queue length / n do
    NewPriorityQueue = PriorityQueue;
    PriorityCount = priority queue length / n;
end
NonPriorityCount = 1;
if Jobnumberof non − priorityqueue > 2 *
  jobnumberpriorityqueue then
    non-priority queue length = non-priority queue
      length / 2*n;
    for i = NonPriorityCount to non-priority queue
      length do
        NewNonPriorityQueue =
          NonPriorityQueue;
        NonPriorityCount = non-priority queue
          length / 2*n;
    end
end
if Jobnumberof non − priorityqueue < 2 *
  jobnumberpriorityqueue then
    NPQ length = NPQ length / n;
end
for i = 0 to NewNonPriorityQueue length plus
  NewPriorityQueue do
    FinalQueue = 1st job group from priory queue
      and 2nd from non-priority;
end
```

6. Techniques such as ML and deep learning (DL) can be utilized to failure vaticination in the context of dependability as a service. Aside from that, those techniques are being utilized to improve the performance of certain job scheduling factors for example the makespan time, the execution time, and so on.
7. Another outstanding topic in cloud computing is data security. Data center physical security systems can't be accessed by cloud providers, so cloud providers must rely on the infrastructure provider for complete data security. It is impossible for the cloud provider to know whether or not a security configuration is fully implemented in a virtual private cloud. At each architectural level of the cloud, it is dangerous to establish trust mechanisms.
8. Autonomic cloud infrastructures are needed to meet the SLA needs of the cloud user and to minimize the amount of interaction between the cloud consumer and the computing environment. Therefore, the development of an early detection approach for SLA violations is a research problem that can prevent performance decline.
9. To ensure success, critical operations such as defining a threshold value, relocating virtual machines (VMs), monitoring CPU consumption, job migration, and others have to be done in a controlled manner.
10. The service provider allocates the number of resources necessary through a cloud service to meet the quality of service (QoS) standards. As a result, an SLA paradigm is developed to identify SLA infractions on a regular basis, which in turn determines whether compensation or a penalty should be issued. Consequently, service providers must dynamically offer enough resources in order to avoid or even mitigate SLA violations.
11. In the cloud context, it is difficult to estimate forthcoming workloads; as a result, more effective workload prediction approaches (such as those based on machine learning) must be created.

8. Conclusion

This study covers a review of cloud computing resource provisioning and task scheduling algorithms, as well as a taxonomic review of the methods. By applying scheduling algorithms to select the most appropriate resource, job scheduling is designed to improve key performance determinative features for example response time, makespan time, flow time, finish time, cost, and resource utilization. We investigated a few state-of-the-art scheduling techniques and classified them according to the problem they were designed to solve. In this research, we've covered the essential concepts and advantages of existing resource provisioning approaches, as well as the classification of scheduling algorithms into static and dynamic categories. Additional important scheduling algorithms are discussed, including those that are depand on the heuristic and meta-heuristic approaches as well as hybrid and training-based approaches, among others. In this section, we explore all of the algorithms stated above in terms of QoS parameters, nature of tasks, problem solving approaches, advantages of the used algorithm, limits of the used algorithm, and simulation tool, all while keeping a deadline as a constraint. We have discovered that the majority of the algorithms do not take into account several critical QoS factors, limitations, and SLA violations, among other things. Maximum algorithms contain various constraints that cause the algorithm's performance to decline over time. When it comes to cloud computing, a JST framework is a comprehensive solution that is supplied by the merging of numerous different approaches in this area of study. The recommended framework is intended to improve the performance of effective job scheduling while simultaneously lowering the computational cost of the process. In cloud computing, the most important thing to remember is that job scheduling must be fair. In this paper, we offer a priority-based work scheduling technique for cloud computing, which will determine which jobs will produce the greatest results. For this study, a number of issues were investigated in order to assess the initiatives of various modern methodologies, and several relevant guidelines. It is expected that our proposed method will serve as a foundation for subsequent investigation into successful task scheduling in more depth.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abd Elaziz, M., Xiong, S., Jayasena, K., Li, L., 2019. Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution. Knowl.-Based Syst. 169, 39–52.

Abdelmaboud, A., Jawawi, D.N., Ghani, I., Elsafi, A., Kitchenham, B., 2015. Quality of service approaches in cloud computing: A systematic mapping study. J. Syst. Softw. 101, 159–179.

Abedi, M., Pourkiani, M., 2020. Resource allocation in combined fog-cloud scenarios by using artificial intelligence. In: 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC). IEEE, pp. 218–222.

Adhikari, M., Nandy, S., Amgoth, T., 2019. Meta heuristic-based task deployment mechanism for load balancing in iaas cloud. J. Netw. Comput. Appl. 128, 64–77.

Al-maamari, A., Omara, F.A., 2015. Task scheduling using pso algorithm in cloud computing environments. Int. J. Grid Distrib. Comput. 8 (5), 245–256.

Alemnesh, G., 2020. Time Optimized Hybrid Scheduling Algorithm for Cloud Computing Environment. Ph.D. thesis. ASTU.

Alhaidari, F., Balharith, T.Z., 2021. Enhanced round-robin algorithm in the cloud computing environment for optimal task scheduling. Computers 10, 63.

Ali, J., Zafari, F., Khan, G.M., Mahmud, S.A., 2013. Future clients' requests estimation for dynamic resource allocation in cloud data center using cgpann. In: 2013 12th International Conference on Machine Learning and Applications. IEEE, pp. 331–334.

Alkayal, E.S., Jennings, N.R., Abulkhair, M.F., 2016. Efficienttaskscheduling multi-objective particle swarm optimization in cloud computing. In: 2016 IEEE 41st Conference on Local Computer Networks Workshops (LCN Workshops). IEEE, pp. 17–24.

Alkhateeb, F., Abed-alguni, B.H., Al-rousan, M.H., 2021. Discrete hybrid cuckoo search and simulated annealing algorithm for solving the job shop scheduling problem. J. Supercomput. 78 (4), 4799–4826.

Allahverdi, A., 2015. The third comprehensive survey on scheduling problems with setup times/costs. Eur. J. Oper. Res. 246 (2), 345–378.

Allahverdi, A., Ng, C.T., Cheng, T.C.E., Kovalyov, M.Y., 2008. A survey of scheduling problems with setup times or costs. Eur. J. Oper. Res. 187 (3), 985–1032.

Alworafi, M.A., Dhari, A., El-Booz, S.A., Nasr, A.A., Arpitha, A., Mallappa, S., 2019. An enhanced task scheduling in cloud computing based on hybrid approach. In: Data Analytics and Learning. Springer, pp. 11–25.

Ananth, A., Chandrasekaran, K., 2015. Cooperative game theoretic approach for job scheduling in cloud computing. In: 2015 International Conference on Computing and Network Communications (CoCoNet). IEEE, pp. 147–156.

Ardagna, D., Casale, G., Ciavotta, M., Pérez, J.F., Wang, W., 2014. Qualityof-service in cloud computing: modeling techniques and their applications. J. Internet Serv. Appl. 5, 1–17.

Aslam, S., Shah, M.A., 2015. Load balancing algorithms in cloud computing: A survey of modern techniques. In: 2015 National software engineering conference (NSEC). IEEE, pp. 30–35.

Bagheri, M.H., Bagherizadeh, M., Moradi, M., Moaiyeri, M.H., 2021. Design of cntfet-based current-mode multi-input m: 3 (4 m 7) counters. IETE J. Res. 67, 322–332.

Belgacem, A., Beghdad-Bey, K., 2021. Multi-objective workflow scheduling in cloud computing: trade-off between makespan and cost. Clust. Comput. 25 (1), 579–595.

Benny, R., Wirawan, I., 2022. Comparison analysis of round robin algorithm with highest response ratio next algorithm for job scheduling problems. Int. J. Open Inf. Technol. 10, 21–26.

Bezdan, T., Zivkovic, M., Bacanin, N., Strumberger, I., Tuba, E., Tuba, M., 2022. Multi-objective task scheduling in cloud computing environment by hybridized bat algorithm. J. Intell. Fuzzy Syst. 42 (1), 411–423.

Chen, X., Long, D., 2019. Task scheduling of cloud computing using integrated particle swarm algorithm and ant colony algorithm. Clust. Comput. 22 (S2), 2761–2769.

Cheng, F., Huang, Y., Tanpure, B., Sawalani, P., Cheng, L., Liu, C., 2022. Cost-aware job scheduling for cloud instances using deep reinforcement learning. Clust. Comput. 25 (1), 619–631.

Chien, W.-C., Lai, C.-F., Chao, H.-C., 2019. Dynamic resource prediction and allocation in c-ran with edge artificial intelligence. IEEE Trans. Ind. Inf. 15 (7), 4306–4314.

Cobham, A., 1954. Priority assignment in waiting line problems. J. Oper. Res. Soc. Am. 2 (1), 70–76.

Coffman Jr, E.G., Kleinrock, L., 1968. Computer scheduling methods and their countermeasures, in: Proceedings of the April 30–May 2, 1968, Spring Joint Computer Conference, pp. 11–21.
Cui, D., Peng, Z., Lin, W., et al., 2017. A reinforcement learning-based mixed job scheduler scheme for grid or iaas cloud. IEEE Trans. Cloud Comput.
Dabbagh, M., Hamdaoui, B., Guizani, M., Rayes, A., 2015. Energy-efficient resource allocation and provisioning framework for cloud data centers. IEEE Trans. Netw. Serv. Manage. 12 (3), 377–391.
Dashti, S.E., Rahmani, A.M., 2016. Dynamic vms placement for energy efficiency by pso in cloud computing. J. Exp. Theor. Artif. Intell. 28 (1-2), 97–112.
Deol, G.J.S. et al., 2021. Hadoop job scheduling using improvised ant colony optimization. Turk. J. Comput. Math. Educ. (TURCOMAT) 12, 3417–3424.
Dubey, K., Kumar, M., Sharma, S., 2018. Modified heft algorithm for task scheduling in cloud environment. Procedia Comput. Sci. 125, 725–732.
Dubey, K., Sharma, S., 2021. A novel multi-objective cr-pso task scheduling algorithm with deadline constraint in cloud computing. Sustainable Comput. Inf. Syst. 32, 100605.
Ebadifard, F., Babamir, S.M., 2018. A pso-based task scheduling algorithm improved using a load-balancing technique for the cloud computing environment. Concurr. Comput.: Pract. Exp. 30, e4368.
Eldesokey, H.M., Abd El-atty, S.M., El-Shafai, W., Amoon, M., Abd El-Samie, F.E., 2021. Hybrid swarm optimization algorithm based on task scheduling in a cloud environment. Int. J. Commun. Syst. 34, e4694.
Endo, P.T., de Almeida Palhares, A.V., Pereira, N.N., Goncalves, G.E., Sadok, D., Kelner, J., Melander, B., Mangs, J.E., 2011. Resource allocation for distributed cloud: concepts and research challenges. IEEE Network 25, 42–46.
Farooq, M.U., Shakoor, A., Siddique, A.B., 2017. An efficient dynamic round robin algorithm for cpu scheduling. In: 2017 International Conference on Communication, Computing and Digital Systems (C-CODE). IEEE, pp. 244–248.
Foster, I., Zhao, Y., Raicu, I., Lu, S., 2008. Cloud computing and grid computing 360-degree compared. In: 2008 grid computing environments workshop. IEEE, pp. 1–10.
Gao, Y., Huang, C., 2021. Energy-efficient scheduling of mapreduce tasks based on load balancing and deadline constraint in heterogeneous hadoop yarn cluster. In: 2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD). IEEE, pp. 220–225.
Gąsior, J., Seredyński, F., 2019. Security-aware distributed job scheduling in cloud computing systems: a game-theoretic cellular automatabased approach. In: International Conference on Computational Science. Springer, pp. 449–462.
Geetha, P., Robin, C., 2021. Power conserving resource allocation scheme with improved qos to promote green cloud computing. J. Ambient Intell. Hum. Comput. 12, 7153–7164.
Geetha, R., Parthasarathy, V., 2021. Anadvancedartificialintelligencetechnique for resource allocation by investigating and scheduling paralleldistributed request/response handling. J. Ambient Intell. Hum. Comput. 12, 6899–6909.
Ghanbari, S., 2019. Priority-aware job scheduling algorithm in cloud computing: A multi-criteria approach. Azerbaijan J. High Perform. Comput. 2, 29–38.
Ghanbari, S., Othman, M., 2012. A priority based job scheduling algorithm in cloud computing. Procedia Eng. 50, 778–785.
Gharbia, R., El Baz, A.H., Hassanien, A.E., Tolba, M.F., 2014. Remote sensing image fusion approach based on brovey and wavelets transforms, in: Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA 2014, Springer. pp. 311–321.
Ghomi, E.J., Rahmani, A.M., Qader, N.N., 2017. Load-balancing algorithms in cloud computing: A survey. J. Netw. Comput. Appl. 88, 50–71.
Gomathi, B., Krishnasamy, K., Balaji, B.S., 2018. Epsilon-fuzzy dominance sort-based composite discrete artificial bee colony optimisation for multi-objective cloud task scheduling problem. Int. J. Bus. Intell. Data Min. 13, 247–266.
Gond, S., Singh, S., 2018. Load balancing in cloud computing: A survey on comparison of two algorithms pso and sjf-mmbf. In: 2018 8th International Conference on Communication Systems and Network Technologies (CSNT). IEEE, pp. 62–66.
Goswami, S., De Sarkar, A., 2013. A comparative study of load balancing algorithms in computational grid environment. In: 2013 Fifth International Conference on Computational Intelligence, Modelling and Simulation. IEEE, pp. 99–104.
Goutam, S., Yadav, A.K., 2015. Preemptable priority based dynamic resource allocation in cloud computing with fault tolerance. In: 2015 International Conference on Communication Networks (ICCN). IEEE, pp. 278–285.
Goyal, K., Jain, V., Chauhan, S., 2020. Relating job scheduling algorithms on job lengths and number of cloudlets in cloud computing .
Gu, Y., Tao, J., Wu, X., Ma, X., 2017. Online mechanism with latestreservation for dynamic vms allocation in private cloud. Int. J. Syst. Assurance Eng. Manage. 8, 2009–2016.
Hameed, A., Khoshkbarforoushha, A., Ranjan, R., Jayaraman, P.P., Kolodziej, J., Balaji, P., Zeadally, S., Malluhi, Q.M., Tziritas, N., Vishnu, A., et al., 2016. A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems. Computing 98, 751–774.
Hassan, M.A., Kacem, I., Martin, S., Osman, I.M., 2015. Genetic algorithms for job scheduling in cloud computing. Stud. Inf. Control 24, 387–400.
Holladay, K., Pickens, K., Miller, G., 2017. The effect of evaluation time variance on asynchronous particle swarm optimization. In: 2017 IEEE Congress on Evolutionary Computation (CEC). IEEE, pp. 161–168.
Horri, A., Mozafari, M.S., Dastghaibyfard, G., 2014. Novel resource allocation algorithms to performance and energy efficiency in cloud computing. J. Supercomput. 69, 1445–1461.
Houssein, E.H., Gad, A.G., Wazery, Y.M., Suganthan, P.N., 2021. Task scheduling in cloud computing based on meta-heuristics: Review, taxonomy, open challenges, and future trends. Swarm Evol. Comput. 100841
Hu, W.X., Zheng, J., Hua, X.Y., Yang, Y.Q., 2013. A computing capability allocation algorithm for cloud computing environment. In: Applied Mechanics and Materials. Trans Tech Publ., pp. 2400–2406.
Ibnyaich, S., Wakrim, L., Hassani, M.M., 2021. Nonuniform semi-patches for designing an ultra wideband pifa antenna by using genetic algorithm optimization. Wireless Pers. Commun. 117, 957–969.
Ilyushkin, A., Epema, D., 2018. The impact of task runtime estimate accuracy on scheduling workloads of workflows. In: 2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID). IEEE, pp. 331–341.
Jain, A., Gupta, R., 2015. Gaussian filter threshold modulation for filtering flat and texture area of an image. In: 2015 International Conference on Advances in Computer Engineering and Applications, pp. 760–763.
Jain, R., Sharma, N., 2022. A deadline-constrained time-cost-effective salp swarm algorithm for resource optimization in cloud computing. Int. J. Appl. Metaheuristic Comput. (IJAMC) 13, 1–21.
Javadi, B., Abawajy, J., Sinnott, R.O., 2012. Hybrid cloud resource provisioning policy in the presence of resource failures. In: 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings. IEEE, pp. 10–17.
Javanmardi, S., Shojafar, M., Amendola, D., Cordeschi, N., Liu, H., Abraham, A., 2014. Hybrid job scheduling algorithm for cloud computing environment, in: Proceedings of the fifth international conference on innovations in bio-inspired computing and applications IBICA 2014, Springer. pp. 43–52.
Jayanthi, S., 2014. Literature review: Dynamic resource allocation mechanism in cloud computing environment. In: 2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE). IEEE, pp. 279–281.
Jena, R., 2017. Task scheduling in cloud environment: A multi-objective abc framework. J. Inf. Optim. Sci. 38, 1–19.
Katyal, M., Mishra, A., 2014. Application of selective algorithm for effective resource provisioning in cloud computing environment. arXiv preprint arXiv:1403.2914 .
Kaur, A., Kaur, B., Singh, D., 2017. Challenges to task and workflow scheduling in cloud environment. In: International Journal of Advanced Research in Computer Science, p. 8.
Khalili, A., Babamir, S.M., 2015. Makespan improvement of pso-based dynamic scheduling in cloud environment. In: 2015 23rd Iranian Conference on Electrical Engineering, pp. 613–618.
Khan, U.A., Khalid, W., Saifullah, S., 2020. Energy efficient resource allocation and computation offloading strategy in a uav-enabled secure edge-cloud computing system. Proceedings - 2020 IEEE International Conference on Smart Internet of Things, SmartIoT 2020, 58–63doi:10. 1109/SmartIoT49966.2020.00018.
Kiruthiga, R., Akila, D., 2021. Prediction-based cost-efficient resource allocation scheme for big data streams in cloud systems. In: Proceedings of First International Conference on Mathematical Modeling and Computational Science. Springer, pp. 233–242.
Kleinrock, L., 1964. A time-shared processor. Naval Research Logistics Quarterly, Version of Record online: 1 AUG 2006 11, 59–73.
Kodli, S., Terdal, S., 2021. Hybrid max-min genetic algorithm for load balancing and task scheduling in cloud environment. Int J Intell Eng Syst. 14, 63–71.
Kopanski, J., Rzadca, K., 2021. Plan-based job scheduling for supercomputers with shared burst buffers. In: European Conference on Parallel Processing. Springer, pp. 120–135.
Kumar, A.S., Venkatesan, M., 2019. Multi-objective task scheduling using hybrid genetic-ant colony optimization algorithm in cloud environment. Wireless Pers. Commun. 107, 1835–1848.
Kumar, E.M., 2018. Cloud computing in resource management. Int. J. Eng. Manage. Res. (IJEMR) 8, 93–98.
Kumar, M., Sharma, S.C., 2018. Pso-cogent: Cost and energy efficient scheduling in cloud environment with deadline constraint. Sustainable Comput. Inf. Syst. 19, 147–164.
Kumar, M., Sharma, S.C., Goel, A., Singh, S.P., 2019. A comprehensive survey for scheduling techniques in cloud computing. J. Netw. Comput. Appl. 143, 1–33.
Kumar, N., Saxena, S., 2015. A preference-based resource allocation in cloud computing systems. Procedia Comput. Sci. 57, 104–111.
Lee, C.Y., 1996. Scheduling: Theory, algorithms, and systems [book review].
Lee, H.M., Jeong, Y.S., Jang, H.J., 2014. Performance analysis based resource allocation for green cloud computing. J. Supercomput. 69, 1013–1026.
Li, C., Li, L., 2013. Efficient resource allocation for optimizing objectives of cloud users, iaas provider and saas provider in cloud environment. J. Supercomput. 65, 866–885.
Li, D., Wu, J., 2014. Minimizing energy consumption for frame-based tasks on heterogeneous multiprocessor platforms. IEEE Trans. Parallel Distrib. Syst. 26, 810–823.
Li, F., Hu, B., 2019. Deepjs: Job scheduling based on deep reinforcement learning in cloud data center. In: Proceedings of the 2019 4th international conference on big data and computing, pp. 48–53.
Liu, L., Mei, H., Xie, B., 2016. Towards a multi-qos human-centric cloud computing load balance resource allocation method. J. Supercomput. 72, 2488–2501.
Ma, X., Gao, H., Xu, H., Bian, M., 2019. An iot-based task scheduling optimization scheme considering the deadline and cost-aware scientific workflow for cloud computing. EURASIP J. Wireless Commun. Network. 2019, 1–19.
Manasrah, A.M., Ba Ali, H., 2018. Workflow scheduling using hybrid ga-pso algorithm in cloud computing. Wireless Commun. Mobile Comput. 2018.
Mansouri, N., Javidi, M.M., 2020. Cost-based job scheduling strategy in cloud computing environments. Distrib. Parallel Databases 38, 365–400.
Manzoor, M.F., Abid, A., Farooq, M.S., Nawaz, N.A., Farooq, U., 2020. Resource allocation techniques in cloud computing: A review and future directions. Elektronika ir Elektrotechnika 26, 40–51.
Meena, J., Kumar, M., Vardhan, M., 2016. Cost effective genetic algorithm for workflow scheduling in cloud under deadline constraint. IEEE Access 4, 5065–5082.
Mehta, H., Prasad, V.K., Bhavsar, M., 2017. Efficient resource scheduling in cloud computing. Int. J. Adv. Res. Comput. Sci. 8, 809–815.
Milani, A.S., Navimipour, N.J., 2016. Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends. J. Network Comput. Appl. 71, 86–98.

Mohamaddiah, M.H., Abdullah, A., Subramaniam, S., Hussin, M., 2014. A survey on resource allocation and monitoring in cloud computing. Int. J. Mach. Learn. Comput. 4, 31–38.
Mohana, R., 2015. Apositionbalancedparallelparticleswarmoptimization method for resource allocation in cloud. Indian J. Sci. Technol. 8, 182–188.
Mondal, R.K., Nandi, E., Sarddar, D., 2015. Load balancing scheduling with shortest load first. Int. J. Grid Distrib. Comput. 8, 171–178.
Mousavi, S., Mosavi, A., Varkonyi-Koczy, A.R., Fazekas, G., 2017. Dynamic resource allocation in cloud computing. Acta Polytech. Hung. 14, 83–104.
Mousavinasab, Z., Entezari-Maleki, R., Movaghar, A., 2011. A bee colony task scheduling algorithm in computational grids. In: International Conference on Digital Information Processing and Communications. Springer, pp. 200–210.
Murad, S.A., Azmi, Z.R.M., Muzahid, A.J.M., Al-Imran, M., 2021. Comparative study on job scheduling using priority rule and machine learning. In: 2021 Emerging Technology in Computing, Communication and Electronics (ETCCE). IEEE, pp. 1–8.
Navimipour, N.J., 2015. Task scheduling in the cloud environments based on an artificial bee colony algorithm. Int. Conf. Image Process., 38–44
Nazir, S., Shafiq, S., Iqbal, Z., Zeeshan, M., Tariq, S., Javaid, N., 2018. Cuckoo optimization algorithm based job scheduling using cloud and fog computing in smart grid. In: International Conference on Intelligent Networking and Collaborative Systems. Springer, pp. 34–46.
Nguyen, T., Nguyen, T., Vu, Q.H., Huynh, T.T.B., Nguyen, B.M., 2021. Multi-objective sparrow search optimization for task scheduling in fogcloud-blockchain systems. In: 2021 IEEE International Conference on Services Computing (SCC). IEEE, pp. 450–455.
Oddi, G., Panfili, M., Pietrabissa, A., Zuccaro, L., Suraci, V., 2013. A resource allocation algorithm of multi-cloud resources based on markov decision process. In: 2013 IEEE 5th International Conference on Cloud Computing Technology and Science. IEEE, pp. 130–135.
Pandi, K.M., Somasundaram, K., 2016. Energy efficient in virtual infrastructure and green cloud computing: A review. Indian J. Sci. Technol. 9, 1–8.
Panetta, C., Menk, J., Jonk, Y., Brown, A., Powers, M., Shapiro, A., 2010. Prospective randomized clinical trial evaluating the impact of vinegar on high density lipoprotein. J. Am. Diet. Assoc. 110, A87.
Pang, S., Li, W., He, H., Shan, Z., Wang, X., 2019. An eda-ga hybrid algorithm for multi-objective task scheduling in cloud computing. IEEE Access 7, 146379–146389.
Papagianni, C., Leivadeas, A., Papavassiliou, S., Maglaris, V., Cervello, C., 2013. On the optimal allocation of virtual resources in cloud computing. Networks 62, 1060–1071.
Parikh, S.M., Patel, N.M., Prajapati, H.B., 2017. Resource management in cloud computing: classification and taxonomy. arXiv preprint arXiv:1703.00374 .
Patel, K., Thakkar, A., Shah, C., Makvana, K., 2016. A state of art surveyonshillingattackincollaborativefilteringbasedrecommendation system, in: Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 1, Springer. pp. 377–385.
Patel, R., Dahiya, D., 2015. Aggregation of cloud providers: a review of opportunities and challenges. Int. Conf. Comput. Commun. Autom. IEEE, 620–626.
Patel, S., Bhoi, U., 2013. Priority based job scheduling techniques in cloud computing: a systematic review. Int. J. Sci. Technol. Res. 2, 147–152.
Pillai, P.S., Rao, S., 2014. Resource allocation in cloud computing using the uncertainty principle of game theory. IEEE Syst. J. 10, 637–648.
Pradhan, P., Behera, P.K., Ray, B., 2016. Modified round robin algorithm for resource allocation in cloud computing. Procedia Comput. Sci. 85, 878–890.
Pratap, R., Zaidi, T., 2018. Comparative study of task scheduling algorithms through cloudsim. In: 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). IEEE, pp. 397–400.
Praveenchandar, J., Tamilarasi, A., 2021. Dynamic resource allocation with optimized task scheduling and improved power management in cloud computing. J. Ambient Intell. Hum. Comput. 12, 4147–4159.
Pu, S., Escudero-Garzás, J.J., Garcia, A., Shahrampour, S., 2020. An online mechanism for resource allocation in networks. IEEE Trans. Control Network Syst. 7, 1140–1150.
Raghava, N., Singh, D., 2014. Comparative study on load balancing techniques in cloud computing. Open J. Mobile Comput. Cloud Comput. 1, 18–25.
Raju, R., Babukarthik, R., Chandramohan, D., Dhavachelvan, P., Vengattaraman, T., 2013. Minimizing the makespan using hybrid algorithm for cloud computing. In: 2013 3rd IEEE International Advance Computing Conference (IACC). IEEE, pp. 957–962.
Ramezani, F., Lu, J., Hussain, F.K., 2014. Task-basedsystemloadbalancing in cloud computing using particle swarm optimization. Int. J. Parallel Prog. 42, 739–754.
Randles, M., Lamb, D., Taleb-Bendiab, A., 2010. A comparative study into distributed load balancing algorithms for cloud computing. In: 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops. IEEE, pp. 551–556.
Ravichandran, P., Krishnamurthy, K., Parameshwaran, R., 2016. A hybrid pso-cs algorithm for parallel line job shop scheduling to minimize makespan. World Appl. Sci. J. 34, 878–883.
Rezvani, M., Akbari, M.K., Javadi, B., 2015. Resource allocation in cloud computing environments based on integer linear programming. The Computer Journal 58, 300–314.
Rjoub, G., Bentahar, J., 2017. Cloud task scheduling based on swarm intelligence and machine learning. In: 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud). IEEE, pp. 272–279.
Rjoub, G., Bentahar, J., Abdel Wahab, O., Saleh Bataineh, A., 2020. Deep and reinforcement learning for automated task scheduling in large-scale cloud computing systems. Concurr. Comput.: Pract. Exp., e5919
Samriya, J.K., Kumar, N., 2022. Spider monkey optimization based energyefficient resource allocation in cloud environment. Trends Sci. 19, 1710–1710.
Saraswathi, A., Kalaashri, Y.R., Padmavathi, S., 2015. Dynamic resource allocation scheme in cloud computing. Procedia Comput. Sci. 47, 30–36.
Sels, V., Gheysen, N., Vanhoucke, M., 2012. A comparison of priority rules for the job shop scheduling problem under different flow time-and tardiness-related objective functions. Int. J. Prod. Res. 50, 4255–4270.
Selvi, S.T., Valliyammai, C., Dhatchayani, V.N., 2014. Resource allocation issues and challenges in cloud computing. In: 2014 International Conference on Recent Trends in Information Technology. IEEE, pp. 1–6.
Shang, Q., 2021. A dynamic resource allocation algorithm in cloud computing based on workflow and resource clustering. J. Internet Technol. 22, 403–411.
Singh, P., 2021. Scheduling tasks based on branch and bound algorithm in cloud computing environment. In: 2021 8th International Conference on Signal Processing and Integrated Networks (SPIN). IEEE, pp. 41–46.
Singh, S., 2015. Green computing strategies & challenges. In: 2015 International Conference on Green Computing and Internet of Things (ICGCIoT). IEEE, pp. 758–760.
Singh, S., Chana, I., 2016. Resource provisioning and scheduling in clouds: Qos perspective. J Supercomput. 72, 926–960.
Stryer, P., 2010. Understanding data centers and cloud computing, 1–7.
Surendran, R., Tamilvizhi, T., 2018. Howtoimprovetheresourceutilization in cloud data center? In: 2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT). IEEE, pp. 1–6.
Taillard, E., 1990. Some efficient heuristic methods for the flow shop sequencing problem. Eur. J. Oper. Res. 47, 65–74.
Tarahomi, M., Izadi, M., Ghobaei-Arani, M., 2021. An efficient poweraware vm allocation mechanism in cloud data centers: a micro geneticbased approach. Clust. Comput. 24, 919–934.
Tchendji, V.K., Myoupo, J.F., Dequen, G., 2016. Deriving cgm basedparallel algorithms for the optimal binary search-tree problem. In: Information Technology: New Generations. Springer, pp. 655–664.
Thakur, A., Goraya, M.S., 2017. A taxonomic survey on load balancing in cloud. J. Netw. Comput. Appl. 98, 43–57.
Tiwari, S.P., Bansal, K.K., Hybrid cs+ aco algorithm for job scheduling. Vakilinia, S., 2018. Energy efficient temporal load aware resource allocation in cloud computing data centers. J. Cloud Comput. 7, 1–24.
Vinothiyalakshmi, P., Anitha, R., 2021. Efficient dynamic resource provisioning based on credibility in cloud computing. Wireless Netw. 27, 2217–2229.
Wang, C.F., Hung, W.Y., Yang, C.S., 2014. A prediction based energy conserving resources allocation scheme for cloud computing. IEEE, pp. 320–324.
Wang, L., von Laszewski, G., Huang, F., Dayal, J., Frulani, T., Fox, G., 2011. Task scheduling with ann-based temperature prediction in a data center: a simulation-based study. Eng. Comput. 27, 381–391.
Wang, Z., Su, X., 2015. Dynamically hierarchical resource-allocation algorithm in cloud computing environment. J. Supercomput. 71, 2748–2766.
Weckman, G.R., Ganduri, C.V., Koonce, D.A., 2008. A neural network job-shop scheduler. J. Intell. Manuf. 19, 191–201.
Weinhardt, C., Anandasivam, A., Blau, B., Borissov, N., Meinl, T., Michalk, W., Stößer, J., 2009. Cloud computing–a classification, business models, and research directions. Bus. Inf. Syst. Eng. 1, 391–399.
Wood, L., Alsawy, S., 2018. Recovery in psychosis from a service user perspective: a systematic review and thematic synthesis of current qualitative evidence. Community Ment. Health J. 54, 793–804.
Xiong, A.P., Xu, C.X., 2014. Energy efficient multiresource allocation of virtual machine based on pso in cloud data center. Math. Probl. Eng.
Xu, X., Fu, S., Cai, Q., Tian, W., Liu, W., Dou, W., Sun, X., Liu, A.X., 2018. Dynamic resource allocation for load balancing in fog environment. Wireless Commun. Mobile Comput.
Xu, X., Yu, H., 2014. A game theory approach to fair and efficient resource allocation in cloud computing. Math. Probl. Eng.
Yao, Y., Cao, J., Li, M., 2013. A network-aware virtual machine allocation in cloud datacenter. In: IFIP International Conference on Network and Parallel Computing. Springer, pp. 71–82.
Yu, M., Yang, B., Chen, Y., 2018. Dynamic integration of process planning and scheduling using a discrete particle swarm optimization algorithm. Adv. Prod. Eng. Manage. 13, 279–296.
Yuvaraj, N., Karthikeyan, T., Praghash, K., 2021. An improved task allocation scheme in serverless computing using gray wolf optimization (gwo) based reinforcement learning (ril) approach. Wireless Pers. Commun. 117, 2403–2421.
Zhang, J., Xie, N., Zhang, X., Yue, K., Li, W., Kumar, D., 2018. Machine learning based resource allocation of cloud computing in auction. Comput. Mater. Continua 56, 123–135.
Zhang, Q., Cheng, L., Boutaba, R., 2010. Cloud computing: state-of-the-art and research challenges. J. Internet Serv. Appl. 1, 7–18.
Zhang, Q., Zhu, Q., Boutaba, R., 2011a. Dynamic resource allocation for spot markets in cloud computing environments. In: 2011 Fourth IEEE International Conference on Utility and Cloud Computing. IEEE, pp. 178–185.
Zhang, Z., Wang, H., Xiao, L., Ruan, L., 2011b. A statistical based resource allocation scheme in cloud. In: 2011 International Conference on Cloud and Service Computing. IEEE, pp. 266–273.
Zheng, J., Wang, Y., 2021. A hybrid multi-objective bat algorithm for solving cloud computing resource scheduling problems. Sustainability 13, 7933.