Semantics of Resource Scheduling Algorithm in Cloud Computing: A Literature Survey

[1]Punit Mittal, [2]Dr. Satender Kumar, [3]Dr. Swati Sharma

[1]Department of Computer Science and Engineering
Quantum University
Roorkee, India
[2]School of Computer Science and Engineering
Quantum University
Roorkee, India
[3]Department of Information Technology
Meerut Institute of Engineering and Technology
Meerut, India

***Abstract*:**Resource scheduling has become a key issue in cloud computing as a result of the growth in on-demand requests and the variety of cloud resources. Through the internet, users may access pay-as-you-go cloud services that are based on elasticity, dynamism, and unpredictability. Due to rising demand for cloud services and increasingly complex and diverse applications, the workload in the cloud environment has grown over the last several years. Inefficient scheduling techniques struggle with resource imbalances that either worsen service performance, in the event of resource overutilization, or waste cloud resources, in the case of resource underutilization. The scheduling algorithm should maximize quality-of-service metrics including energy use, resource usage, makespan time, dependability, reaction time, availability, cost, etc. To accomplish the aforementioned objective, a large number of cutting-edge scheduling algorithms based on heuristic, meta-heuristic, and hybrid approaches have been researched and examined in this literature study. The recommended scheduling techniques are thoroughly analyzed and categorized in this article. Our methodical and thorough analysis will help future cloud computing scholars get started and develop scheduling methods going forward.

Keywords- *Cloud computing, Virtual machines, Resource provisioning, Resource scheduling, Heuristic, Meta-Heuristic, Hybrid-Heuristic*

INTRODUCTION

Customers may access unlimited computer resources at any time and only pay for what they use using cloud computing, a pay-as-you-go strategy. The cloud offers resource pooling, changeable and malleable [1] (horizontal and vertical) scalability [2], throughput, utility services, high availability, performance, managed services, and more. The centralized cloud infrastructure management caused this. Cloud service providers and clients may leverage virtualization and dynamic resource scheduling. Efficient resource scheduling speeds up tasks and maximizes resource use, decreasing resource usage. Due to the cloud datacenter's growing load, task allocation is crucial. Workload increase may drain cloud resources, causing a scarcity. Thus, cloud computing is still developing, requiring more research to distribute workloads to cloud resources and schedule to improve service quality. Scheduling optimizes task resource allocation. Scheduling algorithms may improve resource utilization, dependability, job rejection ratio, execution cost, and energy usage. We must improve without compromising service level agreement. Scheduling must also account for deadlines, priorities, and load imbalance, which occurs when resources are over- or under-utilized. As illustrated in Fig. 1, cloud computing resource management includes scheduling and provisioning. [3] Resource scheduling also includes mapping, load balancing, brokering, and resource allocation. This section covers cloud-computing resource provisioning basics before scheduling. Classifying scheduling techniques and analyzing cutting-edge scheduling algorithms will follow. These algorithms are heuristic, meta-heuristic, and hybrid. Additionally, the assessment will contain key performance features and algorithm restrictions.

Resource provisioning is the process by which virtualized resources are provided for allocation to users. When a cloud service provider gets a request for resources from users, it proceeds to build the appropriate number of virtual machines (VMs) and allocates them to customers in accordance with their respective demands. The process of resource provisioning involves meeting the needs of users by taking into account quality of service criteria (QoS), engaging in negotiations for service level agreements (SLAs), and aligning resources with projected workloads.

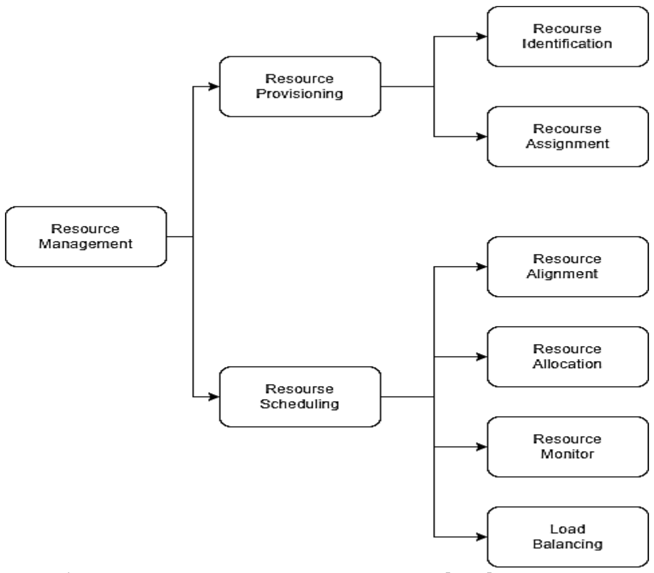

Figure 1. Resource Management in cloud computing

The sequence in which tasks should be done is determined by *resource scheduling*, which takes required quality of service (QoS) considerations into account. Using heuristic or meta-heuristic techniques, scheduling is responsible for selecting the best virtual machines for task execution. Furthermore, scheduling is in charge of ensuring that quality of service (QoS) restrictions are met. Resource scheduling may be done using two different approaches. The first method is on-demand scheduling, in which the cloud service provider quickly distributes resources to diverse workloads in accordance with demand. To address the problem of an unequal workload distribution, where it may be possible to complete a greater number of activities on a single virtual machine (VM), the method indicated above has to be improved. As a result, performance suffers and there is a potential that over-provisioning issues may arise. Long-term reservations, which cause over-provisioning issues when multiple virtual machines are maintained optimally, are the second difficulty. Due to inefficient use of resources and time, over- and under-provisioning raises the cost of services. There are two forms of scheduling: static and dynamic.

Static scheduling methods need knowledge on job length, amount, deadlines, and resources like processing capacity, compute power, and storage to work well. Static algorithms function well when workload variance is low and system behavior is steady. In a cloud context, workload and system behavior vary quickly. Thus, static algorithms should be optimized for cloud computing. Static algorithms are easy to implement, but they must change service quality parameters and perform well in real-world settings. FIFO, RR, and SJF are static algorithms.

Dynamic scheduling does not need job and node knowledge. However, node monitoring is essential. These algorithms work better in the cloud because they can shift jobs from overloaded to under loaded nodes. This dynamic modification of algorithm conditions to node load changes, whether positive or negative, help them work. HEFT, dynamic round robin, CBHD, WLC, ACO, and PSO are dynamic algorithms. Due to workload and system dynamics, the study recommends utilizing dynamic task scheduling techniques in the cloud. Dynamic methods may approximate NP-hard problems.

TASK SCHEDULING TECHNIQUES

Distributed computing systems have several scheduling methods. Figure 4 shows the three main scheduling strategies: meta-heuristic, classical, and heuristic..

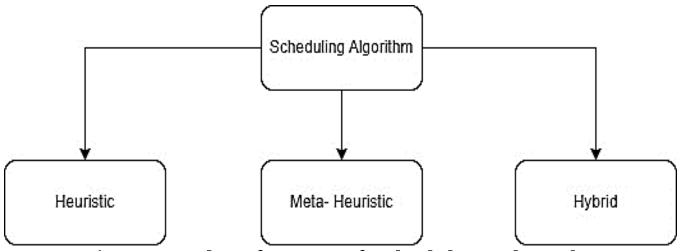

Figure 2. Classification of Scheduling Algorithm

A. Heuristic scheduling algorithm

Heuristic algorithms excel at certain issues but struggle with others. Although heuristic algorithms solve problems in a defined time, they cannot handle complicated optimization problems. Many cloud heuristic algorithms have been developed to schedule workflows and individual operations or apps. Heuristic algorithms underpin HEFT-based, cluster-based, List-based, task duplication-based, etc. techniques.

HEFT-based Heuristic:The HEFT approach was created by [4] and calculates the typical processing time of upcoming tasks as well as the typical communication time between virtual machines (resources) of two future jobs. The following tasks are listed in decreasing priority, with higher rank values giving tasks greater importance. Tasks are scheduled based on their priority and allocated to the virtual machine that can complete them as quickly as feasible. A list of heuristic algorithms built on the HEFT method is shown in Table 1. While (x) indicates that this specific QoS is not concerned, () marks that the matching algorithm is working on it.

Table 1. HEFT-based Heuristic

S. No.	**Algorithm**	**Nature of Tasks**	**Make span**	**Energ y**	**Relia bility**	**nse Time**	**Balan cing**	**Deadl ines**	**rce Alloca**	**SLR**
[5]	Improved HEFT	Workflow	✓	×	×	×	×	×	×	×
[6]	Heterogeneous Earliest Finish with Duplicator (HEFD)	Independent	✓	×	×	×	×	×	×	×
[7]	Clustering Based HEFT with Duplication (CBHD)	Workflow	✓	×	×	×	×	×	✓	×
[8]	Bandwidth-aware algorithm	Independent	×	×	×	×	×	×	✓	×
[9]	Cluster combining algorithm (cca)	Workflow	✓	×	×	×	×	×	×	×
[10]	Improved predict earliest finish time	Independent	✓	×	×	×	×	×	×	✓
[11]	The schedule length using the budget level (MSLBL)	Independent	✓	×	×	×	×	×	×	✓

[5] provide an improved iteration of HEFT in which the heuristic's locally optimum choices take into account the schedule's future as well as the decision's potential effects on the task's progeny. Heterogeneous Earliest Finish with Duplicator (HEFD) was recommended by source [6]. Task priority is an essential characteristic for list scheduling algorithms, and this work provides a unique technique for evaluating task priority that takes the performance change in the target HCS using modification into account. The research also offers a novel suggestion: in order to get the ideal scheduling arrangement, attempt to repeat all parent tasks. [7] proposed a method for job scheduling in cloud computing settings that take bandwidth into account. A nonlinear programming approach for the divisible task-scheduling issue is provided under the restricted multi-port paradigm. You may find the optimum distribution strategy to choose how many tasks to assign to each virtual resource node by solving this model. The bandwidth-aware task-scheduling (BATS) method is a recommended heuristic technique for split load scheduling based on the enhanced distribution scheme.It should look for groupings of process stages that may be carried out concurrently while exchanging data, according to [8]. Running on a machine with several cores may be a viable option for these types of groupings. However, additional groups have to be conducted one after the other. This technique determines whether or not these groups may be executed sequentially. The outcomes of the studies demonstrate that the method has a good impact on how much a process costs to operate and how much time it takes to run.[9] offered three techniques for allocating jobs in a multi-cloud environment that take resource use into account. The techniques have been modified to operate in a multi-cloud environment and are based on the conventional Min-Min and Max-Min algorithms. All algorithms go through the same three processes—matching, assigning, and ordering—to function in a multi-cloud environment. The comparative findings demonstrate that the recommended strategies outperform the ones already in use. This is so that the gaps between "matching" and "scheduling" may be filled in by a new phase termed "allocating." Additionally, collaboration between individuals from various clouds improves mapping efforts. Our Makespan, typical cloud consumption, and performance all increased as a result.
[10] outlines an experience. Based on the Heterogeneous Earliest Finish Time (HEFT) algorithm, HEFT estimates how much work each resource must do by considering how previous tasks have been completed. We propose a novel approach to calculate a task's HEFT score that takes into account the minimal average time it took to perform a job on all relevant resources over its previous runs in order to make the experiential HEFT technique effective. [11] discuss the Experiential Heterogeneous Earliest Finish Time (EHEFT) technique, which added a value indicating the typical duration of a work on each crucial resource to the original HEFT algorithm's calculation of a task's rank. The EHEFT algorithm outperforms CPOP methods and the original HEFT in terms of schedule length ratio and speed.

Cluster based Algorithm: The procedure starts by selecting a centroid value at random for each cluster. Since tasks must be organized, task durations act as the system's data points. Jobs are grouped and scheduled to the appropriate VMs based on the processing capability of the VMs. As a consequence, a balanced workload is distributed across all the VMs. A list of heuristic algorithms built on the Cluster algorithm is shown in Table 2.
[12] examined a contrived model of how tasks are routed to cloud data centers. Jobs are forwarded to one of the computers and queued up there when they come in. Then, in order to have adequate resources to process them all at once, each server selects a group of tasks from its lines. With the premise that work quantities are predetermined and have upper bounds, this issue has already been examined. A technique that maintains steady traffic flow in a region with fewer capacity was proposed. It has been suggested to use a load-balancing and scheduling strategy that optimizes output without making any assumptions about work sizes or upper bounds.

Table 2. Cluster-based Heuristic

S. No.	Algorithm	Nature of Tasks	Make span	Energ y	Relia bility	nse Time	Balan cing	Deadl ines	rce Alloca	SLR
[12]	Throughput-optimal load balancing and scheduling algorithm	Independent	✓	×	×	×	✓	×	✓	×
[13]	Clustering for Minimizing the Worst Schedule Length (CMWSL)	Workflow	×	×	×	×	×	×	×	✓

[13] proposed a technique known as Clustering for Minimizing the Worst Plan Length (CMWSL), which use clustering to determine the most effective way to schedule activities across a variety of computer types. The recommended approach first determines the minimum amount of time that each machine will spend processing data overall by examining both the system and the software. Therefore, by keeping the number of processors utilized for real processing to a minimum, the Worst Schedule Length (WSL) is maintained as low as feasible. The plan is then shortened until the total processing time in a task cluster exceeds the lower limit by performing the real task assignment and task grouping.

Task Duplication-based Algorithm (TBD):The TBD scheme is an essential approach to resolving this problem. The main idea is to spread out jobs across numerous computers so that results may be obtained from other computers and computation time can be switched for communication time. A list of heuristic algorithms built on the Task Duplication method is shown in Table 3. When resources are distributed unevenly, [14] presented the heterogeneous critical task (HCT) scheduling approach, a novel task-scheduling system. Using the directed acyclic graph and the Gantt chart, the HCT algorithm identifies the critical activity and the free interval. The first step of the HCT technique is to identify the essential subtasks of a particular task. Then, at the available time slot, it replicates those subtasks temporarily on the processor with the given work. This helps in reducing the starting time of the specified job. [15] examined the issue of grouping related jobs in cloud computing. When the computational and communication components are too dissimilar, the aim is to reduce the time it takes to complete a task. We propose a Dynamic Priority List Scheduling (DPLS) approach that forecasts when a task will be completed depending on how it is being carried out right now. Based on the server allocation of the jobs that have already been planned, the algorithm estimates the amount of processing time that each particular task will need. The server provision relies on the outcomes of the jobs that have previously been planned, and it also selects the subsequent work to be scheduled. [16] proposes a task duplication and insertion approach based on List Scheduling (DILS) that makes advantage of job duplication, task insertion, and dynamic end time predictions. The software anticipates when activities will be completed depending on when they were first planned.

Table 3. Task Duplication-based Algorithm

S. No.	Algorithm	Nature of Tasks	Makespa n	Energy	Reliabilit y	Respons e Time	Balancin De line	ad s	Allocatio n	SLR
[14]	Heterogeneous critical task	Workfl ow	✓	×	×	✓	×	×	×	×
[15]	Task Duplication based Clustering Algorithm (TDCA)	Workfl ow	✓	×	×	×	×	×	×	×

[16]	Stochastic cost-effective deadline-aware (S-CEDA)	Workflow	✓	×	×	×	×	×	×	×

To speed up task execution, it also replicates jobs on other nodes, reduces transfer time, and schedules tasks for periods when no one else is working.

***Task Priority based Algorithm*:**One of the most common cloud computing scheduling strategies is non-preemptive priority scheduling. Each work is prioritized. The most important procedure should come first, etc. Same-priority processes are completed in sequence. Priority-based heuristic algorithms are summarized in Table 4.

Table 4. Task Priority based Algorithm

S. No.	**Algorithm**	**Nature of Tasks**	**Makespan**	**Energy**	**Reliability**	**Response Time**	**Load Balancing**	**Deadlines**	**Resource Allocation**	**SLR**
[17]	Priority based job scheduling algorithm (PJSC)	Independent	✓	×	×	×	×	×	×	×
[18]	Priority based divisible load scheduling method.	Independent	✓	×	×	×	✓	×	×	×
[19]	heterogeneous scheduling algorithm with improved task priority (HSIP)	Workflow	✓	×	×	×	×	×	×	✓
[20]	HSIP (heterogeneous scheduling algorithm with improved task priority)	Workflow	✓	×	×	×	×	×	×	×
[21]	priority based job scheduling algorithm	Independent	×	×	×	✓	×	×	×	×
[22]	Priority based job scheduling algorithm	Independent	✓	×	×	×	✓	×	×	×
[23]	Dynamic priority list scheduling (dpls)	Workflow	×	×	×	×	×	×	✓	×

[17] introduces priority-based cloud work scheduling. This priority-based task scheduling method may be employed in cloud computing. We picked "PJSC" as our name. Complexity, consistency, and completion time of the recommended method are also reviewed. This study shows that the proposed strategy is doable at its complexity. [18] categorized burdens by significance. To reduce transmission rate cheating's impact on end time. The proposed approach was tested on many function approximation challenges. The suggested method

works better than the other two. [19] presented HSIP with improved task priority, which uses standard deviation and Communication Cost Weight to evaluate tasks and an entrance task copy selection criteria to minimize Makespan and enhance unoccupied time slots. Our solution outperforms others in plan length ratio, speedup, and efficiency.HSIP (heterogeneous scheduling approach with increased task priority) [20]. Both sections of HSIP work: To make scheduling priority more realistic, standard deviation with better magnitude was utilized to determine tasks priority. To shorten the timeframe, an entry task duplication selection strategy was applied. HSIP outperforms HEFT, PEFT, CPOP, and SDBATS. Recommended: advanced design, queueing and priority models, priority assignment module, and priority-based job scheduling. Compared to existing work scheduling algorithms, the recommended method reduces response time and costs. The suggested priority schedule considerably reduced response time and expense. It organizes tasks by wait time well. Higher general output reduces average reaction time and cost. [22] demonstrated that virtual machines' resource demands and workloads fluctuate fast and in bursts, causing load-balancing techniques to move them poorly. This paper proposes stochastic load-balancing to tackle this issue. This approach seeks to statistically guarantee that relocating virtual machines will not overwhelm resources while minimizing moving costs. Our strategy can anticipate resource demand circulation and numerous resource demands with stochastic characterization. We will investigate large-scale VM demands and how task distributions effect load balancing in the future. [23] describes edge computing, which sends data to the cloud via edge devices. This reduces delays and improves system efficiency. Mixed data grouping and deep learning-based resource ordering reduce processing complexity in the proposed approach.

List Based Scheduling Algorithm:The greedy approach for identical machine scheduling is list scheduling. A list of tasks to be done on m machines is the algorithm's input. Task importance or receipt order determines the list's order. Heuristic methods based on List Based Scheduling Algorithm are summarized in Table 5.

[24] examines task scheduling to avoid conflicts. New idea: dynamically schedule edges to connections. The quickest communication time search methodology uses shortest-path search. Another novel notion in this research is iterative rank computation-based scheduling priority on different random processor networks. Ultimately, a parallel solution is provided to reduce program time complexity and speed up O(PPE). [25] proposed real-time rolling-horizon scheduling in virtualized clouds. A task-based energy usage model is then presented and assessed. Based on our scheduling concept, we develop EARH, an energy-aware scheduling strategy for real-time tasks that happen at different times. The EARH employs rolling-horizon optimization and can operate with other energy-aware scheduling techniques. We also provide two ways to balance work scheduling and energy savings by shifting resources up and down. Improved Predict Earliest Finish Time, a list-based static task scheduling solution for varied computer environments, is introduced in [26]. The pessimistic cost table determines task priority, and a critical node cost table implements feature prediction. If a node has at least one immediate successor as the critical node, the best processor is allocated.

Table 5: List Based Scheduling Algorithm:

S. No.	**Algorithm**	**Nature of Tasks**	**Make span**	**Energ y**	**Relia bility**	**Respo nse Time**	**Balan**	**Deadl ines**	**rce Alloca**	**SLR**
[24]	Earliest finish communication time search algorithm	Workflow	✓	×	×	✓	×	×	×	×
[25]	Energy-aware scheduling algorithm named EARH	Workflow	×	✓	×	×	×	×	×	×
[26]	based on a critical node cost table and pessimistic cost table	Workflow	✓	×	×	×	×	×	✓	✓

[27]	Heterogeneous Dynamic List Task Scheduling (HDLTS)	Workflow	✓	×	×	×	✓	×	×	×
[28]	DLS based multi-objective method	Independent	×	×	×	×	×	×	×	✓
[29]	Heterogeneous dynamic list task scheduling (hdlts)	Workflow	✓	×	×	×	×	×	×	×
[30]	Experiential HEFT	Independent	✓	×	×	×	×	×	×	×
[31]	List Scheduling with Task Duplication (LSTD)	Workflow	✓	×	×	×	×	×	✓	×
[32]	Duplication and Insertion algorithm based on List Scheduling	Independent	✓	×	×	×	✓	×	×	×

The Improved Predict Earliest Finish Time algorithm outperforms the Predict and Heterogeneous algorithms in schedule length ratio, frequency of the best result, and robustness while maintaining the same time complexity. [27] proposed Heterogeneous Dynamic List Task Scheduling for application workflow scheduling. This method duplicates tasks only if execution time decreases. The execution mapping starts only for tasks with the necessary input criteria. Top resource for job with maximum execution time variability. HDLTS outperforms the other algorithms on both metrics.[28] proposed utilizing the budget level (MSLBL) to identify processors that fulfill the budget restriction and decrease application schedule duration. Completing the expenditure limit and shortening the strategy are two additional small challenges. Moving the application's budget limit to each job solves the first sub-problem. Heuristically organizing low-time tasks solves the second sub-problem. [29] proposed a Task Duplication-based Clustering Algorithm (TDCA) to increase plan performance utilizing completed tasks. TDCA simplifies parameter calculation, reducing duplication and combining. The study and experiments used random graphs with various DAG depth, breadth, communication-computing cost ratio, and processing power. Our findings demonstrate the TDCA algorithm's excellence. The plan Makespan of task-duplication-based algorithms for various systems with varying communication-to-computing cost ratios is improved. List Scheduling with Task Duplication (LSTD) effectively decreases workflow application create time [30]. The LSTD adds task repetition to the list-scheduling technique without complicating it. The three phases make LSTD operate. First, it ranks occupations by significance. Next, repeat the processor entry task if it improves process flow and maintains the processor low. Last, the processor is allocated jobs using a "insertion-based policy." When idle, this strategy places a new task between two existing processor tasks. In the future, the LSTD algorithm will design complex cloud process applications depending on user cost and time restrictions. We need new processor assignment (or virtual machine assignment) concepts to achieve this. How long it takes to receive VMs, how fast they might fluctuate, if there is a restriction, whether things must be done in a particular sequence, and how long service providers allow you use their space have been evaluated. To enhance work task division, [31] introduced a novel Multi-Objective Ameliorated Repetitive Resource Allocation Algorithm that can respond rapidly to unexpected demands. Resource efficiency and percentage matching lengths maximize resource utilization. [32] focuses on both the problems of stability and date promise in these RL, especially Deep RL (DRL)-based scheduling approaches. We use the retraining time to measure robustness and look into ways to improve the resilience and the date promise of DRL-based scheduling. We present MLR-TC-DRLS, a proper, stable schedule system based on Meta Deep Reinforcement Learning that can promise deadlines for time-critical jobs and adapt quickly to changing situations

B. *Meta-Heuristic scheduling algorithm*

Meta-heuristic algorithms may solve problems across disciplines if they perform well. NP-hard optimization problems are often solved using metaheuristics. A meta-heuristic algorithm summary is in Table 6.

Table 6. Meta-Heuristic scheduling algorithm

S. No.	Algorithm	Nature of Tasks	Makespan	Energy	Reliability	Response Time	Load Balancing	Deadlines	rce Alloca	SLR
[33]	Genetic algorithms based	Workflow	✓	×	×	×	×	×	×	×
[34]	PSO based Algorithm	Workflow	×	×	×	×	×	×	✓	×
[35]	ACO based Algorithm	Workflow	×	×	×	×	✓	×	×	×
[36]	Multiple Priority Queues Genetic Algorithm (MPQGA)	Workflow	✓	×	×	×	×	×	×	×
[37]	Game Theory based task scheduling	Independent	×	×	γ	×	×	×	×	×
[38]	learner genetic algorithm (LAGA)	Workflow	✓	×	×	×	×	×	×	×
[39]	Deep Q-learning task scheduling (DQTS)	Workflow	✓	×	×	×	×	×	×	×
[40]	Modified canopy fuzzy c-means algorithm (MCFCMA).	Independent	×	×	×	×	×	×	✓	×
[41]	Based on deep reinforcement learning model	Independent	✓	×	×	×	✓	✓	×	×
[42]	Dynamic Cost-Efficient Deadline-Aware (DCEDA)	Workflow	×	✓	×	×	×	×	✓	×
[43]	Power Migration Expand (PowMigExpand)	Workflow	✓	×	✓	×	×	×	×	×
[44]	Reliability-aware task scheduling algorithm (RATSA)	Workflow	✓	×	×	×	×	×	×	×
[45]	Limited Duplication-Based List Scheduling Algorithm (LDLS)	Workflow	×	×	×	×	×	×	✓	×
[46]	Grey Wolf Optimizer (GWO) algorithm based method	Workflow	✓	×	×	×	×	×	×	×
[47]	hyper-heuristic algorithm based on reinforcement learning (HHRL)	Workflow	✓	×	×	×	✓	✓	×	×

[33] discusses genetic algorithm-based autonomous DAG scheduling. The genetic algorithm optimizes well and considers several factors. In this paper, the basis for Grid systems using the suggested technique was addressed. [34] The proposed way to provide and organize IaaS Cloud research materials. Particle Swarm Optimization (PSO) is a meta-heuristic optimization methodology. This algorithm reduces process costs while fulfilling deadlines. [35] A cloud task scheduling strategy based on ant colony optimization for load management has been proposed and compared to existing techniques. Ant Colony Optimization (ACO) will place work on virtual computers as they arrive using random optimization search. Our main focus is balancing system load and speeding up project completion. According to the load-balancing factor, various resources should accomplish work at the same pace. This improves load-balancing. [36] A multiple priority queues genetic algorithm (MPQGA) task ordering mechanism is recommended for varied computer systems. Our method aims to take use of evolutionary and heuristic algorithms' strengths while avoiding their weaknesses. A genetic algorithm (GA) prioritizes subtasks and a heuristic-based earliest completion time (EFT) methodology finds a task-to-processor mapping solution. MQGA additionally creates DAG scheduling-friendly crossover, mutation, and fitness functions. [37] This paper uses game theory to simplify the model. Game theory suggests grouping tasks by reliability. The balanced scheduling approach is used for computer node job scheduling. The joint game model uses the task to determine the node rate-sharing approach. This is done via game strategy. Authors propose a novel approach to cloud computing task scheduling, although more work remains. The idea in this work ignores the cost of processing a task based on the internal planner. We may add impact elements to the program and boost performance. A learning genetic algorithm (LABA) for static processor scheduling in homogenous computer systems is described in [38]. For this reason, they created the Steepest Ascent Learning Criterion and the Next Ascent Learning Criterion, which employ punishment and reward to teach. An amazing approach to look for a timetable issue solution is not discovered, so the time to locate a solution is not locked in a local perfect solution. The scheduling mechanism additionally considers recurrent idle time to decrease Makespan. This paper proposes a genetic algorithm and quicker convergence rate technique to improve parallel setups. The author proposed a learning-based genetic method to improve computer utilization. "The suggested method can explore more of the problem's state space than existing algorithms due to its speed. This allows it to identify a better answer. [39] propose deep Q-learning task scheduling (DQTS), a novel artificial intelligence system that combines the advantages of Q-learning with deep neural networks. This novel solution addresses cloud DAG job management. Since deep Q-learning (DQL) is the motivation for fundamental model learning, our technique organizes tasks using DQL. DQTS outperforms other algorithms in learning, according to experiments. This suggests DQTS can plan cloud computing jobs, particularly in complicated contexts. We propose a new energy consumption model for future activities, which may reduce costs when the timeline is flexible. The modified canopy fuzzy c-means algorithm (MCFCMA) can address this cloud computing scheduling issue [40]. PSO links tasks to resources. The recommended strategy starts with load feedback-based autonomous task selection. Each virtual machine receives the required work organized by MCFCMA. VM selects parallel processing in VM management. [41] Deep reinforcement learning is used to suggest a fair green computing resource-sharing mechanism. This approach helps network users share resources well. Due to the "dimensionality problem," classical Q-learning fails when state space increases exponentially. [42] present a Dynamic Cost-Efficient timeline-Aware (DCEDA) heuristic for structuring Big Data activities that produces the cheapest plan within time constraints. DCEDA schedules on-demand jobs depending on whether a deadline will be missed. It also monitors the VM pool for active but idle VMs that incur expenses and overheads and eliminates them[43] Discuss MEC server resource allocation. Due to fluctuating MEC server utilization, their resources must be managed wisely to fulfill users' QoS demands and save energy. Power Migration Expand is a new resource allocation strategy. Our application directs user requests to the best server and offers User Equipment the finest resources based on our comprehensive utility function. Sometimes users relocate, thus PowMigExpand transfers UE requests to new locations. An inexpensive solution dubbed Energy Efficient Smart Allocator (EESA) employs deep learning to deliver requests to the finest computers in the most energy-efficient manner. RATSA is a novel task scheduling algorithm that emphasizes stability. RATSA uses evolutionary algorithms like SFLA and GA to plan tasks on directed acyclic graphs (DAGs). Population-based SFLA-GA solves the NP-complete RATSA Makespan optimization problem. A novel heuristic-based earliest completion time strategy for task mapping to virtual machines (VMs) reduces failures in the recommended manner. RATSA

improves certain existing approaches and performs well in Makespan, according to random DAG testing. [45] propose LDLS, a task duplication-based heuristic scheduling system that reduces project completion time while maintaining complexity. The LDLS ordering procedure has three steps: The maximal amount of duplicates per level and task is determined at the start so that just a few copies prevent regular jobs. Next, the PEFT calculates the optimistic cost table (OCT) and work order. Final step: ranking-based ordering and on-the-spot job doubling. This allows repeated activities to speed up subsequent ones. This solution speeds up work completion and increases machine performance while maintaining PEFT and HEFT's time complexity O (v2, p). [46] Improve resources using a hybrid technique to reduce transfer cost. Proposed technique uses Grey Wolf Optimizer (GWO) algorithm. The GWO was founded to aid society, particularly grey wolves hunt and kill their food. [47] demonstrate a hyper-heuristic strategy that employs reinforcement learning (HHRL) to speed up task completion. First, at the reward table setup stage of HHRL, population diversity and maximum time determine job scheduling and low-level heuristic approaches. Second, linear regression as a task processing complexity estimate is recommended to modify task scheduling.

C. Hybrid Scheduling Algorithm

Hybrid scheduling approaches combine meta-heuristic and heuristic scheduling to solve cloud computing resource scheduling problems. Table 7 describes hybrid scheduling techniques. propose a parallel bi-objective hybrid genetic algorithm that considers Makespan and energy. The island parallel model and multistate comparable model have received most of our attention. A novel energy-efficient approach uses dynamic voltage scaling (DVS). Based on energy usage, our timing strategy is much superior than others. Plans are quicker than algorithms in terms of completion time..

Table 7: Hybrid Scheduling Algorithm

S. No.	Algorithm	Nature of Tasks	Makespan	Energy	Reliability	Response Time	Load Balancing	Deadlines	Resource Allocation	SLR
[48]	Parallel bi-objective hybrid genetic algorithm	Independent	✓	✓	×	×	×	×	×	×
[49]	Energy efficient resource scheduling using Cultural emperor penguin optimizer (EERS-CEPO)	Workflow	✓	×	×	×	✓	×	✓	×
[50]	Enhanced Multi-Objective Particle Swarm Optimization with Clustering (EMOPSOC)	Workflow	✓	×	×	×	✓	×	✓	×

[49] EERS-CEPO in the GCC context schedules resources energy-efficiently utilizing the Cultural Emperor Penguin Optimizer (CEPO) approach. The strategy divides work amongst data centers or other resources to make it manageable. The CEPO algorithm combines CA and EPO. [50] presents a metaheuristic combinatorial model called Enhanced Multi-Objective Particle Swarm Optimization with Clustering (EMOPSOC) and Fog Picker to analyze fog node scheduling and placement. The recommended EMOPSOC is a tiny software that plans fog nodes dynamically using multi-objective optimization. The model improves

evolutionary computation with machine learning. It lowers IOT-fog device latency, wasting resources and causing unequal loads. It conserves resources and speeds responsiveness.

CONCLUSION AND FUTURE SCOPE

We will start by talking about several unresolved open research problems in the area of cloud computing. Numerous resource management strategies have been implemented to enhance key performance indicator parameters. However, they are still in their infancy stage due to some significant problems like heterogeneity, diversity, and complexity of applications, unpredictable end-user demands (for computing, storage, and memory, for example), varying cost, energy consumption, virtualization, reliability, scalability, etc. that have not yet been thoroughly researched as research Researchers are still struggling to come up with a creative compromise between energy efficiency and end-user demand availability.

There are still many unsolved questions. How does power use affect SLA in terms of customer satisfaction? Which devices need to be turned off, and when tasks or virtual machines (VMs) should be moved from one resource to another to reduce energy consumption. How can I discover fault tolerance and VM migration in the cloud at the same time?

The following attempts can be made in the future to get beyond the limitations of current algorithms: 1) Cloud computing is a business model, so the user's work priority is on the top. SLA should always maintained when change is detected. 2) VM migration, Energy consumption, task relocation, resource allocation, CPU monitoring and memory utilization, etc. There should be a method to handle these QoS parameters significantly. 3) A machine learning approach should be used to predict upcoming tasks and failures. 4) Network Bandwidth gets less attention than QoS, which can lead to network failure and contention. To improve the efficiency of the system, this factor should be considered. 5) Heterogeneity and decentralization can be achieved with the use of fog computing

References

[1] Kumar, M., Dubey, K., Sharma, S.C., 2018. Elastic and Flexible Deadline Constraint Load Balancing Algorithm for Cloud Computing. In: 6th International Conference on Smart Computing and Communications, pp. 717–724. India.

[2] Kumar, Mohit, Sharma, S.C., 2017. Dynamic load balancing algorithm for balancing the workload among virtual machine in cloud computing. in Procedia Computer Science 115, 322–329.

[3] Madni, S., et al., 2016. Resource scheduling for infrastructure as a service (IaaS) in cloud computing: challenges and opportunities. J. Netw. Comput. Appl. 68, 173–200.

[4] H. Topcuoglu, S. Hariri and Min-You Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing," in IEEE Transactions on Parallel and Distributed Systems, vol. 13, no. 3, pp. 260-274, March 2002, doi: 10.1109/71.993206.

[5] L. F. Bittencourt, R. Sakellariou and E. R. M. Madeira, "DAG Scheduling Using a Lookahead Variant of the Heterogeneous Earliest Finish Time Algorithm," 2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing, Pisa, Italy, 2010, pp. 27-34, doi: 10.1109/PDP.2010.56.

[6] "Tang X, Li K, Liao G, Li R,""List scheduling with duplication for heterogeneous computing systems,"" 2010, Journal of Parallel and Distributed Computing, Volume 70, Issue 4, Pages 323-329, ISSN 0743-7315, https://doi.org/10.1016/j.jpdc.2010.01.003."

[7] Doaa M. Abdelkader, FatmaOmara, "Dynamic task scheduling algorithm with load balancing for heterogeneous computing system," 2012, Egyptian Informatics Journal, Volume 13, Issue 2, Pages 135-145, ISSN 1110-8665, https://doi.org/10.1016/j.eij.2012.04.001.

[8] Lin, W., Liang, C., Wang, J.Z. and Buyya, R. (2014), Bandwidth-aware divisible task scheduling for cloud computing. Softw. Pract. Exper., 44: 163-174. https://doi.org/10.1002/spe.2163

[9] Deldari, A., Naghibzadeh, M., Abrishami, S., Rezaeian, A. A Clustering Approach to Scientific Workflow Scheduling on the Cloud with Deadline and Cost Constraints. AUT Journal of Modeling and Simulation, 2014; 46(1): 19-29. doi: 10.22060/miscj.2014.532

[10] Panda, Sanjaya K., Indrajeet Gupta, and Prasanta K. Jana. "Task scheduling algorithms for multi-cloud systems: allocation-aware approach." Information Systems Frontiers 21 (2019): 241-259.

[11] ArtanMazrekaj, ArlindaSheholli, Dorian Minarolli, and Bernd Freisleben. 2023. The Experiential Heterogeneous Earliest Finish Time Algorithm for Task Scheduling in Clouds. In Proceedings of the 9th International Conference on Cloud Computing and Services Science (CLOSER 2019). SCITEPRESS - Science and Technology Publications, Lda, Setubal, PRT, 371–379. https://doi.org/10.5220/0007722203710379"

[12] S. T. Maguluri and R. Srikant, "Scheduling jobs with unknown duration in clouds," 2013 Proceedings IEEE INFOCOM, Turin, Italy, 2013, pp. 1887-1895, doi: 10.1109/INFCOM.2013.6566988.

[13] H. Kanemitsu, M. Hanada and H. Nakazato, "Clustering-Based Task Scheduling in a Large Number of Heterogeneous Processors," in IEEE Transactions on Parallel and Distributed Systems, vol. 27, no. 11, pp. 3144-3157, 1 Nov. 2016, doi: 10.1109/TPDS.2016.2526682.

[14] Zhou, Lan, and Sun Shixin. "Scheduling algorithm based on critical tasks in heterogeneous environments." Journal of Systems engineering and Electronics 19, no. 2 (2008): 398-404.

[15] Fan, L. Tao and J. Chen, ""Associated Task Scheduling Based on Dynamic Finish Time Prediction for Cloud Computing,"" in 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 2019 pp. 2005-2014. doi: 10.1109/ICDCS.2019.00198"

[16] Lei Shi, Jing Xu, Lunfei Wang, Jie Chen, ZhifengJin, Tao Ouyang, Juan Xu, Yuqi Fan, "Multijob Associated Task Scheduling for Cloud Computing Based on Task Duplication and Insertion", Wireless Communications and Mobile Computing, vol. 2021, Article ID 6631752, 13 pages, 2021. https://doi.org/10.1155/2021/6631752

[17] K. Dubey, M. Kumar and M. A. Chandra, "A priority based job scheduling algorithm using IBA and EASY algorithm for cloud metaschedular," 2015 International Conference on Advances in Computer Engineering and Applications, Ghaziabad, India, 2015, pp. 66-70, doi: 10.1109/ICACEA.2015.7164647.

[18] Ghanbari, Shamsollah, Mohamed Othman, MohdRizam Abu Bakar, and Wah June Leong. "Priority-based divisible load scheduling using analytical hierarchy process." Applied Mathematics & Information Sciences 9, no. 5 (2015): 2541.

[19] Kumar, V., Katti, C.P. and Saxena, P.C., 2014. A novel task scheduling algorithm for heterogeneous computing. International Journal of Computer Applications, 85(18).

[20] Guan Wang, Yuxin Wang, Hui Liu, He Guo, "HSIP: A Novel Task Scheduling Algorithm for Heterogeneous Computing", Scientific Programming, vol. 2016, Article ID 3676149, 11 pages, 2016. https://doi.org/10.1155/2016/3676149

[21] "T. Choudhari, M. Moh, and T. Moh, ""Prioritized task scheduling in fog computing."" 2018, In Proceedings of the ACMSE 2018 Conference (ACMSE '18). Association for Computing Machinery, New York, NY, USA, Article 22, 1–8. https://doi.org/10.1145/3190645.3190699"

[22] L. Yu, L. Chen, Z. Cai, H. Shen, Y. Liang and Y. Pan, "Stochastic Load Balancing for Virtual Resource Management in Datacenters," in IEEE Transactions on Cloud Computing, vol. 8, no. 2, pp. 459-472, 1 April-June 2020, doi: 10.1109/TCC.2016.2525984.

[23] Vijayasekaran, G., Duraipandian, M. An Efficient Clustering and Deep Learning Based Resource Scheduling for Edge Computing to Integrate Cloud-IoT. Wireless PersCommun 124, 2029–2044 (2022). https://doi.org/10.1007/s11277-021-09442-8

[24] Tang, X., Li, K. & Padua, D. Communication contention in APN list scheduling algorithm. Sci. China Ser. F-Inf. Sci. 52, 59–69 (2009). https://doi.org/10.1007/s11432-009-0010-3

[25] X. Zhu, L. T. Yang, H. Chen, J. Wang, S. Yin and X. Liu, "Real-Time Tasks Oriented Energy-Aware Scheduling in Virtualized Clouds," in IEEE Transactions on Cloud Computing, vol. 2, no. 2, pp. 168-180, April-June 2014, doi: 10.1109/TCC.2014.2310452.

[26] Zhou, Naqin, Deyu Qi, Xinyang Wang, ZhishuoZheng, and Weiwei Lin. "A list scheduling algorithm for heterogeneous systems based on a critical node cost table and pessimistic cost table." Concurrency and Computation: Practice and Experience 29, no. 5 (2017): e3944.

[27] M. Qasim, T. Iqbal, E. U. Munir, N. Tziritas, S. U. Khan and L. T. Yang, "Dynamic Mapping of Application Workflows in Heterogeneous Computing Environments," 2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), Lake Buena Vista, FL, USA, 2017, pp. 462-471, doi: 10.1109/IPDPSW.2017.129.

[28] W. Chen, G. Xie, R. Li, Y. Bai, C. Fan, and K. Li, "Efficient task scheduling for budget constrained parallel applications on heterogeneous cloud computing systems," Future Generation Computer Systems, 2017,Volume 74, Pages 1-11, ISSN 0167-739X, https://doi.org/10.1016/j.future.2017.03.008.

[29] K. He, X. Meng, Z. Pan, L. Yuan and P. Zhou, "A Novel Task-Duplication Based Clustering Algorithm for Heterogeneous Computing Environments," in IEEE Transactions on Parallel and Distributed Systems, vol. 30, no. 1, pp. 2-14, 1 Jan. 2019, doi: 10.1109/TPDS.2018.2851221.

[30] Ahmad, W, Alam, B. An efficient list scheduling algorithm with task duplication for scientific big data workflow in heterogeneous computing environments. Concurrency ComputatPractExper. 2021; 33:e5987. https://doi.org/10.1002/cpe.5987

[31] Dharmadhikari, Dipa D., and SharvariChandrashekharTamane. "Multi Objective Ameliorated Repetitive Resource Allocation Algorithm for Cloud Resource Scheduling and Allocation." In International Conference on Applications of Machine Intelligence and Data Analytics (ICAMIDA 2022), pp. 403-414. Atlantis Press, 2023.

[32] Hongyun Liu, Peng Chen, Xue Ouyang, HuiGao, Bing Yan, Paola Grosso, and Zhiming Zhao. 2023. Robustness challenges in Reinforcement Learning based time-critical cloud resource scheduling: A Meta-Learning based solution. Future Gener. Comput. Syst. 146, C (Sep 2023), 18–33. https://doi.org/10.1016/j.future.2023.03.029"

[33] F. Pop, C. Dobre and V. Cristea, "Genetic algorithm for DAG scheduling in Grid environments," 2009 IEEE 5th International Conference on Intelligent Computer Communication and Processing, Cluj-Napoca, Romania, 2009, pp. 299-305, doi: 10.1109/ICCP.2009.5284747.

[34] M. A. Rodriguez and R. Buyya, "Deadline Based Resource Provisioningand Scheduling Algorithm for Scientific Workflows on Clouds," in IEEE Transactions on Cloud Computing, vol. 2, no. 2, pp. 222-235, 1 April-June 2014, doi: 10.1109/TCC.2014.2314655.

[35] keshk, Arabi E., Ashraf B. El-Sisi, and Medhat A. Tawfeek. ―Cloud Task Scheduling for Load Balancing Based on Intelligent Strategy.‖ International Journal of Intelligent Systems and Applications 6, no. 5 (2014): 25–36. doi:10.5815/IJISA.2014.05.02.

[36] "YumingXu, Kenli Li, Jingtong Hu, Keqin Li, "A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues,", 2014, Information Sciences, Volume 270, Pages 255-287, ISSN 0020-0255, https://doi.org/10.1016/j.ins.2014.02.122."

[37] "Jiachen Yang, Bin Jiang, ZhihanLv, Kim-Kwang Raymond Choo, "A task scheduling algorithm considering game theory designed for energy management in cloud computing," 2020, Future Generation Computer Systems, Volume 105, Pages 985-992, ISSN 0167-739X, https://doi.org/10.1016/j.future.2017.03.024."

[38] Izadkhah, Habib. "Learning based genetic algorithm for task graph scheduling." Applied Computational Intelligence and Soft Computing 2019 (2019).

[39] Zhao Tong, Hongjian Chen, Xiaomei Deng, Kenli Li, Keqin Li, "A scheduling scheme in the cloud computing environment using deep Q-learning," 2020, Information Sciences, Volume 512, Pages 1170-1191, ISSN 0020-0255, https://doi.org/10.1016/j.ins.2019.10.035.

[40] Nanjappan, M, Albert, P. "Hybrid-based novel approach for resource scheduling using MCFCM and PSO in cloud computing environment. "Concurrency ComputatPractExper. 2022; 34:e5517. https://doi.org/10.1002/cpe.5517

[41] Karthiban, K., Raj, J.S. An efficient green computing fair resource allocation in cloud computing using modified deep reinforcement learning algorithm. Soft Comput 24, 14933–14942 (2020). https://doi.org/10.1007/s00500-020-04846-3

[42] Ahmad, W., Alam, B., Ahuja, S. et al. A dynamic VM provisioning and de-provisioning based cost-efficient deadline-aware scheduling algorithm for Big Data workflow applications in a cloud environment. Cluster Comput 24, 249–278 (2021). https://doi.org/10.1007/s10586-020-03100-7

[43] Z. Ali, S. Khaf, Z. H. Abbas, G. Abbas, F. Muhammad and S. Kim, "A Deep Learning Approach for Mobility-Aware and Energy-Efficient Resource Allocation in MEC," in IEEE Access, vol. 8, pp. 179530-179546, 2020, doi: 10.1109/ACCESS.2020.3028240.

[44] AminiMotlagh, A, Movaghar, A, Rahmani, AM. A new reliability-based task scheduling algorithm in cloud computing. Int J Commun Syst. 2022; 35(3):e5022. doi:10.1002/dac.5022

[45] Guo H, Zhou J, Gu H. Limited Duplication-Based List Scheduling Algorithm for Heterogeneous Computing System. Micromachines. 2022; 13(7):1067. https://doi.org/10.3390/mi13071067

[46] Mandal S. "Grey Wolf Optimizer based Resource Allocation and Optimization Algorithm in Cloud Computing Environment." Research Square; 2023. DOI: 10.21203/rs.3.rs-3144508/v1.

[47] Yin, Lei, Chang Sun, Ming Gao, Yadong Fang, Ming Li, and Fengyu Zhou. "Hyper-Heuristic Task Scheduling Algorithm Based on Reinforcement Learning in Cloud Computing." Intelligent Automation & Soft Computing 37, no. 2 (2023).

[48] M. Mezmaz, N. Melab, Y. Kessaci, Y.C. Lee, E.-G. Talbi, A.Y. Zomaya, D. Tuyttens, "A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems,"" 2011, Journal of Parallel and Distributed Computing, Volume 71, Issue 11, Pages 1497-1508, ISSN 0743-7315, https://doi.org/10.1016/j.jpdc.2011.04.007."

[49] Mansour, R.F., Alhumyani, H., Khalek, S.A. et al. Design of cultural emperor penguin optimizer for energy-efficient resource scheduling in green cloud computing environment. Cluster Comput 26, 575–586 (2023). https://doi.org/10.1007/s10586-022-03608-0,

[50] "H Sabireen, NeelanarayananVenkataraman, ""A Hybrid and Light Weight Metaheuristic Approach with Clustering for Multi-Objective Resource Scheduling and Application Placement in Fog Environment,"" 2023, Expert Systems with Applications, Volume 223, 119895, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2023.119895."