Metaheuristic Approaches to Task Scheduling in Cloud Computing: A Comprehensive Review

Manpreet Kaur
PG Department of Computer Science, Mata Gujri College, Fatehgarh Sahib

Abstract: Cloud computing provides computer resources such as software and hardware as a service to users via a network. Because of the size of modern data centers and their dynamic resource provisioning nature, we require effective scheduling solutions to manage these resources. The primary goal of scheduling is to assign assignments to the appropriate resources in order to meet one or more optimisation criteria. Scheduling is a difficult problem in the cloud environment; hence many researchers have tried to find an effective solution for job scheduling in the cloud. Through a network, cloud computing provides customers with computing resources, such as hardware and software, as a service. We require effective scheduling strategies to handle these resources because of the size of contemporary data centres and their dynamic resource providing nature. Assigning tasks to sufficient resources in order to meet one or more optimisation criteria is the primary goal of scheduling. Since scheduling is a difficult problem in the cloud, numerous researchers have tried to investigate the best way to schedule tasks in the cloud. This paper will thoroughly review the task scheduling in cloud computing environment based on various meta – heuristic techniques.

Keywords: Cloud computing, Resource scheduling, Optimization criteria, Scheduling, Task scheduling, Meta-heuristic techniques.

1. INTRODUCTION

Cloud computing represents a significant advancement in distributed computing, offering users numerous benefits. It provides high reliability, security, scalability, cost-effectiveness, enhanced collaboration, and easy access to various applications and resources [1]. It is a concept for enabling suitable, on-demand provisioning of computing resources such as software, hardware, applications, and services that can be rapidly provisioned and freed with the least management overhead or intervention from service providers [2]. Software-as-a-Service (SaaS), Infrastructure-as-a-Service (IaaS), and Platform-as-a-Service (PaaS) are the three main service model types that cloud computing delivers [3]. Public, private, community, and hybrid clouds are the four primary development models for cloud computing, which can be implemented as a tiered architecture [4]. Virtualisation is the key idea of cloud computing. Through the use of virtualisation, users can quickly access computer resources without having to worry about the intricacies and internal workings of the system [5]. On actual servers, it allows users to build Virtual Machines (VMs) [6]. It improves the use of physical resources in cloud computing and lowers the amount of hardware equipment needed. Cloud computing offers both cloud consumers and service providers a number of benefits, some of which are detailed in [5, 7–10] and are enumerated below.

- lowering the price by offering pay-as-you-go computing resources on demand.
- Effective resource allocation and deallocation prevents capital expenditures from squandering unused resources.
- Offering significant speed and flexibility when scaling the infrastructure up or down.
- Distributing the backups around data centers in different places in order to provide an efficient recovery and backup.
- Enabling convenient access at any time and from any location in the world.

Notwithstanding the many benefits of cloud computing environments, a few major problems have affected their effectiveness and dependability [11]. Numerous problems with cloud computing have drawn the interest and concern of scholars. Generally speaking, the primary problems in cloud systems can be divided into seven key categories: resource management, load balancing, cloud migration, privacy and security, scalability and availability, energy efficiency, compatibility and interoperability [12-15]. In the cloud computing context, scheduling allows for the efficient execution of numerous tasks on the pool of available computing resources. Numerous optimisation factors, including dependability, makespan, load balancing, execution cost, budget, and utilisation, are crucial to this procedure [16]. In the task scheduling process, users submit tasks to the cloud scheduler, which then checks the resources' status from the cloud information service. Afterwards, assigning the assignments to different resources according to their needs [17]. The effective scheduler

optimally allocates the right resources (such as virtual machines) to the tasks.

In the cloud computing environment, work allocation on seemingly limitless computer resources is typically a nondeterministic polynomial time (NP)-hard problem. Numerous academics have tried to investigate the best polynomially-time method for job scheduling in cloud environments. An optimal polynomial-time solution to this problem has not been introduced by any one technique. In order to find near-optimal or optimal solutions to these complicated issues, meta-heuristic-based techniques have been employed. Many meta-heuristic techniques, including genetic algorithms (GA), particle swarm optimisation (PSO), ant colony optimisation (ACO), tabu search (TS), simulated annealing (SA), bat algorithm (BA), and memetic algorithm (MA), have been introduced and have been quite popular in recent years [18].

To develop an effective scheduling algorithm, we need to have a solid grasp of resource management and the many problems associated with different scheduling techniques. Therefore, the purpose of this study is to provide a comparative assessment of various task scheduling systems and to describe the main concepts of resource scheduling. Using optimisation criteria suitable for cloud computing environments, a systematic analysis of cloud computing job scheduling is presented. This research will help academics decide on the best approach to suggest a suitable way to schedule user applications in a cloud environment. Only cloud computing scheduling issues are examined in this article; distributed systems as a whole are not. Five sections make up the remainder of the paper. In part 2, we introduce cloud computing resource management. Section 3 discusses scheduling. Section 4 presents the discussion. Section 5 concludes with a summary of future work remarks and a conclusion.

2. MOTIVATION FOR THE STUDY

Cloud computing offers computing resources such as hardware and software as a service through a network, making efficient resource management is a critical challenge. Due to the large-scale data processing nature of modern data centers and their dynamic resource provisioning, effective scheduling solutions are required to optimize performance and resource utilization. Task scheduling plays a crucial role in ensuring that computational tasks are allocated to appropriate resources while meeting optimization criteria such as time efficiency and cost-effectiveness. Given the complexity of scheduling in cloud environments, extensive research has been conducted to find effective solutions. This study aims to provide a comprehensive review of task scheduling in cloud computing, focusing on meta-heuristic techniques to address the challenges associated with resource allocation. The primary aim of this study is to review and analyze task scheduling strategies in cloud computing environments using various meta-heuristic techniques. It seeks to evaluate the effectiveness of these techniques in optimizing resource utilization, reducing execution time, and improving overall system performance.

3. RESOURCE MANAGEMENT IN CLOUD COMPUTING

In distributed computing, like cloud computing, resource management is a significant difficulty [19]. Depending on their evolving needs, different cloud users need different services. Therefore, the goal of cloud computing is to provide all necessary services. However, cloud service providers find it challenging to deliver all of the necessary services on time because of the limited resources at their disposal. The distributed style of virtualisation technology used in cloud computing makes it simple to add dynamically new resources, something that previously challenging with old resource management techniques [20]. The type of resources and the difficulties in managing them in a cloud computing environment are covered in the next section.

3.1 Types of Resources

The classifications of the primary resource categories according to their services—such as energy, storage, compute, networks, and security—are briefly introduced in the section that follows. The many kinds of cloud computing resources are compiled in Table 1.

a. *Storage services:* Over time, computer systems are prone to malfunction. Therefore, continuity is necessary for the business or people to manage and preserve backups. In essence, Storage as a Service (StaaS) is a solution that enables cloud data storage. It consists of hard drives and thousands of database servers [21]. StaaS lowers hardware and space costs, lowers disaster recovery risks, and offers long-term data preservation. Consequently, SaaS improves work continuity and availability [22].
b. *Computation services:* In a cloud computing context, computation as a service, or CaaS, is a quick computational service. It encompasses the operating system, memory capacity, computing power, and effective algorithms [22].

c. *Network services:* Network as a Service (NaaS) includes logical resources like protocols, throughput, bandwidth, delay, loads, and virtual network links, as well as physical resources like physical network lines, sensors, workstations, and intermediary devices [23]. It is impossible to imagine computing and storage services without network services like latency and bandwidth. Since all cloud computing services are delivered via fast Internet, they are the most important services from a network perspective [24].
d. *Security services:* One of the major issues in the cloud computing context is security as a service, or SECaaS [25]. SECaaS offers consumers enhanced security against online threats and attacks [26]. Authentication, trust, intrusion detection, penetration testing, anti-malware, anti-virus, and security event management are among the services it offers [27].
e. *Energy services:* Cloud data centres have extremely high energy consumption. Physical resources like cooling units and uninterruptible power supplies (UPS) make up energy services. Numerous energy-saving methods have been developed to control wasteful resources and lower expenses. By using energy-saving methods on servers and networks, data centres can save a significant amount of energy [28].

3.2 Resource Management Challenges in Cloud

Resource allocation, resource provisioning, resource mapping, resource discovery and selection, resource adaptability, resource brokering, and resource scheduling are the significant issues that are frequently linked to resource management in cloud systems. We'll quickly go over the fundamental idea behind these challenges:

The efficient distribution of cloud resources across various applications via the internet is known as resource allocation [29].

- *Resource provisioning*: Allocating the service provider's resources to cloud users with service quality assurance, as specified in the service level agreement (SLA), is known as resource provisioning. Dynamic and static resource provisioning are the two categories into which it falls [30].
- *Resource mapping:* It's the alignment of resources needed by cloud users with those offered by a service provider [31].
- *Resource discovery and selection*: It is the process of finding every resource that is available in the system, gathering data on the resources' current condition, and then deciding the target resource to choose based on the information gleaned from the discovery [32].
- *Resource adaptation*: It is the system's ability to dynamically modify resources to satisfy user needs.
- *Resource brokering*: is the practice of using an agent to negotiate for the resources needed in order to ensure that they are available in time to meet the goals [32].
- *Resource scheduling:* described by [36] as a schedule of events and resources that documents the start and end times of an activity based on its (1) duration, (2) predecessor activities, (3) previous relationships, and (4) resources allotted.

Table 1. Type of Resources

Storage	Hard drives Database
Computation	Memory. Processing. Algorithms. Operating System.
Network	Physical: Network link. Workstations. Sensors. Intermediate devices Logical: Virtual network link. Protocols. Throughput. Bandwidth. Delay. Loads.
Security	Authentication. Trust. Privacy. Anti-malware. Anti-virus. Intrusion detection. Penetration testing.
Energy	Cooling devices UPS. Energy saving technique.

4. SCHEDULING

In order to accomplish high-performance computing and a desired level of service, scheduling's primary goal is to allocate resources to specific activities in the shortest amount of time. In order to optimise one or more optimisation criteria, the scheduling must arrange the assigned activities according to the resources that are available, subject to specific limitations [33]. Scheduling is in charge of choosing the right resources for task execution in distributed computing systems while taking into account the parameters of both static and dynamic tasks [34]. Depending on the type of jobs in the application,

different scheduling techniques are used. Only after all of its scheduling has been completed can a job with a sequence be scheduled. Another situation is independent task scheduling, which allows jobs to be planned in any sequence when they are unrelated to one another [35].

4.1 Scheduling Procedure

In cloud computing, scheduling refers to the process of allocating tasks to resources in a manner that optimizes performance metrics such as execution time, resource utilization, and cost. Effective scheduling ensures that computational tasks are executed efficiently, leveraging the dynamic and scalable nature of cloud environments.

a. Task Analysis: Evaluate the incoming tasks to determine their specific requirements, including computational power, memory, storage, and any dependencies.
b. Resource Discovery: Identify available resources within the cloud infrastructure that match the task requirements. This involves querying the resource pool to assess current availability and capabilities.
c. Selection of Scheduling Algorithm: Choose an appropriate scheduling algorithm based on the nature of the tasks and desired optimization criteria. Common algorithms include:
    1. Heuristic Algorithms: Such as the Heterogeneous Earliest Finish Time (HEFT) algorithm, which prioritizes tasks based on their computational requirements and schedules them to minimize completion time.
    2. Metaheuristic Algorithms: Techniques like Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) are employed to find near-optimal solutions for complex scheduling problems. [22]
d. Task Prioritization: Assign priorities to tasks based on factors such as deadlines, resource demands, and quality of service requirements.
    1. Resource Allocation: Allocate tasks to selected resources, ensuring that the assignment aligns with the optimization goals and respects any constraints.
    2. Execution Monitoring: Continuously monitor the execution of tasks to ensure they are proceeding as scheduled. This includes tracking performance metrics and resource utilization.
    3. Dynamic Adjustment: In response to changing conditions, such as resource availability fluctuations or task execution delays, adjust the schedule dynamically to maintain optimal performance. [23]

4.2 Cloud Resource Scheduling Layers

In cloud computing, resource scheduling is a critical process that ensures efficient allocation and management of resources to meet diverse user demands. This process is typically structured across multiple layers, each focusing on specific aspects of resource management. [20]

1. Application Layer:

At the topmost level, the application layer handles user interactions and manages service requests. Scheduling at this layer involves allocating resources to various applications based on user demands and predefined policies. The primary goal is to ensure that applications receive the necessary resources to function optimally, thereby enhancing user experience. [29]

2. Deployment Layer:

Situated between the application and virtualization layers, the deployment layer focuses on the efficient placement and management of virtual machines (VMs) and services. Scheduling at this level involves decisions about where and how applications and services should be deployed within the cloud infrastructure to optimize performance, resource utilization, and energy efficiency. [32]

3. Virtualization Layer:

The virtualization layer abstracts physical hardware resources into virtual instances, enabling flexible and scalable resource allocation. Scheduling at this layer involves managing the distribution of virtual resources over physical hardware, ensuring optimal utilization and performance. This includes tasks such as VM placement, load balancing, and resource isolation to maintain quality of service (QoS) and system stability. [42]

By effectively coordinating scheduling activities across these layers, cloud computing systems can achieve high efficiency, scalability, and responsiveness, thereby meeting the dynamic needs of users and applications. [43]

4.3 Task – Resource Scheduling Problem Formulation

Task scheduling optimisation in cloud computing specifies the ideal amount of necessary systems in order to minimise overall costs. assuming that there are n tasks that should be handled on m available computational resources and that each task's execution time on each processing machine is known. The objective is to reduce the overall execution time and

optimise the use of the resources that are available. Assume that there are more tasks than resources (n > m) and that tasks cannot move across resources [44]. The collection of tasks defined as Ti={1,2,…n}, where n is the number of independent tasks and Rj={1,2,…m}, where m is the number of computational resources, is used to frame the problem. Thus, obtaining an optimal mapping (OM) of tasks (Ti) to resources (Rj) OM: Ti◊Rj is the cloud resource scheduling problem. Figure 2 illustrates this dilemma, which is defined as when two or more jobs share a single resource [45].

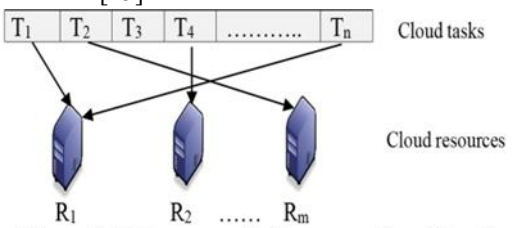

Figure 1: Cloud Resource Scheduling Problem

4.4 Optimization Criteria

The metrics used to assess scheduling effectiveness are described in this section. Numerous optimisation criteria, including makespan, cost, budget, deadline, resource utilisation, throughput, load balancing, and energy efficiency, have been covered in the previous studies. According to cloud service, these optimisation criteria are typically divided into two desires: those of cloud service providers and those of cloud consumers, as shown in figure 3 [47]. Since the majority of the reviewed works address these optimisation criteria, this study attempts to show how these criteria are examined using a comparative method.

3.4.1 User Desire Criteria

- *Makespan Time / Completion Time:* The time it takes to finish the final operation needed to exit the cloud system is known as makespan [18].
- *Cost:* Cost is the sum of money a customer pays a service provider based on how many resources they consume [19].
- *Budget*: It shows the limitations on doing the activities within the allocated budget [10].
- *Deadline:* It signifies the end of active tasks at a specific moment [11].

3.4.2 Provider Desire Criteria

- *Resource Utilization:* maximising the use of existing resources and maintaining their maximum level of activity. On-demand leasing of limited resources to cloud users is beneficial for service providers [12].
- *Throughput:* It calculates how many jobs are finished in a certain amount of time [13]
- *Load Balancing:* In cloud computing, load balancing refers to the equitable distribution of loads among virtual machines (VMs) across physical resources. In [14–16], the authors introduced a number of approaches.
- *Energy Efficiency:* Reducing the amount of energy used by a job is known as energy efficiency [17].

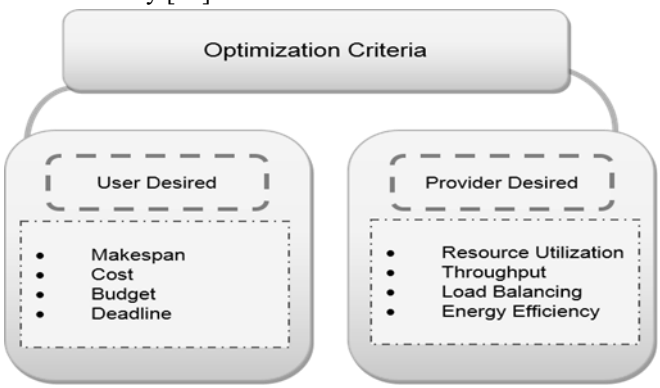

Figure: 2 Optimization Criteria

Discrete and continuous optimisation issues are the two categories. For a combinatorial problem, the choice variables have discrete values, but for a continuous optimisation problem, they can have values inside the domain of real values (Ri) [18,19]. Depending on the amount of criteria in the optimisation problem, this can be divided into single-criteria and multicriteria. The objective of single-criterion optimisation is to identify the optimal solution based on a single criterion function. Finding one or more optimal solutions for each criterion is the challenge at hand when the optimisation problem comprises many criteria functions. In this case, a solution that meets one requirement well may not meet another, and vice versa [10]. Finding a collection of solutions that are optimal in terms of every other criterion is, thus, the aim of multi-criteria optimisation. The majority of real-world issues are evidently multi-criteria. These days, there are optimisation methods that use heuristic-based and meta-heuristic search strategies to get answers. These methods use both deterministic and stochastic search concepts. We can state that an algorithm is capable of solving a problem if it is able to solve every instance of problem (P). Typically, we want to know which method works best for the situation. Efficiency is typically associated with the amount of computer resources (time and space) used to execute a procedure [11, 12]. Generally speaking, the method that solves the problem the quickest is the most effective. The

effective time required to solve the problem on a physical computer is not a reliable indicator of algorithm time complexity in practice due to its lack of criteria. various hardware configurations or even various operating systems may be able to execute the same algorithm. Consequently, the complexity of the method is assessed informally by calculating the complexity in relation to the quantity of input data required for the problem description. The impact of increasing the instance size on an algorithm's time complexity is determined by its time complexity. The so-called asymptotic time complexity function O(f^(n)), which establishes the upper bound of time complexity for problem P, can be used to explain this relationship. For instance, the function O(n2) indicates that the temporal complexity will rise to almost n2 as the instance size n increases. According to the asymptotic time complexity function, the algorithmic theory categorises problems into two groups: NP-hard and P-hard. Problems that exhibit the exponential time complexity O(2n) are categorised as "complicated" in the first class. In other words, an exponential rise in the input data may result in an exponential increase in the problem's solution time due to the exponential time complexity. In the worst scenario, we might have to wait an endless amount of time for the answer. On the other hand, class P-hard problems are regarded as "simple" and have a polynomial time complexity of O(nk). [12].

3.5 Task Scheduling Techniques

Task scheduling in cloud computing is essential for optimizing resource allocation, improving efficiency, and ensuring high system performance. Various scheduling techniques are used to distribute workloads efficiently among cloud resources. Below are the key task scheduling techniques:

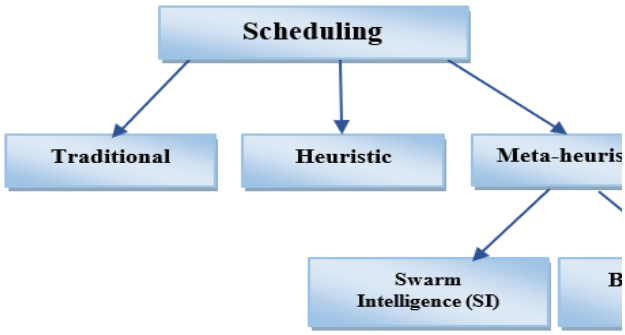

Figure 3: Task Scheduling Techniques

4.5.1 Traditional Techniques: Conventional methods, like Round Robin (RR), First Come First Serve (FCFS), and Shortest Job First (SJF), are crucial for scheduling various activities [13]. These methods yield precise results and are straightforward, quick, and deterministic [14]. However, in many cases, they are ineffective at comprehending the optimality problem [15]. Therefore, it is not possible to schedule in a cloud environment using traditional methods [16]. Numerous efforts have been made to enhance the application of the conventional methods [13, 17-20]. One of these methods that uses a time slice or a quantum is round robin. One disadvantage of the RR algorithm is that it makes use of static time quantum [17]. Round-robin scheduling is the foundation of the suggested CPU scheduling in [18], albeit the method of scheduling computations is altered. Instead of providing static time quantum in the CPU scheduling, it drastically reduces the waiting time and turnaround time as compared to the simple RR scheduling. According to the FCFS algorithm, the first task will be completed. In order to maximise resource utilisation and reduce job execution time, researchers in [19] suggested a task scheduling method based on fuzzy clustering techniques. SJF is a scheduling method that is dependent on how long the task will take to complete. Priority is used to queue the jobs; the tasks with the lowest priority and the longest duration are arranged last and first, respectively [21]. The task with the shortest burst time is given the CPU in this algorithm. The SRDQ method is a hybrid algorithm of RR and SJF that was proposed by Elmougy et al. in [20]. Quantum time is a dynamic variable that is taken into account by this approach.

4.5.2 Heuristic Techniques: To discover the best or nearly best answer, these methods use a sample space of random solutions [26]. There are numerous heuristic methods, including enhanced max-min, max-min, priority-based min-min, and min-min [22]. Although these methods produce better outcomes than previous methods, they do not ensure a high ranking in cloud scheduling [23]. The issue of local minima frequently traps the solutions produced by heuristic approaches [26]. In [24], an enhanced Max-min method is suggested that uses the projected execution time as the selection basis rather than the completion time. A task with an average execution time is assigned. The algorithm makes it more likely that jobs will be assigned to resources synchronously. The fundamental Min-Min algorithm is a simple and efficient method that produces the optimal scheduling in terms of cutting down on job completion time. The largest disadvantage, though, is load balancing, which is seen to be one of the main issues facing cloud service providers. By introducing the Load Balance Improved Min-Min (LBIMM) algorithm, the authors

in [25] have enhanced load balancing. The Min-Min algorithm serves as the foundation for the LBIMM method, which aims to reduce completion time and maximise resource utilisation.

4.5.3 Meta-Heuristic Techniques: Influenced by insects' social behaviour [26]. In 1986, Fred Glover coined the term "meta-heuristic," where "meta" means higher level and "heuristic" means to learn by trial and error. We used the precise definition of the term "metaheuristic" from [27], which states that it is a high-level algorithmic framework that is independent of problems and offers a collection of rules or techniques for creating heuristic optimisation algorithms. A problem-specific implementation of a heuristic optimisation algorithm in accordance with the rules outlined in such a framework is also referred to by this term. Intensification and diversity are the two primary components of all meta-heuristic approaches.

There are two types of meta-heuristic methods: bio-inspired and swarm intelligence (SI). Nearly every branch of research, data mining, biomedical engineering, control systems, and parallel computing has been impacted by bio-inspired design. Numerous bio-inspired algorithms exist, including imperative competitive algorithm (ICA), GA, and MA. Inspired by the social behaviour of insect colonies and other animals, such as PSO, ACO, artificial bee colonies (ABC), glowworm swarm algorithm (GSA), BA, firefly algorithm (FA), cuckoo search (CS), and cat swarm optimisation (CSO), swarm intelligence is a relatively new technique to solve unconstrained optimisation problems. Better algorithms are constantly being sought for by researchers, particularly for cloud computing work scheduling. Here, we compare these methods using a variety of optimisation criteria that support the search space's intensification. In order to solve the local minima problem, researchers in [30] proposed an algorithm for independent task scheduling in grid computing by combining PSO with the gravitational emulation local search (GELS). The Makespan time is significantly reduced by the amalgamation PSO–GELS algorithm. A hyper-heuristic approach for scheduling secure jobs in a grid setting served as the foundation for a new PSO algorithm that was introduced in [31]. Both Makespan and cost are decreased by the hyper-heuristic algorithm. The authors of [32] present a task scheduling method for load distribution across virtual machines (VMs), energy reduction, and makespan time minimisation that is based on the double-fitness adaptive algorithm-job spanning time and load balancing genetic algorithm (JLGA). This algorithm initialises the population using a greedy technique. Instead of using a fixed value, it uses crossover and mutation to determine adaptive probabilities. The authors of [33] suggest a hybrid PSO (HPSO), which combines the TS and PSO algorithms. By using Tabu Search, HPSO offers a local search method. By splitting the randomly produced population into two equal portions, HPSO improves it. PSO is used to improve part one, and TS is used to improve part two. The particles' best local and global positions are then exchanged by combining them once more into a single section. HPSO maximises resource utilisation and reduces makespan. Raghavan et al. [34] used the Bat method to tackle the cloud workflow scheduling problem, which produces better cost processing outcomes than the Best Resource Selection (BRS) approach. In order to increase the search space's intensity, a combination of the PSO and CS algorithms is introduced in [38]. For autonomous task scheduling in cloud computing, the hybrid PSOCS algorithm reduces the makespan and achieves optimal resource utilisation. After every iteration, the authors in [35] use the hill climbing algorithm to improve local search capability and decrease the PSO precocious convergence. The hybrid GHPSO method uses a genetic algorithm's crossover and mutation strategies to solve discrete problems. GHPSO is employed to reduce expenses. To reduce the execution cost of executing workflow applications on the cloud, researchers in [36] used PSO. Whereas the suggested technique in [37] creates the initial population of particles using the shortest job to fastest processor (SJFP) algorithm, PSO creates the initial population at random. PSO and the tabu search mechanism (PSOTBM) were combined by researchers in [88] to provide autonomous job scheduling in cloud computing. An impressive 67.5% reduction in energy usage is demonstrated by the merger PSOTBM. In order to improve resource utilisation, a novel method was introduced in [39] that assigns virtual machines (VMs) to the appropriate physical machines using a family genetic algorithm (FGA).

CSO and GA algorithms are combined to create CSO-GA [40]. Comparing this hybrid algorithm to other scheduling methods, the makespan is optimised. By balancing load by searching under loaded nodes, the researchers in [44] have put forth a unique algorithm based on ant colonies that reduces reaction time. This approach assigns jobs to virtual machines (VMs) using FCFS. The author in [41] employed tree representation for GA solutions for mapping virtual machines and physical machines in order to generate optimal

solutions for the grid scheduling problem. In their description of optimising energy savings and maximising profits for service providers, the authors [42] introduced a multi-metric evolutionary algorithm for scheduling independent jobs, including makespan, cost, and energy efficiency. By developing a technique known as MHPSO, researchers in [43] improve the convergence rate and decrease the computing time of PSO. MHPSO is a hybrid of the standard hierarchical PSO algorithm (HPSO) and the mutation concept based PSO algorithm (MPSO). The authors in [44] introduced a unique power-aware load balancing technique dubbed imperialism competitive algorithm-minimum migration time (ICA-MMT) to balance the load and optimise resource utilisation over hosts on data centres. Data centres for cloud computing use less energy because to this technique. Parallel bee colony optimisation particle swarm optimisation (PBCOPSO) is the name of the suggested method that was developed in [45] by authors that merged bee colony and PSO algorithms. When it comes to maximising resource utilisation and minimising makespan, PBCOPSO exhibits a notable improvement. In [45], a new load balancing technique based on a genetic algorithm that addresses the scheduling of independent jobs is presented. This method offers effective resource utilisation and load balancing. In order to maximise resource utilisation and finish the jobs in the shortest amount of time, the population of particles is initialised in [46]. In [47], the authors present FUGE, a hybrid technique that combines fuzzy theory and GA to achieve optimal load balancing while taking execution time and cost into account. The authors in [48] made a contribution by combining the imperialist competitive and local search optimisation techniques. This method tackles both makespan and reliability issues. This algorithm performed better when compared to genetic and ant colony optimisation algorithms. In order to balance load among virtual machines (VMs) and minimise dynamic VM migration, the load balancing approach [49], which is based on evolutionary algorithms, was presented in cloud computing environments. article swarm optimisation has been used in numerous additional works, including [46,47,48, 50], to address the job scheduling issue. The authors in [53] addressed the issue of reaction time and balancing throughput on a private cloud when multiple cloud users are carrying out their experiments by describing and evaluating a cloud scheduler based on ACO.

In [52], the hybrid algorithm GA-PSO is introduced; it chooses virtual machines (VMs) according to job workflow and speed. This approach lowers costs and makespan while improving load balancing. Using a modified genetic algorithm called the family genetic algorithm, Kamaljit et al. [53] presented a novel context-and load-aware family genetic algorithm approach for effective job scheduling. In order to solve the workflow scheduling problem in cloud computing with the goal of minimising the makespan, researchers in [54] suggested an algorithm based on the bat algorithm (BA). They used MATLAB to develop the BA and compared the outcomes with those of two well-known existing algorithms, CSO and PSO. A task scheduling technique based on a modified GA has been proposed by S. A. Hamad and F. A. Omara [53]. By employing the tournament selection approach to choose the finest chromosomes, they get around the population size restriction. A static task scheduling method based on the PSO algorithm was presented by researchers in [57]. They enhanced PSO by reducing makespan and increasing resource utilisation with the use of the honeybee load balancing technique. PSO and hill climbing algorithms are used in [58]'s hybrid task scheduling approach.

This algorithm uses PSO to randomly distribute the initialisation of a population. Next, a few particles are chosen to be used for hill climbing. The makespan is optimised by this method. The PSO and GA algorithms are combined in [35]'s priority-based job scheduling system, known as the HGPSO method. Prior to applying the HGPSO algorithm, the jobs in HGPSO are sorted according to a priority queue. The HGPSO outperforms particle swarm and genetic optimisation methods in terms of availability, scalability, and completion time. In order to improve resource utilisation and reduce makespan, researchers in [37] introduced an HTSCC Algorithm that combines the advantages of GA and PSO algorithms. The CloudSim simulator is used to implement and simulate the HTSCC algorithm. According to the simulation results, the HTSCC algorithm works better than the GA and PSO algorithms by using more resources and reducing makespan. In order to build a task scheduling model and solve a global optimisation problem, the researchers in [40] introduced the MSDE method, which is based on enhancing the performance of the Moth Search method (MSA) utilising differential evolution (DE).

The researchers in [41] suggested a hybrid shortest–longest scheduling technique to address the starving problem. To solve the starving issue and satisfy provider and user criteria, they assigned the tasks to the most convenient virtual machines (VMs) based on the characteristics of each VM and the duration of the job. [42] introduces a new hybrid QoS-based task

scheduling algorithm for scheduling independent and dependent jobs in a cloud context. This work can be expanded to effectively execute the hybrid task scheduling algorithm by utilising energy efficiency and communication costs. Tabular analysis of all the scheduling techniques is as follows:

Table 2: Traditional Task Scheduling Techniques in Cloud Computing

Scheduling Technique	Description	Key Features	Limitations	Scope of Improvement
Round Robin (RR)	Assigns tasks cyclically to available resources using a fixed time quantum.	Simple, fair, avoids starvation.	Static time quantum leads to inefficiency.	Dynamic time quantum adjustment (e.g., SRDQ).
First Come First Serve (FCFS)	Tasks are executed in the order they arrive.	Simple, deterministic, no starvation.	Long tasks delay shorter ones (Convoy Effect).	Fuzzy clustering-based FCFS for optimization.
Shortest Job First (SJF)	Prioritizes tasks with the shortest execution time.	Minimizes waiting time, improves throughput.	Starvation of longer tasks, requires task time estimation.	Hybrid RR-SJF (SRDQ) for dynamic quantum adjustment.
Min-Min	Selects the shortest task first and assigns it to the fastest available resource.	Minimizes job completion time.	Poor load balancing, longer tasks suffer.	Load Balance Improved Min-Min (LBIMM).
Max-Min	Assigns the longest task to the fastest available resource.	Balances load better than Min-Min.	Smaller tasks may suffer.	Enhanced Max-Min (Projected execution time instead of completion time).

Table 3: Heuristic and Metaheuristic Task Scheduling Techniques in Cloud Computing

Scheduling Technique	Description	Key Features	Limitations	Scope of Improvement
Genetic Algorithm (GA)	Evolutionary-based scheduling that mimics natural selection.	Provides near-optimal scheduling solutions.	High computational complexity.	Hybrid GA-PSO, Fuzzy-based GA.
Particle Swarm Optimization (PSO)	Inspired by the movement of bird flocks; particles adjust positions based on best-known solutions.	Fast convergence, good load balancing.	May get trapped in local optima.	Hybrid PSO-GA, PSO-Hill Climbing, PSO-TBM.
Ant Colony Optimization (ACO)	Inspired by ants' pheromone-based pathfinding to optimize task scheduling.	Good for dynamic task allocation.	Slower convergence compared to PSO.	ACO with improved pheromone update rules.
Artificial Bee Colony (ABC)	Mimics bee foraging behaviour to find optimal scheduling solutions.	Balances exploration and exploitation well.	May require fine-tuning of parameters.	Hybrid ABC-PSO, ABC-GA.
Bat Algorithm (BA)	Based on echolocation of bats; finds optimal task assignments.	Good for large-scale scheduling.	Tuning parameters can be complex.	Hybrid BA-PSO, BA-GA.
Cuckoo Search (CS)	Inspired by the brood parasitism of cuckoo birds.	Good exploration ability, avoids local optima.	Convergence speed can be slow.	Hybrid CS-PSO, CS-GA.
Firefly Algorithm (FA)	Uses firefly luminescence behaviour to optimize scheduling.	Efficient for large datasets.	Needs parameter tuning for better performance.	Hybrid FA-PSO.
Hybrid PSO-TS (HPSO-TS)	Combines PSO and Tabu Search for improved local search.	Maximizes resource utilization.	Increased computational overhead.	Adaptive learning in hybrid PSO-TS.

Imperialist Competitive Algorithm (ICA)	Simulates imperialist competition for task scheduling.	Effective in complex environments.	High processing time.	ICA-MMT (Minimum Migration Time) for power-aware scheduling.

5. CONCLUSION & FUTURE WORK

The primary ideas of scheduling and resource management in cloud computing were presented in this work. Furthermore, taking into account the simulation environment, job types, user and provider preferences, and optimisation goals, we offered a comparative study of meta-heuristic scheduling approaches in cloud computing. We deduced from the studied literature that the majority of the work is based on well-known meta-heuristic approaches in cloud computing, including PSO, GA, and ACO algorithms. The most popular scheduling methods are found to be GA in bio-inspired and PSO in swarm intelligence. Other algorithms, such as ICA, CSO, BA, and ABC, have also been utilised in work scheduling, but to a lesser extent.

Lastly, we draw the conclusion that the makespan is the criterion that has been examined the most in the literature. We intend to use hybrid meta-heuristic techniques to develop a new failure handling model in subsequent work.

REFERENCES

[1] Moghaddam, F.F., Ahmadi, M., Sarvari, S., Eslami, M. and Golkar, A., 2015. Cloud computing challenges and opportunities: A survey. Telematics and Future Generation Networks (TAFGEN), 2015 1st International Conference on, pp.34-38.

[2] Oppitz, M. and Tomsu, P., 2018. Future Technologies of the Cloud Century. In: Inventing the Cloud Century. Springer, pp.511-545.

[3] Laghari, A.A., He, H., Halepoto, I.A., Memon, M.S. and Parveen, S., 2017. Analysis of quality of experience frameworks for cloud computing. IJCSNS, 17, p.228.

[4] Kumar, V., Laghari, A.A., Karim, S., Shakir, M. and Brohi, A.A., 2019. Comparison of Fog Computing & Cloud Computing. International Journal of Mathematical Sciences and Computing (IJMSC), 5, pp.31-41.

[5] Buyya, R., Broberg, J. and Goscinski, A.M., 2011. Cloud Computing Principles and Paradigms.

[6] Mittal, S. and Katal, A., 2016. An Optimized Task Scheduling Algorithm in Cloud Computing. Advanced Computing (IACC), 2016 IEEE 6th International Conference on, pp.197-202.

[7] Zhang, Q., Cheng, L. and Boutaba, R., 2010. Cloud computing: state-of-the-art and research challenges. Journal of Internet Services and Applications, 1, pp.7-18.

[8] Laghari, A.A., He, H., Shafiq, M. and Khan, A., 2016. Assessing effect of Cloud distance on end user's Quality of Experience (QoE). 2016 2nd IEEE International Conference on Computer and Communications (ICCC), pp.500-505.

[9] Laghari, A.A., He, H., Khan, A., Kumar, N. and Kharel, R., 2018. Quality of experience framework for cloud computing (QoC). IEEE Access, 6, pp.64876-64890.

[10] Laghari, A.A., He, H., Shafiq, M. and Khan, A., 2018. Assessment of quality of experience (QoE) of image compression in social cloud computing. Multiagent and Grid Systems, 14, pp.125-143.

[11] Tan, X. and Ai, B., 2011. The issues of cloud computing security in high-speed railway. Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on, pp.4358-4363.

[12] Sharkh, M.A., Jammal, M., Shami, A. and Ouda, A., 2013. Resource allocation in a network-based cloud computing environment: design challenges. IEEE Communications Magazine, 51, pp.46-52.

[13] De Chaves, S.A., Carlos, B., Westphall, C.M. and Gerônimo, G.A., 2011. Customer security concerns in cloud computing. Proceedings of The Tenth International Conference on Networks (ICN), pp.7-11.

[14] Petcu, D., 2011. Portability and interoperability between clouds: challenges and case study. European Conference on a Service-Based Internet, pp.62-74.

[15] Ranaldo, N. and Zimeo, E., 2009. Time and cost-driven scheduling of data parallel tasks in grid workflows. IEEE Systems Journal, 3, pp.104-120.

[16] Mathew, T., Sekaran, K.C. and Jose, J., 2014. Study and analysis of various task scheduling algorithms in the cloud computing environment. Advances in Computing, Communications and Informatics (ICACCI), 2014 International Conference on, pp.658-664.

[17] Talbi, E.-G., 2009. Metaheuristics: From Design to Implementation. John Wiley & Sons.

[18] Wang, L., Ma, Y., Yan, J., Chang, V. and Zomaya, A.Y., 2018. pipsCloud: High performance cloud computing for remote sensing big data management and processing. Future Generation Computer Systems, 78, pp.353-368.

[19] Parikh, S.M., Patel, N.M. and Prajapati, H.B., 2017. Resource management in cloud computing: Classification and taxonomy. arXiv preprint arXiv:1703.00374.

[20] Sookhak, M., Gani, A., Khan, M.K. and Buyya, R., 2017. Dynamic remote data auditing for securing big data storage in cloud computing. Information Sciences, 380, pp.101-116.

[21] Jennings, B. and Stadler, R., 2015. Resource management in clouds: Survey and research challenges. Journal of Network and Systems Management, 23, pp.567-619.

[22] Fischer, A., Botero, J.F., Beck, M.T., De Meer, H. and Hesselbach, X., 2013. Virtual network embedding: A survey. IEEE Communications Surveys & Tutorials, 15, pp.1888-1906.

[23] Duan, Q., Yan, Y. and 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, pp.373-392.

[24] Stergiou, C., Psannis, K.E., Kim, B.G. and Gupta, B., 2018. Secure integration of IoT and cloud computing. Future Generation Computer Systems, 78, pp.964-975.

[25] Manvi, S.S. and Shyam, G.K., 2014. Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey. Journal of Network and Computer Applications, 41, pp.424-440.

[26] Hussain, M. and Abdulsalam, H., 2011. SECaaS: security as a service for cloud-based applications. Proceedings of the Second Kuwait Conference on e-Services and e-Systems, p.8.

[27] Ye, X., Yin, Y. and Lan, L., 2017. Energy-efficient many-objective virtual machine placement optimization in a cloud computing environment. IEEE Access.

[28] Anuradha, V. and Sumathi, D., 2014. A survey on resource allocation strategies in cloud computing. Information Communication and Embedded Systems (ICICES), 2014 International Conference on, pp.1-7.

[29] Nagesh, B.B., 2014. Resource provisioning techniques in cloud computing environment—A survey. IJRCCT, 3, pp.395-401.

[30] Leivadeas, A., Papagianni, C. and Papavassiliou, S., 2013. Efficient resource mapping framework over networked clouds via iterated local search-based request partitioning. IEEE Transactions on Parallel and Distributed Systems, 24, pp.1077-1086.

[31] Endo, P.T., de Almeida Palhares, A.V., Pereira, N.N., Goncalves, G.E., Sadok, D., Kelner, J. et al., 2011. Resource allocation for distributed cloud: concepts and research challenges. IEEE Network, 25.

[32] Reuther, A., Byun, C., Arcand, W., Bestor, D., Bergeron, B., Hubbell, M. et al., 2017. Scalable system scheduling for HPC and big data. arXiv preprint arXiv:1705.03102.

[33] Nallakumar, R., Sengottaiyan, N. and Priya, K.S., 2014. A survey on scheduling and the attributes of task scheduling in the cloud. International Journal of Advanced Research in Computer Communication Engineering, 3, pp.8167-8171.

[34] Yu, J., Buyya, R. and Ramamohanarao, K., 2008. Workflow scheduling algorithms for grid computing. Metaheuristics for Scheduling in Distributed Computing Environments, pp.173-214.

[35] Raghavan, S., Sarwesh, P., Marimuthu, C. and Chandrasekaran, K., 2015. Bat algorithm for scheduling workflow applications in cloud. 2015 International Conference on Electronic Design, Computer Networks & Automated Verification (EDCAV), pp.139-144.

[36] Lakshmi, R.D. and 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.

[37] Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A. and Buyya, R., 2011. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience, 41, pp.23-50.

[38] Kumar, R. and Sahoo, G., 2014. Cloud computing simulation using CloudSim. arXiv preprint arXiv:1403.3253.

[39] Toosi, A.N., Calheiros, R.N. and Buyya, R., 2014. Interconnected cloud computing environments: challenges, taxonomy, and survey. ACM Computing Surveys (CSUR), 47, p.7.

[40] Sridhar, M. and Babu, G.R.M., 2015. Hybrid particle swarm optimization scheduling for cloud computing. 2015 IEEE International Advance Computing Conference (IACC), pp.1196-1200.

[41] Baliga, J., Ayre, R.W., Hinton, K. and Tucker, R.S., 2011. Green cloud computing: Balancing

energy in processing, storage, and transport. Proceedings of the IEEE, 99, pp.149-167.
[42] Rodriguez, M.A. and Buyya, R., 2014. Deadline-based resource provisioning and scheduling algorithm for scientific workflows on clouds. IEEE Transactions on Cloud Computing, 2, pp.222-235.
[43] Wu, Z., Ni, Z., Gu, L. and Liu, X., 2010. A revised discrete particle swarm optimization for cloud workflow scheduling. 2010 International Conference on Computational Intelligence and Security (CIS), pp.184-188.
[44] Manasrah, A.M. and Ba Ali, H., 2018. Workflow scheduling using hybrid GA-PSO algorithm in cloud computing. Wireless Communications and Mobile Computing, 2018.
[45] Kaur, K., Kaur, N. and Kaur, K., 2018. A novel context and load-aware family genetic algorithm-based task scheduling in cloud computing. In: Data Engineering and Intelligent Computing. Springer, pp.521-531.
[46] Bryk, P., Malawski, M., Juve, G. and Deelman, E., 2016. Storage-aware algorithms for scheduling of workflow ensembles in clouds. Journal of Grid Computing, 14, pp.359-378.
[47] Verma, A. and Kaushal, S., 2013. Budget constrained priority-based genetic algorithm for workflow scheduling in cloud.
[48] Pragaladan, R. and Maheswari, R., 2014. Improve workflow scheduling technique for novel particle swarm optimization in cloud environment. International Journal of Engineering Research and General Science, 2.
[49] Fard, H.M., Prodan, R., Barrionuevo, J.J.D. and Fahringer, T., 2012. A multi-objective approach for workflow scheduling in heterogeneous environments. 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), pp.300-309.
[50] Tang, X., Li, K. and Liao, G., 2014. An effective reliability-driven technique of allocating tasks on heterogeneous cluster systems. Cluster Computing, 17, pp.1413-1425.
[51] Singh, L. and Singh, S., 2014. Deadline and cost-based ant colony optimization algorithm for scheduling workflow applications in hybrid cloud. Journal of Scientific & Engineering Research, 5, pp.1417-1420.
[52] Kang, Q.M., He, H., Song, H.M. and Deng, R., 2010. Task allocation for maximizing reliability of distributed computing systems using honeybee mating optimization. Journal of Systems and Software, 83, pp.2165-2174.
[53] Sagnika, S., Bilgaiyan, S. and Mishra, B.S.P., 2018. Workflow scheduling in cloud computing environment using bat algorithm. In: Proceedings of First International Conference on Smart System, Innovations and Computing. Springer, pp.149-163.
[54] Ebadifard, F. and Babamir, S.M., 2018. A PSO-based task scheduling algorithm improved using a load-balancing technique for the cloud computing environment. Concurrency and Computation: Practice and Experience, 30, p.e4368.
[55] Dordaie, N. and Navimipour, N.J., 2017. A hybrid particle swarm optimization and hill climbing algorithm for task scheduling in cloud environments. ICT Express.
[56] Kumar, A.S. and Venkatesan, M., 2018. Task scheduling in a cloud computing environment using HGPSO algorithm. Cluster Computing, pp.1-7.
[57] Al-Arasi, R.A. and Saif, A., 2018. HTSCC: A hybrid task scheduling algorithm in cloud computing environment. International Journal of Computers & Technology, 17, pp.7236-7246.
[58] Elaziz, M.A., Xiong, S., Jayasena, K. and Li, L., 2019. Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution. Knowledge-Based Systems, 169, pp.39-52.
[59] Alworafi, M.A., Dhari, A., El-Booz, S.A., Nasr, A.A., Arpitha and Mallappa, S., 2019. An enhanced task scheduling in cloud computing based on hybrid approach. In: Data Analytics and Learning. Springer, pp.11-25.
[60] Potluri, S. and Rao, K.S., 2020. Simulation of QoS-based task scheduling policy for dependent and independent tasks in a cloud environment. In: Smart Intelligent Computing and Applications. Springer, pp.515-525.