# A Review of QoS-Driven Task Scheduling Algorithms and Their Impact on Data Quality in Process Management

**Anupam Yadav***
PhD (Pursuing); M.Tech (Computer Science); Assistant Professor; Department of Computer Engineering and Application; GLA University; Mathura, India Email: anupam.yadav@gla.ac.in

**Ashish Sharma**
Professor; Department of Computer Engineering and Application; GLA University; Mathura, India Email: ashish.sharma@gla.ac.in

Iranian Journal of Information Processing and Management

**Received: 15, Apr. 2024** **Accepted: 25, May 2025**

**Abstract:** The term “cloud computing (CC)” has been extensively studied and utilized by major corporations since its inception. Within the realm of cloud computing, various research topics and perspectives have been explored, including resource management, cloud security, and energy efficiency. This paper explores the intersection of data quality and business process management within the context of cloud computing. Specifically, it examines how Quality of Service (QoS)-driven task scheduling algorithms in cloud environments can enhance data quality and optimize business processes. Cloud computing still faces the significant challenge of determining the most effective way to schedule tasks and manage available resources. We need effective scheduling strategies to manage these resources because of the scale and dynamic resource provisioning in modern data centers. The purpose of this work is to provide an overview of the various task scheduling methods that have been utilized in the cloud computing environment to date. An attempt has been made to categorize current methods, investigate issues, and identify important challenges present in this area. Our data reveals that 34% of researchers are focusing on makespan for QoS (Quality of Service) metrics, 17% on cost, 15% on load balancing, 10% on deadline, and 9% on energy usage. Other criteria for the Quality of Service (QoS) parameter contribute far less than the ones

**Iranian Research Institute for Information Science and Technology (IranDoc)**
ISSN 2251-8223
eISSN 2251-8231
Indexed by SCOPUS, ISC, & LISTA
Special Issue | Winter 2025 | pp. 219-254
Exploring the Relationship Between Data Quality and Business Process Management
https://doi.org/10.22034/jipm.2024.2026588.1599

* Corresponding Author

mentioned above. According to this study, scheduling algorithms commonly used by researchers include the genetic algorithm in bio-inspired systems and particle swarm optimization in swarm intelligence 80% of the time. According to the available literature, 70% of the studies have utilized CloudSim as their simulation tool of choice. Our findings suggest that current methodologies mainly employ genetic algorithms and particle swarm optimization, with CloudSim being a popular simulation tool. Ongoing work emphasizes refining scheduling strategies to enhance resource management in dynamic data center environments, providing crucial insights into future quality-of-service (QoS)-driven scheduling algorithms for cloud computing.

**Keywords:** Resource allocation, Meta-heuristic, Cloud computation, Resource scheduling, Optimization techniques, Task scheduling

## 1. Introduction

Cloud computing is a groundbreaking technology that has changed the way businesses and individuals acquire and manage computing resources. This paradigm shift allows for more efficient utilization of computing resources, increased scalability, and improved profitability. As the demand for cloud services increases, ensuring the quality of service (QoS) in cloud computing systems remains a critical issue. As cloud computing continues to evolve, quality of service (QoS) becomes increasingly important. The quality of service is important and encompasses various aspects, including response time, performance, reliability, and resource utilization. To meet customer expectations and adhere to service level agreements, cloud providers must effectively prioritize and manage activities. Task scheduling is an essential procedure in cloud computing, involving the allocation of computational resources to jobs while considering various performance criteria (Lim, Kiah, & Ang, 2017). It is primarily concerned with the utilization and consumption of the resources that are made available in the surrounding environment. Users can submit access requests to the cloud server to utilize the various resources and services available. These cloud resources are not only available for use by a single user at a time but also have the potential to be shared by a large number of customers simultaneously. This plan of sharing available resources among the users can be implemented through the scheduling process (Q. Zhang, Cheng, & Boutaba, 2010). The relevance of Quality of Service

(QoS)-based scheduling approaches in cloud computing can be understood by reviewing the following key considerations.

**Enhancing user experience:** Users anticipate cloud services to be reliable and high-performing. QoS-based scheduling activities can ensure that resources are allocated in a manner that meets Quality of Service (QoS) standards, thereby enhancing customer satisfaction. Cloud providers must manage resource usage to reduce costs and enhance resource availability.

**Ensure compliance with service level agreements:** Many cloud services operate within a framework of service level agreements (SLAs), which specify the expected quality of service (QoS). Violations of these agreements may result in financial penalties. Task scheduling algorithms that prioritize Quality of Service (QoS) are crucial for meeting Service Level Agreements (SLAs) and avoiding penalties.

**Scalable:** Cloud systems exhibit highly dynamic performance, characterized by varying workloads and resource requirements. Efficient task scheduling algorithms can adapt to changing conditions, handle larger workloads, and ensure flexibility.

**Power efficiency** is crucial in modern data centers, with green computing and reducing energy consumption being top priorities. Task scheduling methods can enhance energy efficiency by consolidating workloads on fewer servers and powering down inactive resources.

The cloud scheduler or the server is in charge of dividing up the available resources among the many users. In addition, cloud computing offers consumers the benefit of pay-per-use pricing, which means that users only need to pay for the resources that they really utilize. Cloud tenants typically use various deployment strategies, and the cloud is known to provide a wide range of services. Users have access to three fundamental service types via the usage of cloud computing. These service types are known as Platform-as-a-Service (PaaS), Infrastructure-as-a-Service (IaaS), and Software-as-a-Service (SaaS) (Buyya, Broberg, & Goscinski, 2010) as depicted in Figure 1.

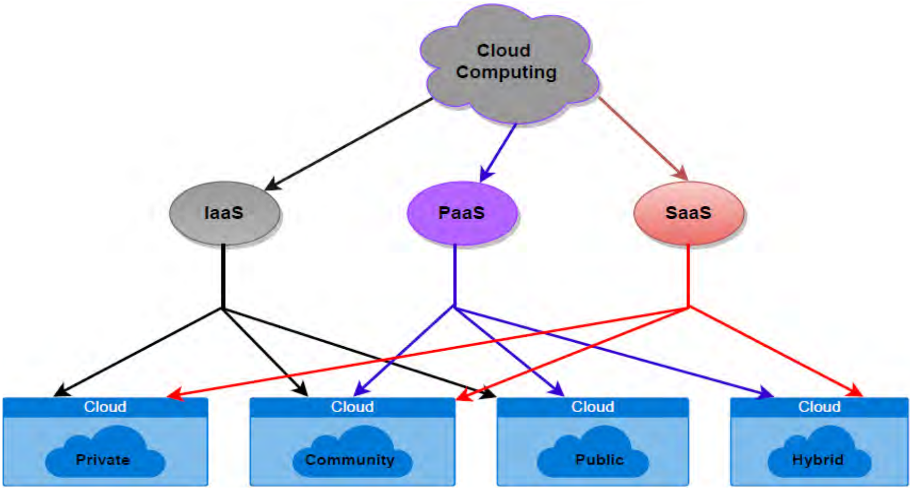


**Figure 1. Models used in cloud computing**

Infrastructure as a Service (IaaS) employs a pay-per-use pricing model to provide customers with a range of computing resources, such as servers, networking, storage, and data center space. Users have the ability to construct, develop, or customize applications when using PaaS. The delivery of software services is accomplished through the use of the SaaS service paradigm. Users can access the software services and pay the subscription using a web browser. Traditional resource scheduling systems deliver services to users based on their needs. The requirements of the individuals utilizing the services could be met by employing some of these strategies. Computing in the cloud can be implemented using a layered architecture and deployed through one of four primary models: private, public, hybrid, and community cloud (Goyal, 2014).

This manuscript serves as a comprehensive resource for academics and network managers interested in current Quality of Service (QoS)-based task scheduling algorithms specifically used in cloud computing.

In this manuscript, instead of examining distributed systems as a whole, we focus on scheduling problems related to cloud computing. The remaining portion of the report is divided into five different parts. In Section 2, we will discuss resource management in cloud computing. The third section covers an extensive survey based on scheduling. Quality of service parameters are covered in Section 4. Section 5 covers task scheduling techniques and related algorithms. Section 6 provides a summary of the findings and offers insights on potential future research directions. An in-depth analysis of the task scheduling process in cloud computing is presented here, emphasizing Quality of Service (QoS) parameters relevant to cloud computing environments. The results of this study will assist academics in determining the most suitable approach for recommending the optimal method for scheduling in cloud environments.

## 2. MANAGING RESOURCES IN CLOUD ENVIRONMENT

The major significant challenge presented by distributed computing, such as cloud computing, is resource management (Duan, Yan, & Vasilakos, 2012). Regarding cloud computing, different customers require various services due to their constantly changing requirements. The objective of CC is to offer all of the necessary facilities, which is a direct result of the previous point. Due to the limited resources that are currently available, it is challenging for providers of cloud services to meet customers' demands for timely delivery of all necessary services. As a result of cloud computing reliance on virtualization technology and its usage of a distributed approach, dynamically adding new resources is now much easier to achieve, whereas, in the past, it was considerably more difficult to do so using conventional resource management approaches (Stergiou, Psannis, Kim, & Gupta, 2018). In the next part, we will discuss the issues of managing resources in a cloud computing environment and describe the many available resources.

### 2-1. Classification of Resources

Here, we will provide a brief overview of the categorizations of key resource categories, including security, network, storage, energy, and computing. The various types of resources that can be utilized through cloud computing are illustrated in Figure 2.

Storage services: Failures in computer systems are almost inevitable over time. Therefore, businesses and individuals require continuity to preserve and maintain backups. StaaS is an acronym that stands for "storage as a service." It is a term for a platform that enables users to access their data stored in the cloud. It has many thousands of database servers and numerous hard drives. The use of SaaS decreases the risks associated with disaster recovery, lowers costs related to physical space and infrastructure, and provides long-term preservation of data. As a result, using Software as a Service (SaaS) enhances both availability and the continuity of work(Bhavani & Guruprasad, 2014).

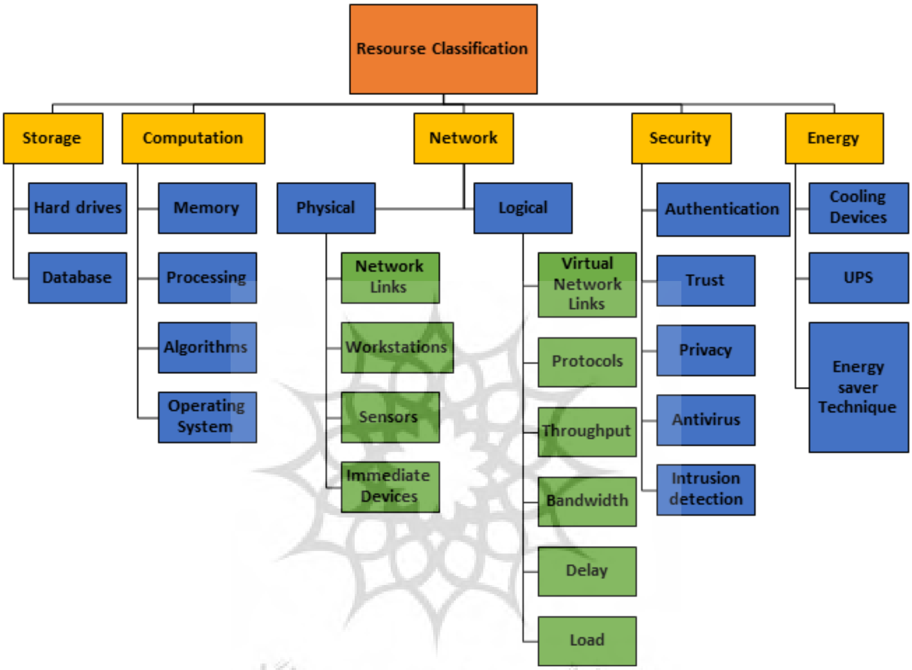


Figure 2. Classification of resources

**Computation services:** "Computation as a Service," or CaaS for short, refers to a fast computing service that can be accessed through cloud computing. The processing power, the amount of memory required, efficient algorithms, and an operating system are all components of it (Bhavani & Guruprasad, 2014).

**Network services:** Both physical and logical resources are included in the construction of a Network as a Service (NaaS) (Leivadeas, Papagianni, & Papavassiliou, 2012). The phrases "physical network links," workstations, and "intermediate devices and sensors" are all examples of various types of physical resources. Logical resources include elements such as throughput, protocols, latency, bandwidth, loads, and virtual network connections, among other factors.

Without network services such as latency and bandwidth, the concepts of storage services and compute services become incomprehensible. Storage services and compute services cannot exist without each other. As every service provided through cloud computing relies on a high-speed Internet connection, these services are considered crucial from a networking perspective (Endo et al., 2011).

**Security services:** One of the most critical challenges in cloud computing is security as a service, sometimes known as SECaaS[25]. SECaaS offers consumers an enhanced level of protection against attacks and threats originating from the internet (Nallakumar, Sengottaiyan, & Priya, 2014). It offers a variety of services such as trust, authentication, penetration testing, intrusion detection, malware and virus protection, and security event administration (Yu, Buyya, & Ramamohanarao, 2008).

**Energy services:** The data centers comprising the cloud have remarkably high levels of energy consumption. The delivery of energy services involves providing physical equipment such as air conditioning units and uninterruptible power supply systems (UPS). The cost of managing underutilized resources has been decreased by implementing various energy-saving strategies. When energy-saving strategies are implemented on servers and networks in data centers, substantial amounts of energy can be conserved (Lakshmi & Srinivasu, 2015).

## 3. SCHEDULING

Cloud computing scheduling is the process of allocating resources within a cloud computing environment to ensure that jobs are completed within a specified time frame. It involves the efficient utilization of resources, including memory, computer processors, and network bandwidth, to ensure that various assigned tasks are completed promptly. Scheduling in cloud computing is mostly performed in two ways: static and dynamic. In static scheduling, different tasks are assigned to various resources in advance. This method is useful for tasks that need to be completed at a predetermined time. On the other hand, dynamic scheduling is more flexible and can assign jobs to the most suitable resource at any given time. Dynamic scheduling can help enhance the performance of an operation or service by providing suitable resources at the right time. Various studies indicate that it helps reduce costs by ensuring that resources are not underutilized. The scheduling process must allocate given tasks to available resources, adhering to a set of specific constraints, to achieve one or more development goals (Xu, Liu, Jin, & Vasilakos, 2013). Within the context of distributed computing systems, scheduling is the process that determines which resources should be used to carry out a given task, taking into account both the dynamic and static characteristics of the work to be done (Casavant & Kuhl, 1988). The characteristics of the activities scheduled within an application each require a unique scheduling method. When a task is part of a sequence, the job in question cannot be scheduled until all of the primary tasks in the sequence have been completed; this type of scheduling is referred to as workflow scheduling. In a different scenario, independent task scheduling refers to arranging activities in a manner that does not depend on the sequence in which they are completed (du Pin Calmon, Cloud, Medard, & Zeng, 2017). This type of scheduling is possible when activities are not dependent on one another.

## 3-1. Procedure Followed in Scheduling

The technique for scheduling in CC can be broken down into different stages: the discovery and monitoring of resources, the selection of resources, and the submission of tasks (Zhan et al., 2015). The three steps are (a) Resource discovery & monitoring (b) Resource Selection and (c) Task submission.

## Resource discovering and monitoring:

In the initial step of the process, the Datacenter Broker (DB) finds all of the available resources presented in the cloud system. It then collects information regarding the current state of all of the accessible resources as well as the potential availability of any remaining resources. In point of fact, the majority of the time, these types of resources are called virtual resources.

**Resource selection:** The cloud scheduler uses the information that was gathered during the discovery phase to make a determination on which target resource should be selected during the second phase. This choice was arrived at after the completion of the discovery phase, based on the information that was gathered during that stage.

**Task submission:** In this phase, the duty is delegated to the most qualified of the available resources that have been chosen.

### 3-2. Formulation of Problem in Task Resource Scheduling

Optimizing task scheduling in cloud computing should aim to determine the optimal number of essential systems to minimize the total cost. Let us assume there are n tasks. The time required to complete each task on each processing machine is already known, and the tasks need to be completed using m of the available computational resources. The objective is to make the most efficient use of all accessible resources while reducing the time required for the overall execution. Assume that there are more tasks than available resources (n > m) and that task migration between resources is not permitted (Lin, Zhong, Lin, Lin, & Zeng, 2014). Consider the collection of tasks described as $CT_i$=1,2,...n, where independent tasks are represented by n, and $CR_j$ =1,2,...m, where m is the number of computing resources needed to address the problem. As a consequence of this, the challenge associated with cloud resource scheduling lies in obtaining an optimal mapping (OM) of tasks ($CT_i$) to resources ($CR_j$) denoted as OM: $T_iR_j$. Figure 3 provides a visual representation of the definition of this issue, which states that two or more tasks may share a single resource (Pacini, Mateos, & Garino, 2015).

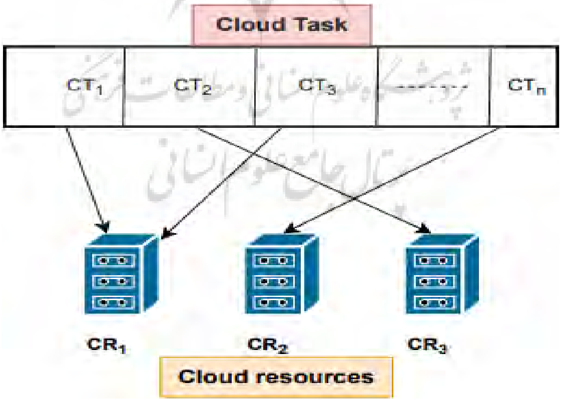


**Figure 3. Task scheduling problem formulation**

## 4. PARAMETERS TO MEASURE QoS

In this part, the parameters that are used to measure how effectively scheduling is done are explained. Existing studies have addressed a variety of optimization criteria, including load balancing, resource utilization, energy efficiency, throughput, and cost, among others, such as cost, makespan, deadline, budget, and deadline. In general, these QoS parameters are divided into two categories depending on cloud service (Dasgupta, Mandal, Dutta, Mandal, & Dam, 2013): The desires of cloud service providers and cloud users. These optimization criteria are discussed in the vast majority of the works that have been evaluated. Therefore, the purpose of this study is to explain how these criteria are investigated using a comparative approach.

### User Desire QOS Parameters

**Makespan (completion time):** Makespan corresponds to the time needed for the last jobs to finish and then leave the cloud system (Jianfang, Junjie, & Qingshan, 2014). The length of time required to finish the task is the conventional definition of it. **Execution Cost:** The user is the one who is responsible for paying the overall cost to the service provider, and the amount that they owe is determined by the number of resources that they utilize (Bajaj, 2021). **Budget**: It illustrates the limits that must be overcome in order to complete the tasks within the budget (Brucker & Du, 2008). **Deadline:** It denotes the completion of all jobs that were currently running at a particular time (Brucker & Du, 2008).

### 4-1. Desire QoS Parameters

**Resource utilization:** Resource utilization refers to maximizing the use of available resources and maintaining a level of activity that keeps those resources busy. It is beneficial for service providers to profit by leasing limited resources to cloud users on an as-needed basis (Blazewicz et al., 2019). **Throughput** determines how many jobs can be completed within a specified time frame (Herrmann, 2006). Load balancing in cloud computing refers to the process of distributing workloads consistently among multiple virtual machines (VMs) and underlying physical resources. The authors in (Yang, Cui, Xiao, Gandomi, & Karamanoglu, 2013) present a wide variety of methods. **Energy efficiency** can be defined as reducing the amount of energy needed to complete a task (Singh, Dutta, & Aggarwal, 2017).

Persistent problems and discrete problems both fall under the category of optimization problems. In a combinatorial problem, the decision variables take on discrete values. In contrast, in a continuous optimization problem, the decision variables can take on any value that falls within the range of real values (Ri) (Kalra & Singh, 2015; Nayak, Malla, & Debadarshini, 2012). This issue can be categorized into single-criterion and multi-criteria based on the number of criteria included in the optimization problem. The objective of optimization based on a single criterion is to find a solution that satisfies that criterion more efficiently than any other. It's possible that a solution that would typically be considered outstanding in this scenario would actually perform worse when measured against another criterion (Sangwan, Sharma, & Kumar, 2017). As a consequence of this, the purpose of multi-criteria optimization is to find a group of solutions that are the best in terms of all of the other criteria. Notably, the majority of situations in real life involve multiple criteria. Modern optimization approaches use meta-heuristic and heuristic-based search strategies to look for solutions. These strategies use deterministic and stochastic search concepts. An algorithm is said to be capable of solving a problem if it can solve all instances As a consequence of this, the goal of multi-criteria optimization is to identify a set of solutions that excel in terms of all the other criteria. Notably, the majority of situations in real life involve multiple criteria. Modern optimization approaches utilize metaheuristic and heuristic-based search strategies to search for solutions. These strategies utilize deterministic and stochastic search concepts. An algorithm is considered capable of solving a problem if it can solve all instances of the problem (P). In most cases, we are interested in the approach that provides the most efficient solution to the problem. Efficiency is often associated with the computer resources (time and space) required to execute a method (Elmougy, Sarhan, & Joundy, 2017; Li, Ma, Tang, Shen, & Jin, 2017).

## 5. TASK SCHEDULING TECHNIQUES

This section gives the categorized work dedicated to task scheduling in a cloud computing. Because of the current number of tasks and the dynamic nature of the cloud-computing environment, task scheduling has evolved into an operation that requires a significant amount of time. Jobs can be submitted from any part of the world, and the rate at which they arrive is not regulated when the issue of task scheduling is particularly complex. Consequently, according to the research that has been conducted, commonly used algorithms in the context of task scheduling can be broadly categorized as either dynamic or static, depending on how jobs are received (Moreno-Vozmediano, Montero, & Llorente, 2012).

**Static scheduling algorithms** are simple to construct; the fluctuating job load makes them unsuitable for usage in the real world.

**Dynamic Algorithm-** The dynamic approach solves this problem by scheduling jobs as soon as they are submitted to the scheduler (Baliga, Ayre, Hinton, & Tucker, 2010; Yu & Buyya, 2006). Dynamic algorithms are further categorized to distributed and centralized types.

**Centralized method**- Jobs are scheduled in the cloud using a centralized method (Kalra & Singh, 2015) that runs on a single node. Despite being straightforward, the strategy has a bottleneck and a hefty burden.

**Distributed method** (Herrmann, 2006; Pandey, Wu, Guru, & Buyya, 2010) resolves the problems of the centralized approach where they use approximatively , and heuristic approaches to schedule jobs in the cloud environment (Lin et al., 2014; Raghavan, Sarwesh, Marimuthu, & Chandrasekaran, 2015).

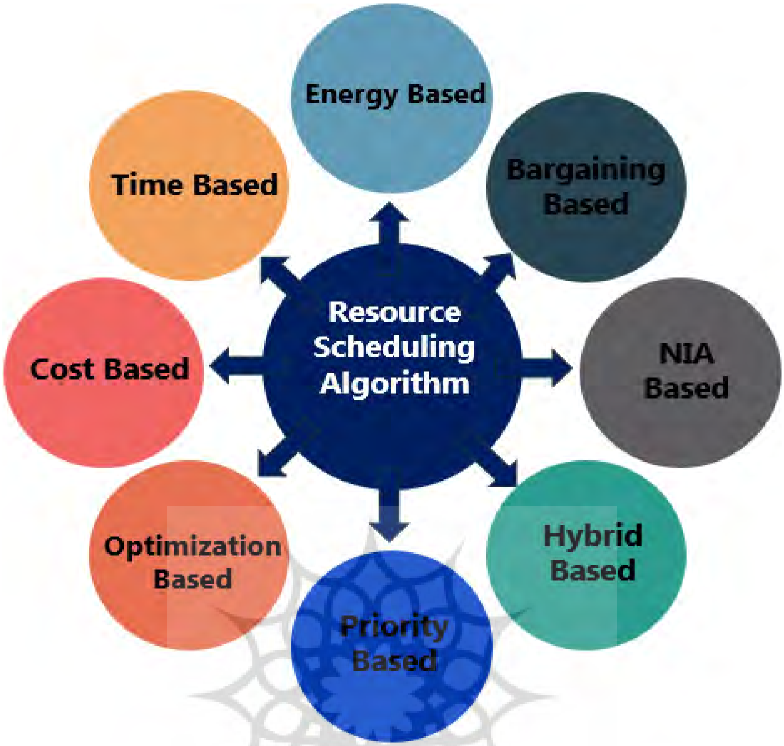


**Figure 4. Resource scheduling algorithm based on different criteria**

Figure 4 provides a classification of the various types of scheduling algorithms proposed in research literature, each based on specific parameter criteria.

The method is utilized in an environment with diverse characteristics to improve cost, resource capacity, completion time, and energy parameters. The authors of (Son & Jun, 2013) described a bargaining-based technique for scheduling that balances the trade-off between resources to improve negotiation speed and help identify the best service provider. The method reduces the percentage of unsuccessful jobs while also improving SLA waiting time and throughput. However, the technique will result in a greater deviation from Second Language Acquisition (SLA) standards. The authors (Zong, 2020) studied multi-objective functions concerning cost and energy in scheduling operations in a cloud environment while ensuring deadlines are met. The groups integrated methods to save energy as well as costs. Based on optimization algorithms, scheduling techniques are divided into three types: traditional scheduling techniques, heuristic scheduling techniques, and meta-heuristic scheduling approaches.

### Traditional Techniques:

Traditional methods such as “First Come First Serve (FCFS), Shortest Job First (SJF) and Round Robin (RR)” are key to scheduling various activities (Thakur & Mahajan, 2017). These methods are straightforward, quick, and deterministic, and they yield precise solutions (Sharma, Tyagi, & Atri, 2017). However, in many circumstances, they are ineffective for comprehending the optimality problem (Raj & Prasad, 2016). Therefore, standard methods cannot be used for scheduling in a cloud context (Bhoi & Ramanuj, 2013). Numerous efforts have been made to enhance the use of traditional approaches (Chen, Wang, Helian, & Akanmu, 2013; Sörensen & Glover, 2013; Thakur & Mahajan, 2017). One such strategy that uses a time slice or quantum is the Round Robin. The RR algorithm’s use of static time quantum is a flaw (Chen et al., 2013). Round-robin scheduling is the foundation of the suggested CPU scheduling (Yang, 2010). However, the scheduling calculations are done differently. SJF is a scheduling method that considers the task’s execution time. The jobs are prioritized in a queue, with the greatest priority tasks being placed first and the lowest priority tasks being placed last (Tsai & Rodrigues, 2013). The task with the least burst time is given CPU attention in this algorithm. A hybrid method of RR and SJF termed the SRDQ algorithm was proposed by Elmougy et al. (Al-maamari & Omara, 2015). This algorithm takes quantum time into account for a dynamic variable job.

### Meta-heuristic Techniques:

It is an NP-Hard issue to allocate tasks to resources in a cloud computing environment. Consequently, the task schedule is made clearer by using meta- and heuristic heuristics to produce close to ideal or optimal solutions. Swarm intelligence (SI) and bio-inspired are the two subcategories of meta-heuristic methods (Raghavan et al., 2015). Bioinformatics engineering, control engineering, data mining, and pipelining are just a few fields of science that have been influenced by biological principles. Numerous bio-inspired algorithms exist, including the Imperialist Competitive Algorithm (ICA), Genetic Algorithm (GA), and Memetic Algorithm (MA). Swarm intelligence is a relatively new method for solving unconstrained optimization problems. It is based on the social behavior of animal and insect communities, including Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Artificial Bee Colonies, Glow-worm Swarm Algorithms, Bat Algorithm (BA), Firefly Algorithms, Cuckoo Searches, and Cat Swarm Optimization (CSO). Better algorithms are constantly being sought after, especially for scheduling tasks in cloud computing. Numerous bio-inspired algorithms exist, including the imperative competitive algorithm, GA, and MA (ICA). Swarm intelligence is a relatively new method for solving unconstrained optimization issues. It is based on the social behaviour of animal and insect communities, including PSO, ACO, artificial bee colonies, glow-worm swarm algorithms, BA, Firefly algorithms, cuckoo searches, and cat swarm optimization (CSO). Better algorithms are constantly being sought after, particularly for cloud computing task scheduling. Table 2 provides a comparison of various algorithms. The comparison considers the optimization measures, the nature of the task, the size of the experiment, and the simulation environment.

## Nature-Inspired Algorithms: How Do They Apply to Cloud Computing?

Nature-inspired optimization techniques, especially algorithms that simulate adaptive, distributed, and intelligent behaviour observed in nature, are of great importance and scope in the field of cloud computing. These algorithms simulate adaptive and intelligent behaviour observed in nature to address a variety of challenges, such as resource optimization, fault tolerance, load balancing, and security in cloud computing. Cloud environments are characterized by dynamic and heterogeneous resources, complex interactions between different components, and fluctuating workloads. These challenges require efficient resource management, workload allocation, and optimization strategies to ensure optimal performance, cost efficiency, and resource utilization. Nature-inspired metaheuristic algorithms offer attractive solutions to address these challenges in cloud computing environments. For example, genetic algorithms can be used for resource provisioning and allocation. Here, virtual machines and containers are dynamically allocated to physical servers based on workload characteristics and resource availability.

Similarly, particle swarm optimization can optimize task scheduling and load balancing. It intelligently distributes tasks among available resources to minimize response time and maximize throughput. Moreover, metaheuristic methods can adapt to changing conditions and optimize system parameters in real-time. This makes it suitable for dynamic and unpredictable cloud environments. These algorithms can continuously evaluate system health, adjust resource allocation, and optimize configuration to maintain performance levels, alleviate bottlenecks, and improve overall system efficiency.

## "Simple Genetic Algorithm (SGA)"

A "three-stage selection technique" was reported by Hu Yao et al. in 2017, while the genetic approach of "total-division-total" is offered as an improvement to the genetic strategy (Yao, Fu, Li, Dong, & Li, 2017). This algorithm overtakes a simple genetic algorithm (SGA) in terms of completion time of tasks and overall performance.

## "Load Balancing Ant Colony Optimization (LB-ACO)"

Gupta and their co-worker used a meta-heuristic technique based on the Ant Colony Optimization to tackle task scheduling in a cloud setting, keeping two key goals in mind: reducing makespan or computation time and improving load balancing (Gupta & Garg, 2017). Their simulation study reveals that their proposed Load Balancing Ant Colony Optimization Algorithm (LB-ACO) outperforms the NSGA-II algorithm, delivering superior load balancing and a smaller makespan.
makespan.

## "Simulated Annealing Multi-Population Genetic Technique (SAMPGA) algorithm"

Xing et al. used the max-min algorithm to enhance search efficiency in his proposed work (Wei, Bei, & Jun, 2017). They integrate SA into SAMPGA to avoid local optima, thereby increasing global optimization performance. In addition to this, they introduce a family evolution strategy with an adaptive mechanism in MPGA to discover superior solutions and speed convergence. Simulation conducted in MATLAB assesses the effectiveness of their method. Comparative analysis against MPGA, SA, and SAGA reveals that SAMPGA outperforms in completion time, completion cost, convergence speed, and load balance degree, as evidenced by simulation results.

### "Flower Pollination Algorithm"

Jamspud Kaur and her colleagues employed a new job scheduling algorithm that exploited the pollen dispersal method (Kaur & Sidhu, 2017). The purpose of this algorithm is to allocate resources to tasks. Simulation results show that the proposed algorithm speeds up the process of completing the task. The proposed method outperforms previously proposed algorithms in terms of completion time. This is determined by comparing the effect of the proposed method with some other methods such as FCFS, GA, and Round Robin (RR) scheduling strategy.

### "Cat Swarm Optimization"

The multi-objective QoS method is proposed by Danlami and his collaborators as a way to meet customer expectations by considering cost and execution time (Gabi, Ismail, Zainal, Zakaria, & Al-Khasawneh, 2017). This group showed that the most effective method is to use the "simulated annealing based on cloud-scalable multi-objective cat swarm optimization" method to solve above mentioned cloud-related problem. To make the local search more exploratory, this group adopted the proposed approach and used the Taguchi orthogonal technique to improve the quality of simulated annealing.

### Parallel genetic algorithm

Effective utilization of resources in cloud systems is proposed by Meran Ashouraei and colleagues (Ashouraei, Khezr, Benlamri, & Navimipour, 2018) by adopting parallel genetic algorithms, which has priority. To eliminate job failures, this priority based GA enhances load balancing when selecting qualified resources for urgent tasks.

### Ant Colony Method

In 2018, researchers Fang Yiqiu and group discovered an ant colony approach for a virtual machine that operates in real-time (Yiqiu, Xia, & Junwei, 2019). The group adopted a method that takes into consideration the allotted amount of time in order to complete the load balancing procedure successfully. The simulation results show that the VM-ACO approach is superior as compare to ant colony algorithm when it comes to latency, load balancing, resource status polling, and overall processing time.

## Rock Hyrax Optimization

In 2021, Saurabh et. al. presented a proposed approach that uses rock hyrax optimization capabilities for job scheduling in dynamic and heterogeneous cloud environments, taking into account QoS characteristics such as completion time and data centre power use (Singhal & Sharma, 2021). The outcome supports the assertion that by minimizing energy usage, the approach made in this study may schedule tasks in a dynamic environment on a virtual computer.

## "Advanced Phasmatodea Population Evolution (APPE) algorithm"

Guo Yiping and his research team introduced a new method, which they named QoS-oriented global multipath traffic scheduling algorithm (QOGMPT). This new algorithm is created by combining a link weight calculation algorithm with a traffic scheduling algorithm. Experimental results show that the proposed algorithm outperforms the two compared algorithms in terms of delay amount and rescheduling rate.

## Aquila Optimizer (Abualigah et al., 2021)

Strengthening the narrative on the Aquila Optimizer's potential to enhance task scheduling efficiency in cloud computing requires emphasizing its unique adaptability, optimization capabilities, and resilience. Although the Aquila Optimizer has not been directly used for calculating Quality of Service (QoS) parameters, its adaptability reflects the dynamic nature of cloud environments, allowing it to adjust scheduling strategies based on real-time data. Moreover, its optimization features facilitate the fine-tuning of task placement and execution timing, thereby minimizing latency and maximizing resource utilization to meet stringent Quality of Service (QoS) requirements. Drawing inspiration from nature, the Aquila Optimizer's inherent resilience equips it to handle failures and mitigate bottlenecks, ensuring the maintenance of optimal service levels by intelligently redistributing tasks and resources. This elaboration underscores how integrating the Aquila Optimizer's unique characteristics can significantly enhance cloud scheduling efficiency. It offers clarity and focus while also laying the groundwork for innovative scheduling techniques without infringing upon existing work.

**Table 1. Major research findings**

| Year | Applied Algorithm | Parameters | Finding | Tools | Ref. |
|---|---|---|---|---|---|
| 2017 | Improvement Genetic Algorithm) | Makespan | Enhance the overall efficiency of job scheduling in comparison to the Simple Genetic algorithm | CloudSim | (Yao et al., 2017) |
| 2017 | SAMPGA Algorithm | Convergence Speed Load Balancing, Cost, Makespan | SAMPGA outperforms SAGA, MPGA, and SA with regard to the total cost, amount of time needed to complete, rate of convergence, and load imbalance. | MATLAB | (Wei et al., 2017) |
| 2017 | TSFPA algorithm | Makespan | TSFPA's performance is superior to that of RR, and FCFS,GA when measured in terms of makespan. | CloudSim | (Kaur & Sidhu, 2017) |
| 2018 | (GGWO.) Algorithm | Consumption, Load Utilization, Computation Time, Cost, Energy | GGWO will result in an improvement in task scheduling in comparison to GA and ordinary GWO. | CloudSim | (Gobalakrishnan & Arun, 2018) |
| 2018 | "Parallel Genetic Algorithm" | "Energy Usage, Migration Rate, Resource Utilization, Load Balancing" | "Improve the load balance by picking better tools to carry out the arrival duties at a lower failure rate and in a shorter amount of time." | MATLAB | (Ashouraei et al., 2018) |

| Year | Applied Algorithm | Parameters | Finding | Tools | Ref. |
|---|---|---|---|---|---|
| 2018 | “VM-ACO algorithm” | “Execution Time Load Balancing Task Transmission” | “In terms of task latency, the amount of time it takes to complete a task, and load balance, it performs far better than the Ant colony algorithm.” | CloudSim | (Yiqiu et al., 2019) |
| 2019 | “A new method for scheduling workload based on VM grouping” | Response Time, Resource Utilization Makespan | Both the makespan time and the average response time for VMs can be cut down by using the grouping strategy. This algorithm improves resource consumption in comparison to SJF and Min-Min algorithms, which both have their advantages. | CloudSim | (Chitgar, Jazayeriy, & Rabiei, 2019) |
| 2019 | Adaptive genetic algorithm (AGA.) | Load Balancing Make Span | “When compared with the adaptive (AGA) and the static (SGA) approach, a good influence on resource scheduling produces a task scheduling result that is both more reasonable and more optimal.” | CloudSim | (Yiqiu et al., 2019) |
| 2019 | “(GA-PSO) algorithm” | Response Time | The result of the hybrid GA-PSO is better than PSO Algorithm | CloudSim | (Kumar, Parthiban, & Shankar, 2019) |

| Year | Applied Algorithm | Parameters | Finding | Tools | Ref. |
|---|---|---|---|---|---|
| 2019 | MGWO algorithm | Degree Of Imbalance Makespan, Cost, | Traditional Grey (GWO) and (WOA) algorithms are superior to MGWO in terms of performance in the categories of cost, makespan, and degree of imbalance respectively. | CloudSim | (Abdulhamid, Abd Latiff, Madni, & Abdullahi, 2018) |
| 2019 | MA algorithm | Cost And Makespan | The MA improved performance over both the PSO and GA as well as reducing the overall makespan. | CloudSim | (Alsmady, Al-Khraishi, Mardini, Alazzam, & Khamayseh, 2019) |
| 2019 | IDPSO algorithm | Convergence And Completion Time, | IDPSO is superior to both FIFO and DPSO in terms of the amount of time it takes to finish and how quickly it converges. | CloudSim | (Liu & Yin, 2019) |
| 2019 | EDA-GA algorithm | Completion Time, Load Balancing And Convergence | The EDA-GA algorithm offers greater convergence and search capability, as well as the ability to minimize the amount of time needed to complete tasks and improve load balancing in contrast to the GA and EDA | CloudSim | (Pang, Li, He, Shan, & Wang, 2019) |

| Year | Applied Algorithm | Parameters | Finding | Tools | Ref. |
|---|---|---|---|---|---|
| 2019 | Humpback Whale Optimization | Cost, Energy Resource Utilization ,Makespan | “When compared to the WOA and Round Robin algorithms, the performance of the suggested algorithm was superior in terms of cost, degree of imbalance, energy consumption, makespan, and resource utilization”. | CloudSim | (Masadeh, Sharieh, & Mahafzah, 2019) |
| 2020 | BMin algorithm | Throughput Load Balancing Completion Time | Reduce the amount of time needed to finish, as well as enhance the load balance in comparison to Min-min. | CloudSim | (Shi, Suo, Kemp, & Hodge, 2020) |
| 2020 | ERR algorithm | Execution Time, Residue Energy Waiting Time | “When compared to Round Robin, the total amount of waiting time for EnhanchedRR tasks was shorter when the identical conditions were applied. In terms of both its execution time and its residue energy, the EnhancedRR algorithm outperforms other algorithms such as MPA, ACO, GA, PSOand Min-Min”. | CloudSim | (Sanaj & Prathap, 2020) |
| 2020 | ERAS algorithm | Makespan. Reliability | When it comes to allocation, the ERAS algorithm offers more reliability along with improved performance when compared to the EFT method. | CloudSim | (Lepakshi & Prashanth, 2020) |

| Year | Applied Algorithm | Parameters | Finding | Tools | Ref. |
|---|---|---|---|---|---|
| 2020 | combined GA and ACA. | Energy Execution Time, | The proposed approach cuts down on the amount of time as well as the overall energy consumption that jobs for cloud-based computing require. | CloudSim | (Zong, 2020) |
| 2021 | IPSO Task scheduling | Resource Utilization Make Span | Maximize the resource utilization. improves the standard deviation and minimizes the makespan. | CloudSim | (Saleh, Nashaat, Saber, & Harb, 2018) |
| 2021 | Rock Hyrax for scheduling | Makespan Energy Consumption | Uses nature-inspired algorithm optimize the QoS Parameter | CloudSim | (Singhal & Sharma, 2021) |
| 2022 | DS-DT Algorithm | Makespan Resource Utilization | Algorithm for scheduling a workflow application to optimize the QoS Parameter in heterogenous environment. | CloudSim | (Mahmoud, Thabet, Khafagy, & Omara, 2022) |
| 2022 | APPE Algorithm | Makespan, Cost and Load Balancing | Provides a way for scheduling jobs in a cloud architecture that is decentralised and has varied degrees of similarity across its components. | | (A.-N. Zhang, Chu, Song, Wang, & Pan, 2022) |
| 2022 | QOGMPT Algorithm | Delay, Rescheduling Rate | The experimental findings show that the proposed technique reduces the amount of delay while increasing the rescheduling rate. | Pycharm-community-2019.1.1 | (Guo, Hu, & Shao, 2022) |

## 6. RESULTS AND DISCUSSION

### Process and Source of Data collection

In August 2023, without any specific time constraint, a search was carried out focusing on the article that matched our title. This search yielded 525 articles published between 2010 and 2023. The research articles were sourced from reputable platforms such as Springer, IEEE, Elsevier, Wiley, Hindawi, MDPI, and Google Scholar to identify relevant content. Out of 525, we selected 200 articles in which desired QoS parameters were discussed (Figure 6). This paper undertakes a comprehensive review of existing literature to assess the current endeavors and emerging trends in cloud computing, specifically focusing on enhancing Quality of Service (QoS). In the first phase, we used Google Scholar as the primary search platform. Keywords related to our work are identified to shape the search domain. These keywords include QoS, resource management, scheduling, load balancing, and optimization. Logical operators such as "AND" and "OR" are employed to connect these keywords and locate relevant articles.

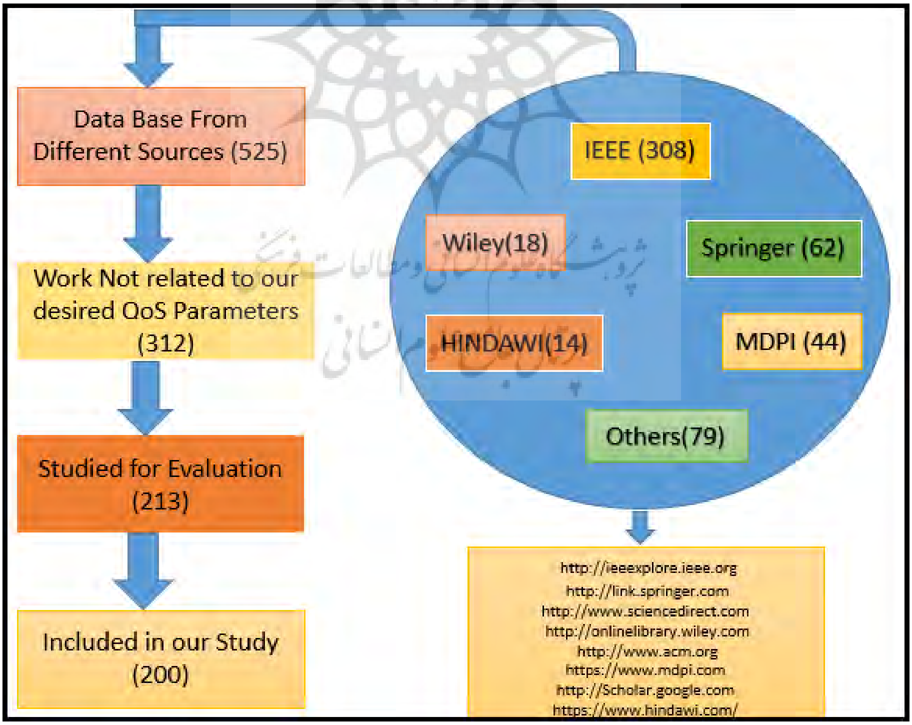


Figure 5. Process and source of data collection

## Data Analysis

In this part, we analyze and explore meta-heuristic methods for job scheduling in cloud computing. Meta-heuristic techniques are categorized and optimization criteria are used to frame the discussion. According to the many optimization criteria listed in section 3.4, several strategies are examined. Figure 6 provides insights into the QoS parameters usually optimized within cloud computing environments by using various optimization techniques. “Makespan,” topping the list at 34%, reflects the emphasis on minimizing task completion time, a crucial factor in ensuring efficient resource utilization and customer satisfaction in cloud services. The significant focus on “Cost” optimization at 17% underscores the economic considerations inherent in cloud computing, where businesses seek to maximize ROI while delivering high-quality services. “Load Balancing” at 15% highlights the importance of evenly distributing workloads across cloud resources to avoid bottlenecks and maintain system performance. “Deadline” and “Resource Utilization” at 10% and 8% respectively demonstrate the attention to meeting time constraints and efficiently utilizing available resources in cloud environments. The lower percentages for “Budget” (4%), “Throughput” (3%), and “Energy Efficiency” (9%)

Both the user and the suppliers favor different sets of these characteristics. Users place a greater emphasis on timeliness, cost, budget, and load balancing over resource consumption, throughput, load balancing, and energy efficiency, which are all important to service providers. For cloud customers who want to hasten the execution of their apps, makespan minimization is a key consideration. The makespan metric of the applications is the primary metric used by scheduling algorithms to make decisions. Thus, reducing the makespan became the researchers’ top priority to improve performance. In addition, the researcher is most interested in cost. In the literature, various expenses have been taken into account, including those for communication, computing, data storage, transfer, and renting.

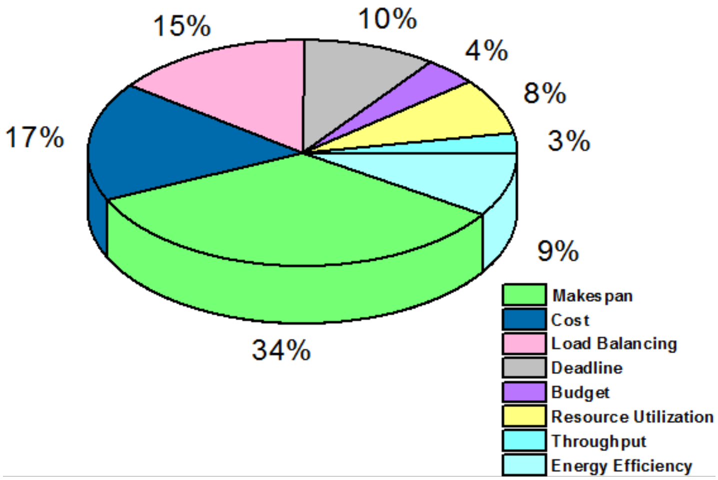


**Figure 6. Parameters Considered by various research group (based on 200 research articles)**

Data management becomes more challenging and expensive because responsibilities are distributed across multiple disparate data centers of various cloud service providers. Additionally, if the data is kept in permanent storage, the price will increase. On the other hand, when data is shared among multiple applications or stored in a single data center of the same cloud service provider, data storage costs are reduced to a negligible level. Resource utilization and virtual machine load balancing are two of the most important factors that service providers consider. These elements affect the success and profitability of providers. Resource consumption increases when the workloads of virtual machines are combined. Virtual machines can be turned on or off and used for other tasks. The energy usage of data centers is reduced by turning off the virtual machines (VMs), which is a significant aspect for service providers. Budget and throughput in task scheduling have rarely been considered by researchers. In Figure 7, it is shown that only 4% of the total approaches are scored by both of them. PSO is the method most frequently utilized by researchers in the field of cloud scheduling difficulties, whereas GA is the method most commonly used in the field of bio-inspired methodologies.

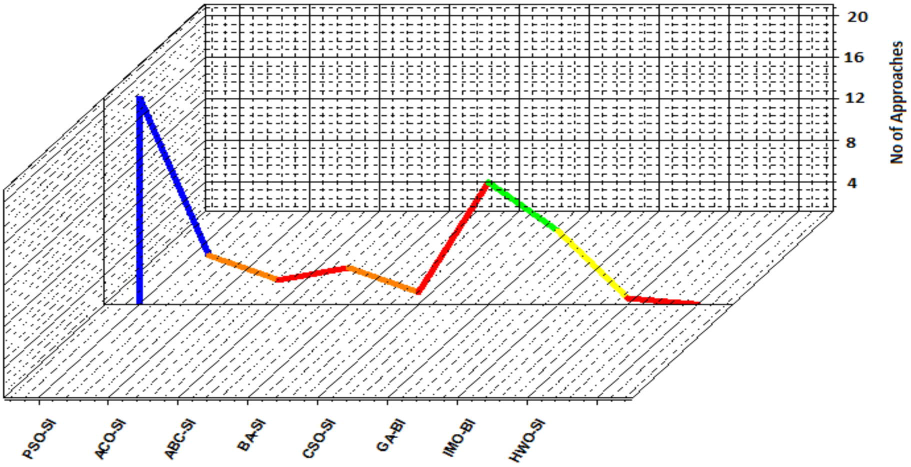


**Figure 7. No of approaches based on various algorithm**

## Challenges and opportunities in task scheduling algorithms

Differing priorities between end users and service providers in job scheduling optimization have profound implications for the design and implementation of various task scheduling algorithms. Users typically prioritize factors such as job completion time, cost-effectiveness, and resource utilization. In contrast, service providers focus on maximizing resource utilization, minimizing overall operational costs, and meeting user expectations. Balancing these contradictory priorities poses significant challenges in algorithmic complexity, trade-offs between different objectives, adaptability to dynamic environments, and ensuring fairness in multi-user scenarios.

One of the main challenges is designing algorithms that can effectively balance these conflicting priorities without sacrificing performance in cloud computing. Hybrid approaches that integrate different scheduling strategies, such as priority-based, cost-based, or deadline-driven approaches, offer promising solutions. Furthermore, by introducing machine learning algorithms to predict user preferences and optimize scheduling decisions can further enhance algorithm performance in dynamic environments. Continuous feedback mechanisms and customizable scheduling parameters also provide opportunities for improving algorithmic efficiency and user satisfaction. By adopting above strategies, task scheduling algorithms can better align with the diverse needs and priorities of both users and service providers, leading to more efficient and satisfactory job scheduling outcomes.

## 7. Conclusion and Future Directions

This review manuscript discusses the fundamental concepts of two key components of cloud computing: resource management and scheduling. In addition, we offer a comparison of various cloud computing scheduling approaches. The investigation considered job types, user and provider preferences, simulation settings, and service quality standards. We reviewed over 100 articles published between the beginning of 2017 and 2022 to thoroughly examine the relevant literature. Finally, we examined 25 studies that focused on utilizing scheduling techniques in cloud computing. The majority of the scheduling algorithms were found to take into consideration delay, cost, makespan, load balancing, and energy, with each factor contributing around 10%, 17%, 34%, 15%, and 9%, respectively. The assessment metrics indicated that these factors were taken into account. The researchers also concluded that measures related to availability and Service Level Agreement (SLA) played the smallest role in overall relevance. The purpose of this study was to conduct comprehensive research on cloud computing scheduling algorithms and their fundamental properties. After analyzing the relevant body of research, we concluded that the majority of works are based on well-known meta-heuristics such as PSO, GA, and ACO algorithms. Existing algorithms are mostly focused on the delay, cost, and makespan matrix. Therefore, it is suggested that for future research, a scheduling algorithm be proposed that can support dynamic environments and pay more attention to evaluation metrics such as availability, security, and energy parameters. This is because the proposed algorithm would replace the existing algorithms with one that could support dynamic environments. We have incorporated novel algorithms that have not been previously explored in the realm of cloud computing. This innovative inclusion paves the way for future research opportunities and highlights the potential for advancements in cloud computing technology. This contribution will have a significant impact on the field.

## References

Abdulhamid, S. i. M., Abd Latiff, M. S., Madni, S. H. H., & Abdullahi, M. (2018). Fault tolerance aware scheduling technique for cloud computing environment using dynamic clustering algorithm. *Neural Computing and Applications, 29*(1), 279-293. DOI: https://doi.org/10.1007/s00521-016-2448-8

Abualigah, L., Yousri, D., Abd Elaziz, M., Ewees, A. A., Al-Qaness, M. A., & Gandomi, A. H. (2021). Aquila optimizer: a novel meta-heuristic optimization algorithm. *Computers & Industrial Engineering, 157*, 107250. https://doi.org/10.1016/j.cie.2021.107250

Al-maamari, A., & Omara, F. A. (2015). Task scheduling using hybrid algorithm in cloud computing environments. *Journal of Computer Engineering (IOSR-JCE), 17*(3), 96-106. DOI: 10.9790/0661-173696106

Alsmady, A., Al-Khraishi, T., Mardini, W., Alazzam, H., & Khamayseh, Y. (2019). *Workflow scheduling in cloud computing using memetic algorithm.* Paper presented at the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT). DOI: 10.1109/JEEIT.2019.8717430

Ashouraei, M., Khezr, S. N., Benlamri, R., & Navimipour, N. J. (2018). *A new SLA-aware load balancing method in the cloud using an improved parallel task scheduling algorithm.* Paper presented at the 2018 IEEE 6th international conference on future internet of things and cloud (FiCloud). DOI: 10.1109/FiCloud.2018.00018

Bajaj, S. (2021). Current drift in energy efficiency cloud computing: new provocations, workload prediction, consolidation, and resource over commitment. In *Research Anthology on Architectures, Frameworks, and Integration Strategies for Distributed and Cloud Computing* (pp. 1198-1214): IGI Global. DOI: 10.1109/FiCloud.2018.00018

Baliga, J., Ayre, R. W., Hinton, K., & Tucker, R. S. (2010). Green cloud computing: Balancing energy in processing, storage, and transport. *Proceedings of the IEEE, 99*(1), 149-167. DOI:10.1109/JPROC.2010.2060451

Bhavani, B., & Guruprasad, H. (2014). Resource provisioning techniques in cloud computing environment: a survey. *International Journal of Research in Computer and Communication Technology, 3*(3), 395-401.

Bhoi, U., & Ramanuj, P. N. (2013). Enhanced max-min task scheduling algorithm in cloud computing. *International Journal of Application or Innovation in Engineering and Management (IJAIEM), 2*(4), 259-264.

Blazewicz, J., Ecker, K. H., Pesch, E., Schmidt, G., Sterna, M., & Weglarz, J. (2019). Scheduling in logistics. In *Handbook on Scheduling* (pp. 761-811): Springer.

Brucker, P., & Du, D. (2008). Scheduling Algorithms. *SIAM review, 50*(1), 169.

Buyya, R., Broberg, J., & Goscinski, A. M. (2010). *Cloud computing: Principles and paradigms*: John Wiley & Sons.

Casavant, T. L., & Kuhl, J. G. (1988). A taxonomy of scheduling in general-purpose distributed computing systems. *IEEE Transactions on software engineering, 14*(2), 141-154. DOI: 10.1109/32.4634

Chen, H., Wang, F., Helian, N., & Akanmu, G. (2013). *User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing.* Paper presented at the 2013 National Conference on Parallel computing technologies (PARCOMPTECH). DOI: 10.1109/ParCompTech.2013.6621389

Chitgar, N., Jazayeriy, H., & Rabiei, M. (2019). *Improving cloud computing performance using task scheduling method based on vms grouping.* Paper presented at the 2019 27th Iranian Conference on electrical engineering (ICEE). DOI: 10.1109/IranianCEE.2019.8786391

Dasgupta, K., Mandal, B., Dutta, P., Mandal, J. K., & Dam, S. (2013). A genetic algorithm (ga) based load balancing strategy for cloud computing. *Procedia Technology, 10*, 340-347. https://doi.org/10.1016/j.protcy.2013.12.369

du Pin Calmon, F., Cloud, J. M., Medard, M., & Zeng, W. (2017). Multi-path data transfer using network coding. In: Google Patents.

Duan, Q., Yan, Y., & Vasilakos, A. V. (2012). A survey on service-oriented network virtualization toward convergence of networking and cloud computing. *IEEE Transactions on Network and Service Management, 9*(4), 373-392. DOI: 10.1109/TNSM.2012.113012.120310

Elmougy, S., Sarhan, S., & Joundy, M. (2017). A novel hybrid of Shortest job first and round Robin with dynamic variable quantum time task scheduling technique. *Journal of Cloud computing, 6*(1), 1-12. DOI 10.1186/s13677-017-0085-0

Endo, P. T., de Almeida Palhares, A. V., Pereira, N. N., Goncalves, G. E., Sadok, D., Kelner, J., ... Mangs, J.-E. (2011). Resource allocation for distributed cloud: concepts and research challenges. *IEEE network, 25*(4), 42-46. DOI: 10.1109/MNET.2011.5958007

Gabi, D., Ismail, A. S., Zainal, A., Zakaria, Z., & Al-Khasawneh, A. (2017). *Cloud scalable multi-objective task scheduling algorithm for cloud computing using cat swarm optimization and simulated annealing.* Paper presented at the 2017 8th International Conference on Information Technology (ICIT). DOI: 10.1109/ICITECH.2017.8080065

Gobalakrishnan, N., & Arun, C. (2018). A new multi-objective optimal programming model for task scheduling using genetic gray wolf optimization in cloud computing. *The Computer Journal, 61*(10), 1523-1536. https://doi.org/10.1093/comjnl/bxy009

Goyal, S. (2014). Public vs private vs hybrid vs community-cloud computing: a critical review. *International Journal of Computer Network and Information Security, 6*(3), 20-29. DOI: https://doi.org/10.5815/ijcnis.2014.03.03

Guo, Y., Hu, G., & Shao, D. (2022). QOGMP: QoS-oriented global multi-path traffic scheduling algorithm in software defined network. *Scientific Reports, 12*(1), 14600.

Gupta, A., & Garg, R. (2017). *Load balancing based task scheduling with ACO in cloud computing.* Paper presented at the 2017 International Conference on Computer and Applications (ICCA). DOI: 10.1109/COMAPP.2017.8079781

Herrmann, J. W. (2006). *Handbook of production scheduling* (Vol. 89): Springer Science & Business Media.

Jianfang, C., Junjie, C., & Qingshan, Z. (2014). An optimized scheduling algorithm on a cloud workflow using a discrete particle swarm. *Cybernetics and Information Technologies, 14*(1), 25-39. DOI: https://doi.org/10.2478/cait-2014-0003

Kalra, M., & Singh, S. (2015). A review of metaheuristic scheduling techniques in cloud computing. *Egyptian informatics journal, 16*(3), 275-295. https://doi.org/10.1016/j.eij.2015.07.001

Kaur, J., & Sidhu, B. K. (2017). *A new flower pollination based task scheduling algorithm in cloud environment.* Paper presented at the 2017 4th International Conference on Signal Processing, Computing and Control (ISPCC). DOI: 10.1109/ISPCC.2017.8269722

Kumar, A. S., Parthiban, K., & Shankar, S. S. (2019). *An efficient task scheduling in a cloud computing environment using hybrid Genetic Algorithm-Particle Swarm Optimization (GA-PSO) algorithm.* Paper presented at the 2019 International Conference on Intelligent Sustainable Systems (ICISS). DOI: 10.1109/ISS1.2019.8908041

Lakshmi, R. D., & Srinivasu, N. (2015). A review and analysis of task scheduling algorithms in different Cloud computing environments. *International Journal of Computer Science and Mobile Computing, 4*(12), 235-241.

Leivadeas, A., Papagianni, C., & Papavassiliou, S. (2012). Efficient resource mapping framework over networked clouds via iterated local search-based request partitioning. *IEEE Transactions on Parallel and Distributed Systems, 24*(6), 1077-1086. DOI: 10.1109/TPDS.2012.204

Lepakshi, V. A., & Prashanth, C. (2020). *Efficient resource allocation with score for reliable task scheduling in cloud computing systems.* Paper presented at the 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA). DOI: 10.1109/ICIMIA48430.2020.9074914

Li, J., Ma, T., Tang, M., Shen, W., & Jin, Y. (2017). Improved FIFO scheduling algorithm based on fuzzy clustering in cloud computing. *Information, 8*(1), 25.

Lim, S. Y., Kiah, M. M., & Ang, T. F. (2017). Security issues and future challenges of cloud service authentication. *Acta Polytechnica Hungarica, 14*(2), 69-89.

Lin, J., Zhong, Y., Lin, X., Lin, H., & Zeng, Q. (2014). Hybrid ant colony algorithm clonal selection in the application of the cloud's resource scheduling. *arXiv preprint arXiv:1411.2528*.

Liu, S., & Yin, Y. (2019). *Task scheduling in cloud computing based on improved discrete particle swarm optimization.* Paper presented at the 2019 2nd International Conference on Information Systems and Computer Aided Education (ICISCAE). DOI: 10.1109/ICISCAE48440.2019.221703

Mahmoud, H., Thabet, M., Khafagy, M. H., & Omara, F. A. (2022). Multiobjective Task Scheduling in Cloud Environment Using Decision Tree Algorithm. *IEEE Access, 10*, 36140-36151. DOI: 10.1109/ACCESS.2022.3163273

Masadeh, R., Sharieh, A., & Mahafzah, B. (2019). Humpback whale optimization algorithm based on vocal behavior for task scheduling in cloud computing. *International Journal of Advanced Science and Technology, 13*(3), 121-140.

Moreno-Vozmediano, R., Montero, R. S., & Llorente, I. M. (2012). Key challenges in cloud computing: Enabling the future internet of services. *IEEE Internet Computing, 17*(4), 18-25. DOI: 10.1109/MIC.2012.69

Nallakumar, R., Sengottaiyan, N., & Priya, K. (2014). A survey on scheduling and the attributes of task scheduling in the cloud. *Int. J. Adv. Res. Comput. Commun. Eng, 3*(10), 8167-8171. DOI: 10.1109/ICIIBMS55689.2022.9971622

Nayak, D., Malla, S. K., & Debadarshini, D. (2012). Improved round robin scheduling using dynamic time quantum. *International Journal of Computer Applications, 38*(5), 34-38.

Pacini, E., Mateos, C., & Garino, C. G. (2015). Balancing throughput and response time in online scientific Clouds via Ant Colony Optimization (SP2013/2013/00006). *Advances in Engineering Software, 84*, 31-47.

Pandey, S., Wu, L., Guru, S. M., & Buyya, R. (2010). *A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments.* Paper presented at the 2010 24th IEEE international conference on advanced information networking and applications. https://doi.org/10.1109/AINA.2010.31

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. DOI: 10.1109/ACCESS.2019.2946216

Raghavan, S., Sarwesh, P., Marimuthu, C., & Chandrasekaran, K. (2015). *Bat algorithm for scheduling workflow applications in cloud.* Paper presented at the 2015 International Conference on Electronic Design, Computer Networks & Automated Verification (EDCAV). DOI: 10.1109/EDCAV.2015.7060555

Raj, R. J. S., & Prasad, S. M. (2016). *Survey on variants of heuristic algorithms for scheduling workflow of tasks.* Paper presented at the 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT). DOI: 10.1109/ICCPCT.2016.7530288

Saleh, H., Nashaat, H., Saber, W., & Harb, H. M. (2018). IPSO task scheduling algorithm for large scale data in cloud computing environment. *IEEE Access, 7*, 5412-5420. DOI: 10.5772/intechopen.86873

Sanaj, M., & Prathap, P. J. (2020). *An enhanced Round robin (ERR) algorithm for effective and efficient task scheduling in cloud environment.* Paper presented at the 2020 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA). DOI: 10.1109/ACCTHPA49271.2020.9213198

Sangwan, P., Sharma, M., & Kumar, A. (2017). Improved round robin scheduling in cloud computing. *Adv. Comput. Sci. Technol, 10*(4), 639-644.

Sharma, N., Tyagi, D. S., & Atri, S. (2017). HYMM: A New Heuristic in Cloud Computing. *International Research Journal of Engineering and Technology (IRJET), 4*(5), 3520-3526.

Shi, Y., Suo, K., Kemp, S., & Hodge, J. (2020). *A Task Scheduling Approach for Cloud Resource Management.* Paper presented at the 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4). DOI: 10.1109/WorldS450073.2020.9210422

Singh, P., Dutta, M., & Aggarwal, N. (2017). A review of task scheduling based on meta-heuristics approach in cloud computing. *Knowledge and Information Systems, 52*(1), 1-51. https://doi.org/10.1007/s10115-017-1044-2

Singhal, S., & Sharma, A. (2021). A job scheduling algorithm based on rock hyrax optimization in cloud computing. *Computing, 103*(9), 2115-2142. https://doi.org/10.1007/s00607-021-00942-w

Son, S., & Jun, S. C. (2013). *Negotiation-based flexible SLA establishment with SLA-driven resource allocation in cloud computing.* Paper presented at the 2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing. DOI: 10.1109/CCGrid.2013.81

Sörensen, K., & Glover, F. (2013). Metaheuristics. *Encyclopedia of operations research and management science, 62*, 960-970.

Stergiou, C., Psannis, K. E., Kim, B.-G., & Gupta, B. (2018). Secure integration of IoT and cloud computing. *Future Generation Computer Systems, 78*, 964-975.

Thakur, P., & Mahajan, M. (2017). Different scheduling algorithm in cloud computing: a survey. *International Journal of modern computer science, 5*(1).

Tsai, C.-W., & Rodrigues, J. J. (2013). Metaheuristic scheduling for cloud: A survey. *IEEE Systems Journal, 8*(1), 279-291. DOI: 10.1109/JSYST.2013.2256731

Wei, X. J., Bei, W., & Jun, L. (2017). *SAMPGA task scheduling algorithm in cloud computing.* Paper presented at the 2017 36th Chinese Control Conference (CCC).

Xu, F., Liu, F., Jin, H., & Vasilakos, A. V. (2013). Managing performance overhead of virtual machines in cloud computing: A survey, state of the art, and future directions. *Proceedings of the IEEE, 102*(1), 11-31. DOI: 10.1109/JPROC.2013.2287711

Yang, X.-S. (2010). *Nature-inspired metaheuristic algorithms*: Luniver press.

Yang, X.-S., Cui, Z., Xiao, R., Gandomi, A. H., & Karamanoglu, M. (2013). *Swarm intelligence and bio-inspired computation: theory and applications*: Newnes.

Yao, H., Fu, X., Li, H., Dong, G., & Li, J. (2017). Cloud task scheduling algorithm based on improved genetic algorithm. *International journal of performability engineering, 13*(7), 1070. doi: 10.23940/ijpe.17.07.p9.10701076

Yiqiu, F., Xia, X., & Junwei, G. (2019). *Cloud computing task scheduling algorithm based on improved genetic algorithm.* Paper presented at the 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). DOI: 10.1109/ITNEC.2019.8728996

Yu, J., & Buyya, R. (2006). Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms. *Scientific Programming, 14*(3-4), 217-230. https://doi.org/10.1155/2006/271608

Yu, J., Buyya, R., & Ramamohanarao, K. (2008). Workflow scheduling algorithms for grid computing. In *Metaheuristics for scheduling in distributed computing environments* (pp. 173-214): Springer. https://doi.org/10.1007/978-3-540-69277-5_7

Zhan, Z.-H., Liu, X.-F., Gong, Y.-J., Zhang, J., Chung, H. S.-H., & Li, Y. (2015). Cloud computing resource scheduling and a survey of its evolutionary approaches. *ACM Computing Surveys (CSUR), 47*(4), 1-33. https://doi.org/10.1145/27883

Zhang, A.-N., Chu, S.-C., Song, P.-C., Wang, H., & Pan, J.-S. (2022). Task Scheduling in Cloud Computing Environment Using Advanced Phasmatodea Population Evolution Algorithms. *Electronics, 11*(9), 1451. https://doi.org/10.3390/electronics11091451

Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications, 1*(1), 7-18. https://doi.org/10.1007/s13174-010-0007-6

Zong, Z. (2020). *An improvement of task scheduling algorithms for green cloud computing.* Paper presented at the 2020 15th International Conference on Computer Science & Education (ICCSE). DOI: 10.1109/ICCSE49874.2020.9201785

**Anupam Yadav**

Anupam Yadav is an Assistant Professor in the Department of Computer Engineering and Applications at GLA University, Mathura. Her primary research areas include Cloud Computing and Machine Learning. With a strong academic and research background, she has contributed extensively to the field, publishing over 35 research papers in peer-reviewed Web of Science (WoS) journals. Her work reflects a commitment to advancing knowledge and innovation in her areas of expertise.

Iranian Journal of Information Processing and Management

**Ashish Sharma**

Prof. Ashish Sharma is currently serving as the Dean of Academics at GLA University, Mathura, India. His research interests encompass Cloud Computing, Machine Learning, and Software Engineering. With a distinguished academic career, Prof. Sharma has published over 50 research papers in peer-reviewed journals indexed in the Web of Science. His contributions to these domains reflect his dedication to advancing knowledge and technology.

Iranian Journal of
Information
Processing and
Management