Received: 13 June 2024 | Accepted: 18 July 2024

DOI: 10.1049/tje2.12420

The Journal of Engineering

The Institution of Engineering and Technology WILEY

REVIEW

Critical review on resource scheduling in IaaS clouds: Taxonomy, issues, challenges, and future directions

Syed Hamid Hussain Madni[1] | **Muhammad Faheem**[2] | **Muhammad Younas**[3] | **Maidul Hasan Masum**[4] | **Sajid Shah**[4]

[1]School of Electronics and Computer Science, University of Southampton Malaysia, Johor Bahru, Johor, Malaysia

[2]School of Technology and Innovations, University of Vaasa, Vaasa, Finland

[3]Department of Computer Science, Government College University Faisalabad, Faisalabad, Pakistan

[4]Faculty of Computing, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia

Correspondence
Muhammad Faheem, School of Technology and Innovations, University of Vaasa, Vaasa 65200, Finland.
Email: muhammad.faheem@uwasa.fi

Abstract

In a cloud computing environment, the primary goal of resource scheduling is to reduce the economic expenditures for cloud users and grow fiscal achievement for cloud providers. In this article, the study of numerous forms of resource scheduling algorithms is presented that has been applied in IaaS clouds. The selected research articles are classified into six categories, according to the nature of the algorithm used. Further, the authors pointed out several issues and challenges with the help of the resource scheduling algorithms, the comparative performance metrics, and the simulation tools used to validate the several algorithms. The enhancement illustrates better performance concerns decreasing the cost and time while improving the competence and utilization of resources for IaaS clouds. These algorithms are executed in numerous simulation tools and real environments similar to CloudSim, SimGrid, MATLAB, and test-bed environments (practical implementation). This critical review and classification will serve as a foundation for further research in IaaS clouds in the Internet of Things (IoT) environments.

1 | INTRODUCTION

The field of cloud computing has become broader and broader, and it will be considered the fifth essential of humans in the future. Cloud computing is a current revolution, where cloud users can use resources at anytime, anywhere, and pay for them as per usage. In other words, it stores and processes the data and applications on/over the Internet instead of the user's computer hard drives via remote availability, high scalability, pay-as-you-go, and shared infrastructure [1, 2]. The fast movement towards cloud computing concerns ultimate facts for the achievement of communication, data availability, and integrity, information systems and security, public auditing, scientific application, and virtualization [3, 4].

In the cloud environment, IaaS providers organize a huge amount of IaaS resources of Infrastructure as a Service, which is allocated to cloud users on demand. The concept of IaaS is basically offering Hardware as a Service. IaaS deals with computing, storage, and network as standardized services over the cloud. CPUs, servers, VMs, storage systems, nodes, and networks (switches, routers, and other systems) are shared resources of IaaS and are made accessible to handle workloads [4, 5].

In particular, resource allocation and scheduling play a significant part in cloud computing. Moreover, its algorithms, strategies, and procedures directly affect the cost and performance of the cloud to improve the utilization of resources, execution competency, and energy consumption, maximize the cloud providers' profit, and satisfy the requirements of the cloud users' Quality of Service (QoS) [6, 7]. The most challenging problem is the resource scheduling to manage and provide the well-organized utilization of cloud resources for IaaS clouds. This problematic issue has attracted a lot of attention from researchers and industries in the previous few years [8, 9]. Hence, fluctuating resource demands, resource heterogeneity, environmental requirements, limited resources, and limitations of the locality require an effective and efficient resource scheduling procedure that must be applicable and suitable for IaaS clouds.

1.1 | Overview of resource scheduling

Resource scheduling is a process of scheduling that arranges tasks and resources. Pooled resources are accessible at a specific

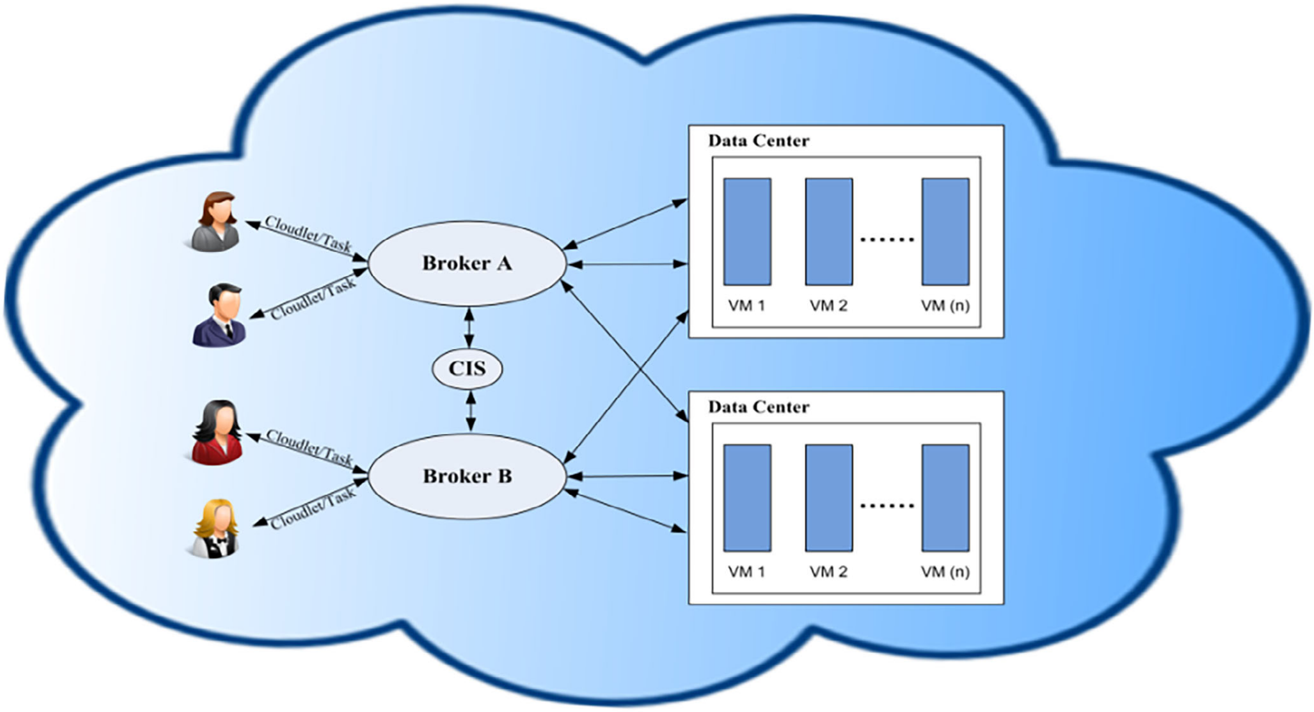

FIGURE 1 Resource scheduling in IaaS clouds.

time, and during this time, the tasks are planned for allocation to the cloud users. In simple words, it determines when the process should be started and ended, based on its period, precedent activities, precedence relationships, and resources assigned [10]. However, resource scheduling specifies the process of organizing and managing the resources amongst numerous cloud users regarding an explicit regulation of using the resources under the IaaS clouds. Scheduling tackles the difficulty of which resource is required to be assigned to the expected cloudlets or tasks [11].

Optimization is a procedure of finding the ideal solutions for a particular problem of attention [12]. In the case of scheduling, optimization algorithms are the techniques and methods to resolve the optimization problems in real-world applications to find optimality. However, it might not always be reachable [13]. In IaaS clouds, optimal resource scheduling is compulsory for the optimal utilization of cloud resources. How to schedule the distribution of resources depends upon various aspects: how many quantities of numerous resources have been requisite in general, how many cloud providers offer their services, how much is the ability of every accessible cloud provider, and many other conditions that influence either cloud computing or the quantity of available resources (despite in Figure 1).

Optimization is a procedure of finding the ideal solutions for a particular problem of attention. In the case of scheduling, optimization algorithms are the techniques and methods to resolve the optimization problems in real-world applications to find optimality. However, it might not always be reachable [13]. In IaaS clouds, optimal resource scheduling is compulsory for the optimal utilization of cloud resources. How to schedule the distribution of resources depends upon various aspects: how many quantities of numerous resources have been requisite in general, how many cloud providers offer their services, how much is the ability of every accessible cloud provider, and many other conditions that influence either cloud computing or the quantity of available resource (despite in Figure 1).

Cloud information service (CIS) is responsible for recording infrastructure resources and giving initial prices for those resources by pricing strategy. While the broker handles user requests and searches for the available resources in a particular cloud with the help of optimization algorithms. Once the required resources are located, they are allocated to the user requesting those resources [14]. Furthermore, the problematic issue is also subject to some constraints: a cloud provider delivers a finite amount of resources, the number of providers is fixed, and the cloud provider satisfies the number of available resources that cannot exceed by own capacity [15]. Thus, the existing schemes improve resource scheduling but do not consider the whole aspects or constraints. Since the cloud providers meet the demand of all cloud users or only some of them, determines the capacity of a limited resource, virtual machine's (VM) servers are selected for the distribution of resources.

This critical review presents the fundamental concepts and technologies regarding resource scheduling and the taxonomy of algorithms, and some performance metrics. It focuses on the open challenges and issues for resource scheduling in IaaS clouds. The primary objective of this critical review is to analyse and review the various algorithms used for resource scheduling in IaaS clouds. The forthcoming segments of this critical review are in the following organization: Section 2 presents some associated surveys and reviews of resource scheduling for IaaS clouds, with the motivation to further enhance research. Section 3 presents and analyses the taxonomy of resource

TABLE 1 Summaries of existing reviews for resource or tasks scheduling in IaaS clouds.

Previous review	Cloud computing	Resource or tasks scheduling	Algorithms: Non-heuristics	Algorithms: Meta-heuristics	Algorithms: Traditional	No of parameters	Evaluation tool and environment	Comparative analysis	Range
Cao et al. [17]	✓	✓	-	-	✓	3	✓	✓	2008–2013
Huang et al. [11]	✓	✓	-	✓	-	-	-	-	2008–2013
Abdulhamid et al. [18]	✓	✓	✓	-	✓	11	✓	-	2008–2014
Huang and Ou [19]	✓	✓	-	✓	-	-	-	✓	2008–2014
Ma et al. [6]	✓	✓	✓	✓	✓	5	-	-	2008–2014
Mathew et al. [20]	✓	✓	✓	✓	✓	8	-	✓	2008–2014
Tsai and Rodrigues [21]	✓	✓	-	✓	-	-	-	-	2008–2014
Kalra and Singh [22]	✓	✓	-	✓	-	8	✓	✓	2008–2015
Zhan et al. [23]	✓	✓	-	✓	-	8	✓	✓	2008–2015
Masdari et al. [24]	✓	✓	-	✓	-	7	-	✓	2008–2015
Liu and Qiu [25]	✓	✓	✓	✓	✓	5	-	-	2008–2016
Babu and Rajam [26]	✓	✓	✓	✓	-	6	-	-	2008–2016
Singh et al. [27]	✓	✓	-	✓	-	9	✓	✓	2008–2017
Kumar et al. [28]	✓	✓	-	✓	✓	12	✓	✓	2011–2018
Arunarani et al. [29]	✓	✓	-	✓		7	-	✓	2008–2018
Mahmoud et al. [30]	✓	✓	-	-	✓	6	-	✓	2015–2019
George and Pramila [31]	✓	✓	-	-	✓	-	-	-	2017–2019
Houssein et al. [32]	✓	✓	-	✓	✓	9	✓	✓	2011–2020
Singh et al. [33]	✓	✓	-	✓	-	7	-	✓	2009–2021
Rahimikhanghah et al. [34]	✓	✓	✓	✓	✓	3	✓	✓	2015–2021
Chaurasia et al. [35]	✓	✓	✓	✓	✓	9	✓	✓	2008–2021
Current article	✓	✓	✓	✓	✓	19	✓	✓	2008–2023

scheduling algorithms for IaaS clouds. Section 4 emphasizes the importance and description of performance metrics for resource scheduling. In Sections 5 and 6, we offer open issues and challenges of resource scheduling and conclusion.

2 | REVIEW AND MOTIVATION

Cloud computing technology is progressively used in businesses and the commercial market. A critical review demonstrates that dynamic resource scheduling is raising cloud providers' requirements for more cloud users. In IaaS clouds, an effective and efficient resource scheduling algorithm is essential for attaining revenue maximization for cloud providers and cloud user satisfaction [15, 16]. The scheduling problem arises in various research fields that evolve, along with the rising technology in new eras as well as in IaaS clouds.

2.1 | Existing reviews and surveys

Resource scheduling algorithms are familiar to different kinds of research and industry environments. This section summarizes the existing reviews and surveys of resource scheduling for IaaS clouds, as shown in Table 1. In existing reviews and surveys, algorithms are categorized into three types: meta-heuristics, traditional is based on rule-based algorithms, and non-heuristics are based on user-defined algorithms.

The crucial inspiration of the analysis Abdulhamid et al. [18] is to review the numerous theories and scheduling algorithms that functioned for on-demand Grid as a Service cloud in correspondence to resource scheduling parameters by the existing researcher in their research. Furthermore, Huang et al. [11] summarize some dynamic scheduling algorithm's procedures depending upon a threshold, improved ant colony algorithm for scheduling tasks, and optimized Genetic Algorithm (GA) with double suitability. Moreover, analyse resource allocation and job scheduling problems and define consistent solutions researchers anticipated under the cloud environment. Similarly, Mathew et al. [20] present a brief description of various existing task scheduling algorithms for cloud computing environments. A minor evaluation of different parameters is considered for scheduling in these algorithms also deliberated.

Besides, Tsai and Rodrigues [21] analysed the meta-heuristic scheduling algorithms in the context of challenges and issues for cloud computing and expanded its functionality via meta-heuristics. Further, Kalra and Singh [22] extensively review the applications of meta-heuristic algorithms in scheduling for cloud and grid environments. Heuristic algorithms are typically tardy than meta-heuristic algorithms, and the produced solutions are not optimized. Moreover, it delivers the guidelines for researchers to shift to meta-heuristics practices as a substitute for rule-based scheduling algorithms in IaaS clouds. Thus, most research is done to enhance the convergence speed and improve the solution's quality.

Correspondingly, Zhan et al. [23] present the categorization of management and scheduling cloud resources through the assessment of cloud computing architecture for resource scheduling algorithms. The classification contains three categories: the scheduling in the application, deployment, and virtualization level (layer). In each category, existing works, and notions of scheduling goals are concerned with the system, cloud user, and cloud provider. Future research challenges and directions are also pointed out for resource scheduling in IaaS clouds. Similarly, Ma et al. [6] present the five main issues in IaaS clouds: locality, energy, reliability, SaaS, and workflow-related resource allocation and scheduling. Moreover, these five issues are categorized according to performance and cost. Existing resource allocation and scheduling algorithms and policies are discoursed in detail according to specified parameters. In the last, some future directions are suggested for resource allocation and scheduling for IaaS clouds.

In the same way, Cao et al. [17] evaluate different job scheduling and VM allocation policies with threshold limitations for resource management for IaaS clouds. These policies include the first come first served (FCFS), SJNF, SJEF, LJNF, and LJEF for evaluating the waiting time, lease time, and cost utilization. Similarly, Mahmoud et al. [30] contrast the existing task scheduling algorithms founded on the heuristics approach in IaaS clouds. The comparison of a current algorithm depends upon the types of scheduling, including the static or dynamic, task dependency, scheduling goal, and heuristic used. Moreover, George and Pramila [31] present a survey of various user-defined techniques for resource allocation and scheduling for IaaS clouds in terms of their advantages and disadvantages.

Further, Huang and Ou [19] establish and analyse the Ant Colony Optimization (ACO), GA, and Particle Swarm Optimization (PSO) algorithms for task scheduling in IaaS clouds. Parallel and global search solution space is the specific part of GA; genetic iterations are hard to process when genetic entities are the same. PSO is fast in the primary phase, whereas slow convergence speed is in a different stage. ACO can slow convergence speed in the preliminary phase but is not suitable for the local search. Finally, summarize and suggest some future research directions. Moreover, Liu and Qiu [25] evaluate the VM scheduling solutions that consider the models, characteristics with heuristics, and meta-heuristics algorithms for cloud computing. VM scheduling policies are based on different schedules. These schedulers are classified based on five features: the QoS, energy, utilization, cost, and workload. Further, Babu and Rajam [26] provide a short survey on resource scheduling policies considering the price, time, VM, priority, SLA, and energy with non-heuristics and meta-heuristics (ACO, GA, and PSO) algorithms. Also, point out the challenges and issues that occurred in resource scheduling procedures in IaaS clouds.

Singh et al. [27] present a wide-ranging review of the various meta-heuristic algorithms for scheduling tasks in IaaS clouds. Most of these meta-heuristic algorithms are founded on bio-inspired and swarm intelligence approaches to improve the performances, including the PSO, Lion Algorithm (LA), GA, Bat Algorithm (BA), and ACO algorithms. For the optimization of task scheduling, cost, energy consumption, and time are considered. Similarly, Kumar et al. [28] observe and review the various scheduling algorithms and categorize them according to the problem addressed, such as resource provisioning and scheduling. The basic concepts and achievements of existing resource provisioning and scheduling algorithms are categorized into heuristic, meta-heuristic, and hybrid algorithms. In the same way, Singh et al. [33] focus on the working of several nature-inspired metaheuristic algorithms for task scheduling in IaaS clouds. Also, comparison analysis of PSO, Penguin Swarm Optimization Algorithm, GA, Crow Search Algorithm, ACO, and ABC algorithms are shown on the basis of performance metrics.

It is a widely accepted fact that the central aspect of tasks execution is scheduling. In this regard, Arunarani et al. [29] organized the task scheduling survey based on three perspectives: applications, methods, and utilized parameters. The survey is categorized into ten broad groups: ACO, Fuzzy, GA, PSO, QoS, Multiprocessor, Cost, Deadline, and other methods. The survey highlights the limitations of task scheduling in the IaaS Clouds environment. Similarly, Masdari et al. [24] analyse the PSO-based task scheduling algorithms in-depth with their objectives, properties, and limitations for IaaS clouds. More specifically, Houssein et al. [32] addressed the basic concepts of task scheduling according to the nature of the algorithm for IaaS clouds. Also, task scheduling algorithms are categorized on the objective functions, mapping schemes, and scheduling constraints, highlighting the existing challenges and potential solutions.

Rahimikhanghah et al. [34] present a systematic literature review by classifying the scheduling techniques for fog or cloud computing. Also, an extensive and systematic analysis of scheduling techniques is done based on performance, energy, and resource utilization with some improvements for future directions. However, Chaurasia et al. [35] present a systematic survey of existing energy-efficient techniques by examining the obstacles involved in enhancing the energy efficiency of cloud servers, as well as their limitations and issues faced in implementation.

2.2 | Motivations

This critical review will motivate future researchers to develop smarter and more secure optimal resource scheduling techniques and algorithms to strengthen and develop robust IaaS clouds. The benefits of IaaS cloud are numerous, including the ability to share physical infrastructure across several users. In addition, cloud customers can utilize IaaS to access resources through the Internet, and IaaS providers also provide pay-per-use IaaS services. All of this information will inspire and

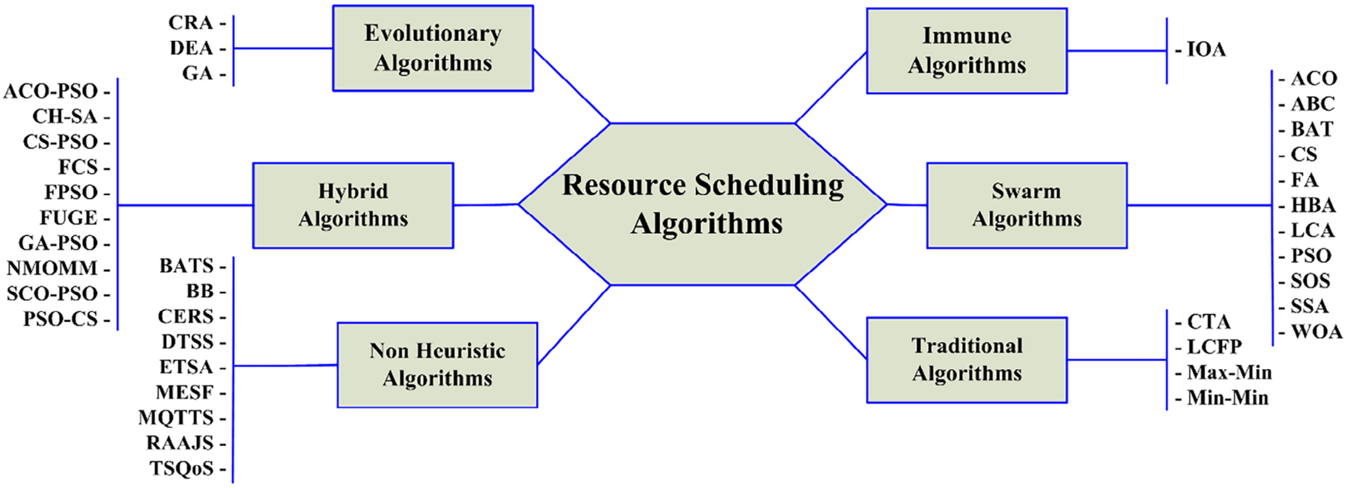

FIGURE 2 Taxonomy of resource scheduling algorithms for IaaS clouds.

urge future researchers to work for IaaS clouds with their advantages.

Resource scheduling is the practice of allocating and assigning resources among different cloud users in accordance with specific resource consumption standards and restrictions in a given IaaS cloud. This critical review identifies various difficulties and challenges by evaluating resource scheduling algorithms for IaaS Clouds, which helps to enhance the competency and utilization of resources for IaaS Clouds. It includes resource scheduling categorization techniques as well as an updated descriptive literature analysis for cloud computing research. Moreover, It examines and evaluates the many challenges and tactics utilized in resource scheduling strategies, highlighting their advantages and disadvantages. Furthermore, it emphasizes the performance measures that are used to assess existing literature. It also summarizes the proposed future works from past studies, which aids in determining the direction of present and future work.

3 | TAXONOMY OF RESOURCE SCHEDULING ALGORITHMS

Here is a taxonomy of the resource scheduling algorithms based on the popular algorithms used for resource scheduling in IaaS clouds, illustrated in Figure 2. Generally, overall one hundred and forty-three studies have been considered in this critical review. Conversely, 81 studies were carefully chosen that concentrated on the resource scheduling algorithms. The research methodology requires the relevant papers from various databases, including the Web of Science, Taylor & Francis, Springer, Scopus, Science Direct, Google Scholar, IEEE Explore, and ACM Digital Library. This research study contains the existing studies from 2008–2021. We define the search keys strings that give us the sufficient amount of most suitable related research studies: ("Resource Scheduling" + "Algorithm" + "Infrastructure as a Service" + "IaaS" + "Cloud Computing") and ("Resource Scheduling" AND "Algorithm" AND "Infrastructure as a Service" OR "IaaS" AND "Cloud Computing").

The selected research articles are classified into six categories according to the nature of the algorithms, including the evolutionary, immune, hybrid, non-heuristic, swarm, and traditional algorithms, as shown in Figure 2. The comparative analysis of these algorithms is presented according to the categorization in Tables 1–6, and their performance metrics are explained in Section 4 with the help of Table 7.

3.1 | Evolutionary algorithms

The evolutionary computation area is the study of evolutionary algorithms, where biological evolution inspires mechanisms used for optimization. This evolutional mechanism works based on genetically modified materials and the propagation of protein-like structures. The performance of evolutionary algorithms provides the optimal solutions to all types of approximate problems. These algorithms are most successfully used in arts, biology, chemistry, economics, marketing, research, physics, and computer science [36, 37]. In cloud computing, Differential Evolution Algorithm (DEA) introduced by [38] and GA announced by [39] are applied to the resource scheduling for IaaS clouds, shown in Table 2 with specific details.

3.1.1 | Chemical optimization algorithm

Chemical reaction optimization (CRO) is an innovative evolutionary algorithm that inspires the molecule's interaction to achieve the optimal energy state during a chemical reaction. In IaaS clouds, Zain and Yousif [40] introduce a CRO algorithm for job scheduling to reduce execution time. The simulation results reveal that the CRO algorithm works better than FCFS in the first scenario and cat swarm optimization (CSO) and glowworm swarm optimization (GSO) algorithms in the second scenario to reduce the execution cost.

TABLE 2 Evolutionary algorithms for resource scheduling.

Algorithms	References	Objectives	Performance metrics	Tools	Comparison algorithms	Achievements	Weaknesses
AGA	[49]	Task scheduling	Execution time	Not mentioned	GA	Improve the performance	Contrast with basic algorithms
CRO	[40]	Job scheduling	Execution time	CloudSim	FCFS, cat swarm optimization, and glow-worm swarm optimization	Improve the performance	Compare with Swarm Optimization algorithm and only focus on execution time
GA	[43]	Task scheduling	Time and resource utilization	Practically implemented	Nil	Improve the performance	Do not compare with other techniques
GA	[42]	Energy-aware resource scheduling	Resource utilization and cost	Not mentioned	PM, variance, and migration ratio	Improve the performance	Do not compare result with other techniques
GA	[44]	VM placement	Energy	Java	First fit decreasing	Improve the performance	No benchmark is used
GA	[45]	Optimal resource scheduling	Makespan and resource utilization	CloudSim	FCFS and RR	Improve the performance	Contrast with basic algorithms
GA	[47]	Multi-objective task scheduling	Execution time	CloudSim	Greedy approach and FCFS	Improve the performance	Consider only single objective
GA	[48]	Task scheduling	Cost and execution time	C++	Selection method and game theory	Improve the performance	Multiple tasks with one price and execution time vector.
GEC	[51]	Multi-objective task scheduling	Cost of rental VMs, degree of imbalance	CloudSim	FCFS, max-min, min-min, and SGA	Improve the performance	Only use single objective function
IDEA	[60]	Multi-objective task scheduling	Cost and time	Not mentioned	DEA, NSGA-II, SPEA2, and IBEA	Improve the performance	Consider only on time and cost
IGA	[46]	Task scheduling	Throughput and execution time	CloudSim	GA	Improve the throughput and reduce execution time	Contrast with basic algorithms
IGA	[41]	Optimal resource scheduling	Execution time	CloudSim	Min-min, max-min, and basic GA	Improve the performance	Contrast only with basic algorithms
MGGS	[50]	Task scheduling	Execution time, degree of imbalance, and response time	CloudSim	FCFS, GA, and min-min	Improve the performance	Compare with basic algorithms

3.1.2 | Differential evolution algorithm

To optimize resource allocation and task scheduling with the help of an improved differential evolution algorithm (IDEA) focused on the recommended time and cost of the model in IaaS cloud environment by [24], mutually to accomplish the general time and cost-effectively resource scheduling.

3.1.3 | Genetic algorithm

GA has elasticity, implicit parallelism, and universal optimization capability, not initiated in other algorithms. For optimal resource scheduling to realize practical tasks, Improved GA (IGA) is assumed in IaaS clouds for resource scheduling research, verifying the accuracy and reliability [41]. Chen et al. [42] suggest a policy for resource scheduling in IaaS clouds that depends on a GA with several specific parameters pursuing improving resource utilization and saving energy costs in the IaaS cloud environment.

In the study, Zhao et al. [43] familiarized a procedure of GA in task scheduling to train the memory controls and high demand of performance in IaaS clouds. The algorithm unusually focuses both resource utilization and time into attention, so the outcome is highly significant. Moreover, Wu et al. [44] proposed a GA for a current VM placement problem that assumes energy usage in mutually physical machines (PMs) as a single node and the whole network in the data centre.

Sindhu and Mukherjee [45] formulate a multi-objective GA that simultaneously enhances the application-centric (makespan) and resource-centric average CPU utilization. In the cloud, the proportion of VMs to hosts in a data centre fluctuates depending on time and workload. Several groupings of genetic operators are checked for the best one, which meets quicker and provides a favourable result, and are acknowledged.

To resolve task scheduling, Ma et al. [46] recommend an advanced and dynamic task scheduling algorithm based on IGA.

TABLE 3 Hybrid algorithms for resource scheduling.

Algorithms	References	Objectives	Performance metrics	Tools	Comparison algorithms	Achievements	Weaknesses
ACO-PSO	[57]	VM scheduling	Execution time and degree of imbalance	Practically implement using C language	ACO, ACS, PRACO, FCFS+RR, GA, and SA	Improve the performance	Focus on the load balancing
ACO-PSO	[55]	Optimal resource scheduling	Time and cost	MATLAB	ACO	Improve resource utilization	Contrast with basic algorithms
ACO-PSO	[56]	Optimal resource scheduling	Execution time	MATLAB	ACO	Improved performance	Contrast with only ACO basic algorithm
CHSA	[64]	Task scheduling	Cost, energy, and memory usage	Not mention	Cuckoo search and harmony search	Improve the performance	Compare with single objective function algorithms
CS-PSO	[62]	Deadline constrained task scheduling	Profit	CloudSim	SPSO and SLPSO	Enhance the performance	Focus only on the profit
CS-PSO	[63]	Task scheduling	Cost, deadline-driven, and makespan	CloudSim	ACO, min-min, and FCFS	Improve the Performance	Do not show the criteria of checking the multi-objective function of the algorithm
FCS	[69]	Resource scheduling	Failure rate	CloudSim	GA, HB, and PSO	Improve the Performance	Focus only on failure rate
FPSO	[67]	Job scheduling	Degree of imbalance, efficiency, execution time, and makespan	CloudSim	GA, modified GA, PSO, modified PSO, and FUGE	Improve the performance	Focus on only distributed systems
FUGE	[58]	Job scheduling	Makespan	Not Mention	ACO and Multiple ACO	Enhance the execution cost and time.	Do not consider the energy consumption
FUGE	[59]	Job scheduling	Makespan and cost	CloudSim	ACO and Multiple ACO	Improve the performance	Concentration on only execution cost
GA-ACO	[61]	Task scheduling	Failure rate and execution time	IM	GA and ACO	Improve the search efficiency	Contrast with basic algorithms
HGDGEO	[65]	Resource scheduling	VM cost during big data processing	Hadoop	FIFO, BFDHP, ILPEPS, and Morpheous	Reduce VM Cost	Do not consider the makespan and throughput
NMOMM	[66]	Task scheduling	Makespan and resource utilization	Code blocks version 15.12	Min-min and max-min	Enhance the performance	Contrast with basic traditional algorithms
PSO-SC	[68]	Task scheduling of workflow	Time, cost, and execution time	Not mention	FIFO	Improve the performance, including time, cost, and execution time	Contrast with traditional algorithms
PSGWO	[70]	Task scheduling	Waiting time	NetBeans	ACO and PSO	Improve the performance, makespan, execution time, energy, resource utilization, and efficiency	Compare with basic algorithms

TABLE 4 Immune algorithms for resource scheduling.

Algorithms	References	Objectives	Performance metrics	Tools	Comparison algorithms	Achievements	Weaknesses
IOA	[72]	QoS-aware resource scheduling	Resource utilization and response time	Eucalyptus platform	GA, artificial fish swarm algorithm (AFSA), and improved AFSA	Better performance	Focus only on cloud users

TABLE 5 Non-heuristic algorithms for resource scheduling.

Algorithms	References	Objectives	Performance metrics	Tools	Comparison algorithms	Achievements	Weaknesses
BATS	[73]	Task scheduling	Time	CloudSim	FBTS, COTS, and BOTS	Improve the efficiency and utilization	Focus on the time
BBA	[74]	Energy-aware resource scheduling	Resource utilization and energy	CloudSim	FCFS and best effort policy	Improve the performance	Contrast with the existing algorithm of CloudSim
CERS	[75]	Load-balanced resource scheduling	Time and cost	Practically implement	Greedy algorithm for joint workload scheduling and VM allocation	Improve time and cost	Comparison algorithm not mentioned clearly
DTSS	[62]	VM scheduling	Response time and waiting time	Not mentioned	FCFS and SJF	Improve the performance	Contrast with basic algorithms
ETSA	[76]	Task scheduling	Makespan	MATLAB	RR, DCLS, ETC, ECTC, and MaxUtil	Enhance the performance	Energy and execution cost are not considered
MESF	[77]	Energy-aware task scheduling	Response time and energy consumption	MATLAB	Random scheme	Improve the performance	Focus on only response time of the task
MQTTS	[78]	Task scheduling	Makespan and failure rate	CloudSim	Not mentioned clearly	Enhance the performance	Comparison algorithm not mentioned clearly
RAAJS	[79]	QoS-aware resource scheduling	Execution time and throughput	SimGrid	Min-min and max-min	Enhance the performance	Based on grid environment
TSQoS	[80]	Task scheduling	Execution time	CloudSim	Min-min and Burger Model	Improve the execution time	It is based on priority
TSQoS	[81]	Task scheduling	Cost	CloudSim	FCFS	Improve the performance	Compare with only traditional algorithm

Based on the GA, the suggested algorithm provides complete attention to the active appearances of IaaS clouds. Experimental consequences prove that the IGA effectively enhances the throughput and minimizes the execution time for task scheduling in IaaS cloud systems. Further, Agarwal and Srivastava [47] suggest the GA algorithm for the multi-objective optimization problem of task scheduling for IaaS clouds. GA assigns the given task efficiently to minimize the execution time, proved by the compassion of results.

Gawande et al. [48] discuss multiple task scheduling in an IaaS cloud environment for various users. For this ambition, a GA algorithm is proposed for scheduling the task with limited resources. The proposed GA algorithm is applied for optimal scheduling that maps the task to the available resources. The selection method of GA algorithms is extended for mapping the dependent functions with time and cost constraints. The simulation results show the outperformance of GA algorithms against the selection function and game-theoretic method. Furthermore, Mahmood and Khan [49] propose an adaptive GA (AGA) with appropriate mutation and crossover operations to schedule and allocate real-time tasks on heterogeneous VMs. AGA uses population diversity to control crossover's fitness, while the fitness of mutation maintains the capability to discover an optimal solution. The simulation consequences show that AGA performs better than the greedy and non-adaptive GA in providing better solution quality.

For task scheduling, Zhou et al. [50] modify the GA algorithm with the help of a greedy strategy and propose a modified GA with greedy strategy (MGGS) algorithm in an IaaS cloud environment. MGGS algorithm is improved according to the cloud providers and returns more investment. The simulation results demonstrate that the MGGS algorithm performs better than current algorithms in terms of execution time and response time while balancing the workload of VMs.

Devi and Valli [51] proposed the GA-based encoded chromosome (GEC) algorithm for scheduling in the IaaS cloud environment. The GEC algorithm predicts the suitable VMs required in the future and maps on the incoming requests are identified as tasks. The total cost of rental VMs is reduced based on the prediction results and also verified by simulation results.

3.2 | Hybrid algorithms

Hybrid algorithms are those algorithms that combine two or more algorithms to solve the same problematic issue. The ACO algorithm introduced by [52], cuckoo search (CS) announced by [53], GA, service cost optimization, and PSO publicized by [54] are united with other meta-heuristic or heuristic algorithms to enhance the performance and achieve better results for the resource scheduling for IaaS clouds. The summary of these algorithms is presented in Table 3.

TABLE 6 Swarm algorithms for resource scheduling.

Algorithms	References	Objectives	Performance metrics	Tools	Comparison algorithms	Achievements	Weaknesses
ACO	[89]	Task scheduling	Makespan	CloudSim	FCFS and RR	Enhance the performance	In contrast with heuristic algorithms
ACOLB	[91]	Load balancing	Execution time	CloudSim	FIFO, ACO, and ACO-RE	Enhance the performance	Neglect the cost
ACO-VMP	[93]	VM placement	Makespan, energy, and failure rate	CloudSim	BRS and random searching algorithm	Enhance the performance	Does not compare the results
Bat algorithm	[97]	Workflow scheduling	Cost	Not mentioned	Best resource selection algorithm	Reduce the execution cost	Does not consider the overall efficiency
BPSO	[116]	Resource provisioning	Energy, makespan, throughput, and execution cost	CloudSim	Bullet, TV_BPSO, and WRR	Improve the performance	Focus on resource allocation and multi-objective optimization
BULLET PSO	[127]	Resource scheduling	Energy, execution cost, and execution time	Not mentioned	PSO-HPC, PSO-SW, and PSO-DVFS	Enhance the efficiency	Do not compare with existing meta-heuristic algorithms
CMOSOS	[124]	Task scheduling	Cost and makespan	CloudSim	EMS-C, ECMSMOO, and BOGA	Improve the performance	Focus only on global search
CS	[99]	Task scheduling	Execution time	CloudSim	FIFO and greedy search	Improve the performance	Compare with heuristic algorithm
CS	[98]	Resource scheduling	Response time, makespan, and throughput	CloudSim	ACO	Enhance the performance	Compare with the traditional algorithm
CS	[100]	Job scheduling	CPU utilization and turnaround time	Haizea Scheduler	Not mentioned	Enhance the performance	Does not compare the results with existing algorithms
CSSA	[121]	Task scheduling	Cost, degree of imbalance, response time, and resource utilization	CloudSim	ABC, GA, and PSO	Improve the performance	In contrast with only basic algorithms
DSOSLS	[123]	Task scheduling	Makespan	Not mentioned	DSOS	Improve the performance	Contrast only with basic algorithms
DCLCA	[120]	Task scheduling	Makespan and failure rate	CloudSim	ACO, NSGA-II, max-min and MTCT	Enhance the performance	Focus on the local optimization
Enhanced BCA	[104]	Load balancing and task scheduling	Makespan and migration time	Not mentioned	Bee colony	Improve the performance	Contrast with basic algorithms
GBLCA	[119]	Task scheduling	Makespan and response time	CloudSim	ACO, GA, min-min, and max-min	Improve the performance	Focus on security
GPSO	[107]	Task scheduling	Execution time and workload	Not Mention	PSO	Improve the performance	Does not present the global search
GWOA	[126]	Resource provisioning	Execution time and cost	CloudSim	GWO, min-min, and FCFS	Improve the performance	Focus on resource allocation and compare with traditional algorithms
IABC	[96]	Load balancing	Execution time	CloudSim	ABC	Enhance the performance	Contrast only with artificial bee colony algorithm
Improved Firefly	[102]	Task scheduling	Resource utilization	CloudSim	Artificial firefly algorithm	Enhance the performance	Contrast only with basic algorithm
IPSO	[115]	Multi-objective task scheduling	Cost, degree of imbalance, and makespan	Not mentioned	Round Off PSO and smallest position value PSO	Improve the performance	Do not check the criteria of multi-objective algorithm

(Continues)

TABLE 6 (Continued)

Algorithms	References	Objectives	Performance metrics	Tools	Comparison algorithms	Achievements	Weaknesses
IPSO (LJFP-PSO and MCT-PSO)	[117]	Task scheduling	Degree of imbalance, energy consumption, execution time, and makespan	MATLAB	MCT, LJFP, SJFP, PSO, min-min and max-min	Improve the initialization	Contrast only with basic algorithms
LBACO	[92]	Task scheduling	Makespan	CloudSim	FCFS and ACO	Enhance the performance	Tasks are mutually independent
LCA	[118]	Task scheduling	Makespan	CloudSim	First come first served (FCFS), best effort first (BEF), and last job first (LJF)	Improve the execution time	Contrast only with basic algorithms
LBPSO	[128]	Reliability and availability-aware task scheduling	Cost, makespan, execution time, and round-trip time	CloudSim	ETLongest, random, MPSO, and SPSO	Improve the performance	Comparison algorithms are not mentioned clearly and do not consider the parameters for reliability and availability
Load and Fault Aware Honey Bee Scheduling Algorithm	[103]	Fault-aware resource scheduling	Failure rate	CloudSim	Basic load aware honey bee	Enhance the performance	Contrast only with a basic algorithm
MOACO	[95]	VM scheduling	Execution time	CloudSim	Random allocation	Enhance the performance	Contrast with a traditional algorithm for allocation
MOACO	[94]	Task scheduling	Makespan, cost, and resource utilization	CloudSim	ACO, min-min, and FCFS	Improve the performance	Contrast with traditional algorithms
MOPSO	[112]	Task scheduling	Execution time and cost	CloudSim	PSO	Improve the QoS	Contrast with basic algorithms
MOPSO	[113]	Task scheduling	Time and energy	C++ and CloudSim	BRS and random searching algorithm	Enhance the performance	Comparison algorithms are not strong
Modified PSO	[108]	Task scheduling	Makespan	Eucalyptus cloud	GA and PSO	Enhance the performance	In testing every time use only four resources
PSO	[105]	Job scheduling	Cost	CloudSim	GA and random scheduling algorithm	Improve the performance	Focus only cloud provider
PSO	[106]	VM allocation	Resource utilization	Not mentioned	Best fit, first fit, and worst fit	Enhance the performance	Compare with traditional algorithms
PSO	[109]	Resource scheduling in data centre	Time	CloudSim	Not compare	Improve the performance	Concentration on only energy consumption
PSO	[110]	Cloudlet scheduling	Makespan	CloudSim	Numbers of cloudlets	Improve the makespan	Compare performance in different ranges only
PSO	[111]	Workflow application scheduling	Time and cost	CloudSim	Best resource selection and PSO	Improve the performance	Does not compare with existing meta-heuristic algorithms
RNPSO	[114]	Cost minimizing and workflow scheduling	Execution time and execution cost	Not mentioned	PSO	Improve the performance	In contrast with only basic algorithms
SACO	[90]	Task scheduling	Execution time and Makespan	Not mentioned	ACO and IACO	Improve the performance	In contrast with only basic algorithms
SOS	[122]	Task scheduling	Makespan, execution time, and degree of imbalance	CloudSim	PSO	Improve the performance	Focus also on the load balancing
WOA	[125]	Task scheduling	Cost	MATLAB	ACO and PSO	Improved the convergence rate	Does not focus on multi-objective optimization

TABLE 7 Traditional algorithms for resource scheduling.

Algorithms	References	Objectives	Performance metrics	Tools	Comparison algorithms	Achievements	Weaknesses
Elastic Cloud Max-min	[132]	Task scheduling	Waiting time and response time	CloudSim	RR and max-min	Enhance the performance	Does not consider load balancing parameter
Improved Max-min	[133]	Task scheduling	Makespan	Not implemented	Max-min, min-min and RASA	Enhance the performance	Theoretically analysis
Improved Max-min	[134]	Task scheduling	Response time	CloudSim	Max-min and RR	Enhance the performance	Does not consider load balancing parameter
Improved Min-min	[135]	Task scheduling	Execution time and load balancing	CloudSim	Min-min	Better performance	Less utilization rate
LCFP	[131]	Task scheduling	Makespan	Eucalyptus cloud	Shortest cloudlet fastest processing and FCFS	Better performance	Contrast with basic algorithms
PTCT	[136]	Task scheduling	Makespan and execution time	MATLAB	Max-min, min-max, min-min, and QoS-guide	Enhance the performance	Based on static scheduling

3.2.1 | ACO-PSO

Wen et al. [55] make PSO inoculated into ACO and introduce crossover tactics and mutation strategies in GA to reduce the imperfection that the algorithm is easy to lie in a local optimum solution for resource scheduling IaaS clouds. Results express that ACO-PSO accelerates the convergence speed, also neglected to lie into optimal local solution. The user task efficiently provides valuable resources to improve the resource utilization ratio in IaaS clouds. Further, Yang [56] introduces the ACO based on the PSO algorithm and produces the optimum solutions with the PSO algorithm's help to do mutation and crossover operation to escape trapping in local optimal. In conclusion, the experiment results indicate that the ACO-PSO algorithm is coming together, which is more appropriate for IaaS cloud-based resource scheduling.

Cho et al. [57] propose an algorithm ACO-PSO for VM scheduling intended for load balancing. ACO-PSO attends modified VM's demands where three resources, including the CPU, disk utilization, and storage, are considered load balancing. To reduce the time of scheduling, pre-reject modules functioned to decrease the solutions' measurements and added the PSO operator to maximize the convergence speed for the scheduling procedure of ACO. The ACO-PSO is implemented with finite information and practices the workload of ancient demands to expect new input demands. To appraise an algorithm's performance for load balancing, the traditional methods of using only fixed task sets and single scheduling are used for comparison and show better performance.

3.2.2 | FUGE

A hybrid algorithm, known as FUGE, is founded on fuzzy theory and GA that aims to accomplish optimum load balancing contemplating cost and execution time [58]. However, it adapts the standard GA (SGA) and applies fuzzy theory to formulate a fuzzy-based steady-state GA to enhance SGA's performance in terms of makespan. Furthermore, Javanmardi et al. [59] also recommend a job scheduling hybrid algorithm that works for the load balancing of the task and decreases the entire cost and time for execution. With the help of IGA, it tries to minimize the repetition of generating a population with the fuzzy theory's support.

3.2.3 | GA-ACO

Liu et al. [61] recommend an algorithm that is formulated on GA and ACO for task scheduling. It benefits the strong optimistic response of ACO on the algorithm's convergence rate account. However, the selection of primary pheromones is a critical influence on the convergence rate. The technique enhances the function of the global search capability of GA to resolve the optimum result rapidly and then changes it into the primary pheromone of ACO. The experiment results prove that the proposed GA-ACO overweighs GA and ACO, exhibiting better proficiency benefits on a large scale under the same IaaS clouds.

3.2.4 | Cuckoo search driven particle swarm optimization

The primary goal is to improve the internal utilization of IaaS clouds and enhance the price of outsourced tasks to the external clouds. This problematic issue is resolved using cuckoo search-driven PSO (CS-PSO) algorithm and articulated as an integer programming standard. It performs the local search extra professionally, and it neglects the local optima issue of PSO. The tasks' runtime utilization and the cost for each job are calculated with the help of CloudSim. Simulation results express that the average profit of the CS-PSO algorithm is better than the standard PSO and self-adaptive learning PSO (SLPSO) for non-trivial size issues [62]. Similarly, Jacob and Pradeep [63] propose combining two optimization algorithms, including the CS and PSO, for the multi-objective task scheduling in IaaS clouds. CS-PSO algorithm is applied to resolve the multi-objective task scheduling issue by reducing the cost, deadline-driven, and makespan in the heterogeneous IaaS clouds.

3.2.5 | Cuckoo harmony search algorithm

Pradeep and Jacob [64] propose the hybrid cuckoo harmony search algorithm (CHSA) for multi-objective optimization in IaaS clouds. Multi-objective CHSA task scheduling algorithm is compared with single objective function algorithms to enhance performance by reducing the cost, energy consumption, and memory usage.

3.2.6 | Gradient descent golden eagle optimization

For Hadoop heterogenous clouds, a hybrid gradient descent golden eagle optimization (HGDGEO) algorithm is proposed for resource scheduling during big data processing. The main objective of resource scheduling is reducing the VM cost. Simulation results show that HGDGEO algorithm performs better than benchmark algorithms in terms of job completion time and VM cost for heterogenous clouds [65].

3.2.7 | Min-max normalization

Multi-cloud computing is combined with the current trends of cloud computing. Although each cloud has scheduling policies and schedulers, these are difficult in the multi-cloud environment. Gajera et al. [66] suggest a normalized multi-objective min-min max-min (NMOMM) algorithm for task scheduling in IaaS clouds, which considers the two parameters together makespan and resource utilization. The results are compared with different weights at different benchmarks, which show the effectiveness of the algorithm.

3.2.8 | Fuzzy system and modified PSO

Mansouri et al. [67] propose a fusion algorithm by combining the fuzzy approach and PSO (FPSO) algorithm for balancing the DI while reducing the execution time and makespan. In the HFPSO algorithm, the fuzzy method is used to calculate the fitness functions. HFPSO is also applied to minimize the optimization performance with the help of mutation and crossover operations.

3.2.9 | Service cost optimization algorithm based on PSO

Xue et al. [68] propose the service cost optimization algorithm formulated on particle swarm optimization (PSO-SC) algorithm to execute the task scheduling of workflow. Simulation results demonstrate that the PSO-SC algorithm diminishes the total time to complete the task. Simultaneously, it adjusts the execution task based on the processing power of the VMs in the data centres and allocates these VMs realistically with minimum achieved cost for users.

3.2.10 | Fuzzy cuckoo search

Madni et al. [69] propose an advanced fuzzy cuckoo search (FCS) algorithm for IaaS clouds for the reliability-aware resource scheduling problem. It enhances the QoS for allocating the particular cloudlet or task to the explicit VMs by improving the failure rate that helps to increase resource utilization for cloud providers. The simulation results show FCS's comparison with existing algorithms and enhance the performance in terms of failure rate.

3.2.11 | Particle swarm grey wolf optimization (PSGWO)

Khan and Santhosh [70] focused on task scheduling in IaaS cloud environment and propose a new hybrid optimization algorithm based on particle swarm grey wolf optimization (PSGWO). PSGWO scheduling algorithm is experimentally verified and compared with conventional ACO and PSO algorithms that scheduled the jobs with minimum waiting time. Hybrid PSGWO prioritizes tasks based on the trust level of virtual machines and achieves better performance with the optimal makespan, execution time, energy, resource utilization and efficiency compared to ACO and PSO algorithms.

3.3 | Immune algorithms

Immunology is the study of the biological and artificial immune systems. A similar process in computational methods refers to resistant algorithms. The immune system's primary function is to protect the body's organs from external threats, like foreign toxic substances and pathogens from the family of microorganisms like bacteria, viruses, parasites, and pollen. There are two main functions of the immune system, including detecting and removing potentially harmful pathogens from the body. And the second one is the body recovery and maintenance system [36, 37]. Here, immune optimization algorithm (IOA) is introduced by [71] and used for resource scheduling in IaaS clouds, presented in Table 4.

3.3.1 | Immune optimization algorithm

The immune optimization algorithm (IOA) of multidimensional QoS proposed by [72] effectively accomplishes rapid multi-objective optimization in calculating application preferences and utilizing multidimensional QoS. The outcomes of simulations express a weighty enhancement in the average rate of equipment utilization, response time, and service time as associated with comparable algorithms.

3.4 | Non-heuristic algorithms

Non-heuristic algorithms for resource scheduling in IaaS clouds include bandwidth-aware task scheduling (BATS), cost-effective resource scheduling (CERS), deadline-aware two-stage scheduling (DTSS), multi-QoS and trusted task scheduling (MQTTS), task scheduling QoS, resource allocation and adaptive job scheduling (RAAJS), bin balancing (BB), and most efficient server first (MESF). The comparative analysis of non-heuristic algorithms is presented in Table 5.

3.4.1 | Bandwidth aware task scheduling algorithm

A non-linear programming approach is applied to settle the bounded multi-port model and successfully build an innovative task scheduling technique, the bandwidth-aware task scheduling (BATS) algorithm. It assigns the appropriate number of tasks to every VM depending on its processing power, network speed, and storage space. In comparison, the previous task scheduling algorithms measure the availability of processing and storage resources for task scheduling in cloud computing. Only a few algorithms concentrated on the constraint of network speed in task scheduling. No one pursued to improve the utilization of all three resources [73].

3.4.2 | Bin balancing algorithm

The BBA algorithm fulfils minimal energy consumption objectives, high efficiency, and load balancing considering the task deadline, host processing elements, memory, and bandwidth. The simulation results demonstrate that the BBA performed better than existing CloudSim scheduling algorithms [74].

3.4.3 | Cost-effective resource scheduling algorithm

The study presents the load-balanced resource scheduling also indicates a minimized actual cost. A modified CERS algorithm is recommended that reduces the entire cost of the resource. Several simulation outcomes confirmed that an excellent load-balanced architecture signifies enhanced time, failure rate, response time, throughput, and energy wastage. Also, the algorithm deals with the energy efficiency perception, which moves towards green computing [75].

3.4.4 | Deadline aware two-stage scheduling algorithm

DTSS algorithm is used to organize VMs for the demanded jobs expected from cloud users in IaaS clouds. In this method, every job needs two types of VMs to execute its task. This algorithm assigns VMs as a resource to the demanded jobs depending upon scheduling the appointments and execution time given deadlines according to response time and waiting time [62].

3.4.5 | Energy-efficient task scheduling algorithm

Panda and Jana [76] introduced an energy-efficient task scheduling algorithm (ETSA) in a heterogeneous IaaS cloud to reduce carbon emissions and energy consumption. The purpose of the ETSA algorithm is to balance energy consumption and resource utilization from the viewpoint of cloud providers with task scheduling. The ETSA algorithm attains better results in optimal makespan and energy consumption due to the normalization, selection, and execution phases.

3.4.6 | Most efficient server first

The recommended MESF scheduling algorithm organizes tasks for the utmost servers with energy efficiency in a data centre. The proposed MESF algorithm reduces the average task response time and the server-related energy cost. The system functioning with MESF is uncertainly established under task appearance with exponential dispersal. The performance of the MESF was appraised and contrasted with that of a randomly based task scheduling algorithm via the MATLAB tool. Experimental outcomes express that MESF performs better than the comparison algorithm, saving energy, cost of the extended task, and response time, although contained by the extreme constraint [77].

3.4.7 | Multi QoS and trusted task scheduling algorithm

A useful MQTTS resource management model and algorithm in the environment of IaaS clouds are presented by [78]. Due to fuzzy clustering being a multi-variate clustering method, the suggested approach is provided with a sound of multi-QoS performance.

3.4.8 | Resource allocation and adaptive job scheduling algorithm

Kumar et al. [79] recommend a three-step structure, a new RAAJS scheduling algorithm, scheduling users' demands with pooled resources, and a resource selection method, which increases the QoS distributed by the cloud. The suggested strategy maximizes the reliability of resource accessibility for a job and minimizes the execution time, enhancing the quality of services supplied to cloud users.

3.4.9 | Task scheduling quality of service algorithm

Wu et al. [80] propose a task scheduling algorithm that relies on QoS, which combines numerous task attributes, including user anticipation, privileges, expectations, duration of the task, and remaining time for the finishing. Also, it supports scheduling each task on service with the shortest execution time. Moreover, Bansal et al. [81] evaluate the cost of QoS driven task scheduling algorithm for task scheduling in IaaS clouds.

3.5 | Swarm algorithms

The swarm intelligence is the area of the computational system. It is motivated by sharing intelligence and collaboration between large numbers of identical agents in a single environment. The everyday examples are bird flocks, fish schooling, and ant colonies. In these examples, they shared the intelligence of self-organizing and decentralized the information. This problem-solving method is typical for relocating the colonies hiding from enemies, and searching for food. The participants from these problems share knowledge of the environment to efficiently react and solve the same problem. The swarm intelligence algorithm uses the same strategies of adapting problem solutions in its domain [36, 37]. Table 6 shows the analysis of swarm algorithms for resource scheduling in IaaS clouds.

Here, ant colony optimization (ACO) [52], artificial bee colony (ABC) [82], bat algorithm [83], firefly algorithm [84, 85], league championship algorithm (LCA) [86], particle swarm optimization (PSO) [87], and symbiotic organism search (SOS) [88] are considered as swarm algorithms for resource scheduling in IaaS clouds.

3.5.1 | Ant colony optimization

Tawfeek et al. [89] use the ACO to discover the optimum resource allocation for tasks in the dynamic cloud to reduce the makespan of tasks on the whole system. ACO is an arbitrary optimization search algorithm that is applied for assigning the arriving jobs to the VMs. The scheduling strategy is executed via CloudSim. Simulation results compared with FCFS and RR algorithms and exposure to the ACO satisfy anticipation. Further, Moon et al. [90] suggest the slave ACO (SACO) algorithm for scheduling the cloudlets to VMs in the IaaS cloud environment. SACO avoids long paths whose pheromones are wrong by adapting diversification and reinforcement strategies. SACO uses minimal preprocessing overheads and performs better than the existing ACO algorithm while maximizing the cloud server's utilization.

Xue et al. [91] propose the ACO load balance (ACOLB) algorithm for the load inequity of VM in the progression of task scheduling, and it familiarizes to the dynamically in IaaS clouds. Simultaneously, a Load Balancing ACO (LBACO) algorithm is used to dynamically examine the ideal resource scheduling for each task in IaaS clouds. It reduces the makespan of a particular group of tasks and gets used to the dynamic IaaS clouds and stability of the workload of the whole system [92]. However, it is essential to allocate professionally virtual machines to physical machines and enhance resource utilization. Liu et al. [93] suggest an algorithm ACO virtual machine placement (ACO-VMP), for reducing the number of servers used.

Multi-objective ACO (MOACO) Scheduling algorithm enhances the performance and cost as the expenditure constraints of task scheduling's optimization issue. Also, MOACO algorithm deals with multi-objective optimization, including the cost, makespan, resource utilization, and time deadline [94]. MOACO is used for IaaS cloud resource scheduling regarding the actual QoS parameters necessity of IaaS cloud environment. To reduce data centre resource management imperfections in IaaS clouds and confirm that IaaS clouds provide enhanced QoS services. The experiment results indicate that the proposed algorithm increases resource utilization and minimizes processing time [95].

3.5.2 | Artificial bee colony algorithm

Based on the interaction ABC algorithm (IABC), a load balance procedure is offered to resolve the problematic workload issues in IaaS clouds. A large multitude of simulations shows the proficiency of the algorithm. The proposed IABC algorithm enhances the performance and efficiently schedules the task to VMs with minimum execution time [96].

3.5.3 | Bat algorithm

Raghavan et al. [97] applied the Bat algorithm with the support of the binary bat algorithm for workflow scheduling in IaaS clouds. Precisely the mapping of tasks and resources is completed with the proposed approach in IaaS clouds. The consequences show that the proposed BA performed fifty percent higher than the comparison algorithm to minimize the overall cost.

3.5.4 | Cuckoo search algorithm

For the optimization, Madni and Abd Latiff [98] introduce the cuckoo search (CS) based resource scheduling meta-heuristic algorithm for IaaS clouds. Simulation results demonstrate that the suggested CS algorithm achieves better optimal solutions than the ACO algorithm regarding reducing the makespan, response time, and throughput by using the Levy Flight function for searching. Similarly, Agarwal and Srivastava [99] present the CS-based task scheduling algorithm for IaaS cloud environments. The robust search characteristic using the Levy Flight function is used in the CS algorithm for optimizing the execution time of the cloudlets or tasks in the heterogeneous IaaS clouds. Similarly, Aloboud and Kurdi [100] propose the CS algorithm for job scheduling in IaaS clouds regarding CPU utilization and turnaround time. In the same way, Madni et al. [101] hybrid CS algorithm with a gradient decent approach for optimizing the convergence rate.

3.5.5 | Firefly algorithm

Zhao et al. [102] establish an improved intelligence firefly algorithm for task scheduling with the firefly algorithm's help in IaaS clouds to discover the superior result and turn it into the primary pheromone of the enhanced firefly algorithm for precision in prediction and competent searching.

3.5.6 | Honey bee algorithm

To allocate resources so that the request is fulfilled with minimum cost and maximum quality of service, it can be provided to cloud users. Consequently, honey bee scheduling algorithm awareness of load and fault is anticipated for IaaS clouds. This algorithm considers the fault rate and loads on a data centre to improve the QoS and performance in the IaaS clouds [103]. In the study, the BCA is enhanced, and it is applied to schedule efficiently and balance the load between nodes in dynamic and vigorous cloud environments. The enhanced BCA algorithm attempts to achieve minimum makespan and migration, providing QoS to the cloud users [104].

3.5.7 | Particle swarm optimization

A job scheduling model depends upon the PSO algorithm recognized for IaaS clouds to decrease energy consumption and increase revenue for cloud providers [105]. Further, the strategy is also based on PSO presented by Kumar and Raza [106], emphasizing the efficient VM distribution to PMs to reduce the total wastage of resources and the number of PMs used. Hence, Zhong et al. [107] introduce the greedy PSO (GPSO) algorithm for resolving the task scheduling problem in IaaS clouds. GPSO helps to decrease the overall competition time and balance loads of each VM. GPSO uses a greedy approach to rapidly resolve the initial particle value of PSO consequent from the VMs. The simulation results determine that the GPSO algorithm performs better to provide a faster convergence rate, balanced workload, and strong local and global search competencies.

The modified PSO algorithm is recommended to enhance the performance of the standard PSO algorithm. The shortest job first perform algorithm is combined with the basic PSO algorithm for producing an initial population to decrease makespan [108]. The performance of the resource scheduling strategy in IaaS clouds is affected by the data centre's performance and electricity usage directly by using the PSO algorithm. The resource scheduling problem is applied in complexity research and finds high reliability, performance, and scalability; lastly, the user's necessities associated with the field of IaaS clouds are achieved [109].

The advantage of using the pay-per-use of clouds is impacted by the less utilization of the already reserved resources. For this purpose, maximum resource utilization while concurrently reducing makespan gives cloud providers immense attention to minimize cost. To resolve this issue, Al-Olimat et al. [110] present a way out for refining the makespan scheduled tasks and the resource utilization of the cloud with the help of the famous population-based metaheuristic algorithm PSO. It is applied inside the CloudSim tool to enhance the work of the previously executed simple broker.

Zhang et al. [111] suggest innovative modelling for optimum user application scheduling for IaaS clouds to cloud resources, considering execution cost and current workload. A novel inertia weight is also announced to effectively achieve the global and local search while neglecting to plunge into the local optimum. In conclusion, simulation results show better performance and convergence analysis.

An inclusive multi-objective PSO (MOPSO) algorithm for enhancing task scheduling is designed by [112] to minimize the task's transferring time, execution time, and execution cost. Jswarm package is extended to a multi-objective Jswarm (MO-Jswarm) package for implementation and evaluation. CloudSim achieves the results showing that the MOPSO can discover optimal solutions for multi-objective task scheduling problematic issues.

Further, Jena [113] also presents the MOPSO algorithm to resolve the task scheduling problem under the IaaS clouds, where the number of data centre and user jobs changes dynamically. In the fluctuating environment of cloud computing, resources need to function in optimal manners. Consequently, MOPSO is suitable because it can effectively utilize the cloud's resources to minimize energy consumption and makespan. The simulation results demonstrate that MOPSO performs superior to the BRS and RSA. Moreover, Li et al. [114] designed a renumber PSO (RNPSO) for cost-minimizing and workflow scheduling by encoding the particle position for reordering and renumbering the resources according to their computational ability. In this way, particle position values are made logically and the difference between the well and poor performance.

The study's main purpose is to improve the general performance of task scheduling problems in IaaS clouds. The proposed algorithm is designed on a heuristic algorithm using load balancing mutation PSO (LBMPSO). The LBMPSO is suggested to accomplish the reliability of task scheduling. LBMPSO considers execution time, makespan, load balancing, round trip time, transmission cost, and transmission time between tasks and VMs [8]. Similarly, Beegom and Rajasree [115] propose an Integer PSO algorithm to optimize a single objective function and multiple objective functions for task scheduling in IaaS clouds. Simulation results demonstrate that IPSO algorithm is better in terms of reducing convergence rate and load balancing in different scenarios of task traffic.

A binary PSO (BPSO) algorithm is proposed for resource allocation and scheduling in heterogeneous environments of IaaS clouds by Kumar et al. [116]. The BPSO algorithm is improved for the multi-objective optimization by conflicting the Pareto optimality and non-dominated solutions. The simulation results demonstrate that the BPSO algorithm optimizes the cloud users' parameters compared to selected existing algorithms. However, Alsaidy et al. [117] proposed an Improved PSO algorithm by enhancing the initialization that employs LJFP and MCT heuristic algorithms. Simulation results show that LJFP-PSO and MCT-PSO performed better than comparative algorithms in terms of degree of imbalance, energy consumption, execution time, and makespan.

3.5.8 | League championship algorithm

The NP-hard problem, which minimizes the makespan of task scheduling in IaaS clouds, is known to resolve different algorithms in the past. Still, Abdulhamid et al. [118] apply the league championship algorithm (LCA) to solve this issue. The LCA aids cloud users in saving the costs for the time used. Further, Abdulhamid et al. [119] enhance the LCA algorithm to solve the non-deterministic polynomial-time problematic issue of secure global task scheduling. Global LCA (GBLCA) algorithm is constructed to schedule the tasks of the cloud users to the frights of on-demand IaaS clouds' resources in an optimum manner. Moreover, Abdulhamid et al. [120], a dynamic clustering LCA (DCLCA) scheduling algorithm for addressing the cloud task execution regarding fault tolerance awareness, reveals the currently accessible resources and decreases the failure rate of independent tasks.

3.5.9 | Social spider algorithm

A chaotic version of the Social Spider Algorithm (CSSA) is introduced to resolve the task scheduling of VMs in the heterogeneous environment of IaaS clouds by Xavier and Annadurai [121]. By effective load balancing, the prime objective is to lower the makespan with the help of chaotic, random weight selection. In comparing simulation results, CSSA performs better than selected comparison algorithms for improving scheduling efficiency that is suitable for reducing the cloud user's expenditures.

3.5.10 | Symbiotic organism search

IaaS clouds' task scheduling problem is solved by applying a discrete version of the symbiotic organism search (DSOS) algorithm. The symbiotic organism search (SOS) is suggested by Abdullahi and Ngadi [122]. In the ecosystem, the shared communication among organisms inspires; thus, a task scheduling issue is a type of discrete optimization identified to be NP-complete. The execution of the DSOS algorithm is assessed via four statistical distributions of established data sets. The proposed algorithm surpassed the well-known optimization problem solver algorithms such as PSO. Further, Kumar et al. [123] suggest the DSOS with local search (DSOSLS) algorithm for obtaining the lowest makespan time during the VM scheduling. The simulation results reveal that the proposed DSOSLS diminishes the makespan time with population size variation and the number of irritations. Moreover, Abdullahi et al. [124] propose the chaotic multi-objective SOS (CMOSOS) algorithm for resolving the multi-objective task scheduling at a large scale in the IaaS cloud environment.

3.5.11 | Whale optimization algorithm

Chen et al. [125] introduce the whale optimization algorithm (WOA) in IaaS clouds for task scheduling in terms of multi-objective optimization. Furthermore, the Improved WOA (IWOA) algorithm improves the optimal solution's search capability for task scheduling. IWOA algorithm aims to enhance the performance of IaaS clouds with given computing resources to reduce the cost.

3.5.12 | Wolf optimization algorithm

For the resource and task allocation, Natesan and Chokkalingam [126] propose the performance cost based on grey wolf optimization (PCGWO) algorithm to optimum resource provisioning for IaaS clouds. The objective functions concentrate on lowering the execution time and cost according to resource allocation and task scheduling. The simulation results illustrate that the proposed PCGWO algorithm completes better than the comparison algorithms in terms of performance metrics, including the makespan and cost while maximizing the number of tasks finished within a deadline.

3.6 | Traditional algorithms

The traditional algorithms work on the basis of the transition of operations from one place to another place at a fixed time and shifted to the next position. These algorithms start in a particular manner and place. Max-min [129], min-min algorithm [130], and longest cloudlet fastest processing [131] are used as traditional algorithms for resource scheduling in IaaS clouds that are exposed with the help of Table 7.

3.6.1 | Max-min algorithm

The max-min algorithm is proposed to realize the load balancing in elastic clouds. Moreover, it maintains the status of the task table to calculate the real-time load of VMs and the expected completion time. It allocates the workloads between nodes and recognizes the load balance. The experiment results show that the max-min algorithm improves resource utilization and decreases the response time of task scheduling tasks in IaaS clouds [132]. Further, Devipriya and Ramesh [133] propose an improved max-min algorithm for minimizing makespan for task scheduling. The enhanced max-min algorithm uses the expected execution time instead of the total time to improve IaaS clouds' scheduling. Similarly, Li et al. [134] propose the improved max-min algorithm to assess the real-time load of VMs and anticipated completion time tasks. It helps in the allocation of workload and realizing the workload by minimizing the response time.

3.6.2 | Min-min algorithm

Wang and Yu [135] propose an enhanced min-min algorithm for task scheduling to increase the proficiency for IaaS clouds. The min-min algorithm finalizes the complete and minimum execution time for tasks initially and is comprehensive and straightforward in the shortest period to optimize the influence of the data dependence.

3.6.3 | Longest cloudlet fastest processing algorithm

Task scheduling algorithms show a central role in scheduling tasks efficiently by decreasing the overall time and boosting resource utilization [131]. In the longest cloudlet fastest processing (LCFP) algorithm, the processing of the cloudlet complication is reflected when creating scheduling assessments.

3.6.4 | Computation time algorithm

Al-Maytami et al. [136] present the prediction of task computation time (PTCT) heuristic algorithm to increase task scheduling performance by applying the principal component analysis for IaaS clouds. The simulation results demonstrate that the PTCT algorithm achieves better than the max-min, min-max, min-min, and QoS-guided to reduce the execution time and makespan.

Most of the algorithms provide ideal solutions for the more critical problematic issues like optimal resource scheduling, VM distribution and placement, task and job scheduling, energy, and QoS-aware resource scheduling (frequently energy consumption for data centres). For solving these types of issues, generally, the best algorithms are used to achieve better performance. Still, the researcher does not consider all aspects of the cloud providers and users to get optimal solutions.

4 | PERFORMANCE METRICS

This section provides a description and comprehensive analysis of performing metrics used for resource scheduling in IaaS clouds.

4.1 | Availability

It determines whether the resource or task is accessible, ready, and suitable for usability or service in an appropriate format and specified location. It is the composite of resource serviceability, reliability, maintainability, and accessibility in IaaS clouds [137, 138].

$$\text{Availability} = \sum_{\text{resource}^i} \left(\frac{MTBM}{MTBM + MTTR} \right) \tag{1}$$

where MTBM denotes mean time between maintain while MTTR represents the mean time to repair. MTBM is the proportion of total uptime to the number of breakdowns and MTTR is the proportion of total downtime to the amount of breakdown of the i^{th} resource.

4.2 | Bandwidth

It is defined as the amount of data shifted or accomplished in a static time. It is frequently represented in bits or bytes per second (bps) [139].

$$\text{Bandwidth} = \sum_{\text{resource}^i} \left(\frac{\text{Size}}{\text{Capacity}} \right) \tag{2}$$

4.3 | Cost

It is the whole amount given to the cloud provider relative to resource utilization by the cloud user. The immediate resolution of the cloud providers is to make the most of the profit and revenue, whereas the cloud users want to minimize the expenditures. Therefore, the cost concentrates on all four factors: user expenses, the cloud provider's revenue, resource cost, and cloud provider's profit [140, 141].

$$\text{Cost} = \sum_{\text{resource}^i} (Ci \times Ti) \tag{3}$$

where Ci denotes the cost of i^{th} resource per unit time and Ti indicates the time of consumption of i^{th} resource.

4.4 | Energy consumption

It is the use of power or energy, which is the discrete utilization of energy by resources to perform cloudlets or tasks. The energy utilization of specified i^{th} resource at a time T with placement F is provided by [142, 143].

$$Energy = \sum_{resource^i} \int_{StrTime}^{FnhTime} E_i\,(F, T) \tag{4}$$

where E_i signifies the energy is consumed by i^{th} resource from the starting time (Str_{Time}) to finishing time (Fnh_{Time}) of consumption.

4.5 | Execution cost

The cost is the variation between the suitable price with service level agreement (SLA) and the actual cost of j^{th} cloudlet or task for completion [144, 145].

4.6 | Execution time

It is also known as finishing time, described as the time of a specific task or cloudlet consumed while implementing that cloudlet or task [146].

$$\text{Exection Time} = \text{task}_j(Fnh_{Time} - Str_{Time}) \tag{5}$$

where Fnh_{Time} represents the finishing time and Str_{Time} signifies the starting time of j^{th} cloudlet or task.

4.7 | Failure rate

The failure rate is the frequency in which a cloudlet or task fails and cannot accomplish the execution, expressed in failures per unit of time. It is highly used in the reliability of IaaS clouds [147, 148].

$$\text{Failure Rate} = \frac{\sum_{task^j} (\text{Failure})}{Total_{Time}} \tag{6}$$

4.8 | Makespan

It identifies the upper limit of the completion time of a task. In other words, it demonstrates the time at which the resources should be assigned to the users. It is to be noted that if the makespan of a particular cloudlet or task is not diminished, then the cloud users' demands will not be accomplished in the required given period of time [118].

$$\text{Makespan} = max_{task^j} (Fnh_{Time}) \tag{7}$$

where Fnh_{Time} denotes the finishing time of j^{th} cloudlet or task.

4.9 | Migration time

The migration process is applied to transfer or migrate a VM from one server to another, which is the migration time. For the better performance of the system, it should be a minimum [149].

$$\text{Migration time} = \frac{M_x}{B_y} \tag{8}$$

where M_x denotes the memory used and B_y represents the bandwidth available for k^{th} VM.

4.10 | Priority

It is the importance or the right of a cloudlet or task to execute or proceed before other cloudlets and tasks based on priority. In IaaS clouds, it is essential since the cloud user pays additional charges for its urgent requirements or benefits for the cloud providers [150].

$$\text{Priority} = \sum_{\text{task}^j} \left(Exe_{Time} + \text{Capicity} \times \text{Number of requests}\right) \tag{9}$$

4.11 | Reliability

Reliability is the calculation of the capability of a cloudlet or task to compute its required functions successfully in a stipulated time. It offers the assurance of completing a specific cloudlet or task while avoiding or reducing the failure rate in IaaS clouds [147, 151].

$$\text{Reliability} = \frac{\sum_{task^j} (Exe_{Time})}{Total_{Time}} \tag{10}$$

4.12 | Response time

The response time refers to the sum of the waiting and submission time. Precisely, response time is the time of task or cloudlet that it takes to respond to a specified input [146].

$$\text{Response time} = task_j(Sb_{Time} + Str_{Time}) \tag{11}$$

where Sb_{Time} represents the submission time and Str_{Time} signifies the starting time of the j^{th} cloudlet or task.

4.13 | Resource utilization

It is demonstrated by the utilization of a resource in a way to maximise the throughput. Resource utilization can be maximized by decreasing the resource volume used in terms of enhancing the profit and minimizing energy consumption [140, 152].

$$\text{Resource utilization} = \frac{\sum_{resource^i} (Exe_{Time})}{\text{makespan}} \tag{12}$$

where Exe_{Time} indicates the execution time of j^{th} task or cloudlet that uses.

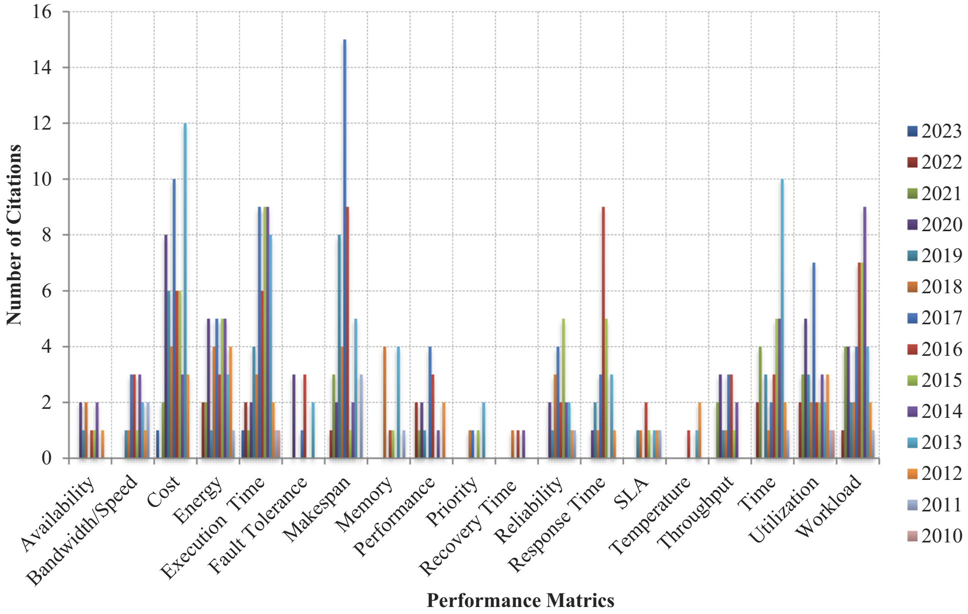

FIGURE 3 Analysis of resource scheduling parameters from 2010 to 2023.

4.14 | Throughput

Generally, the rate of production or processing, such as production or processing in a fixed time, is referred to as throughput. In IaaS clouds, throughput identifies the cloudlets or tasks that are computed in a specific period of time [153, 141].

$$\text{Throughput} = \sum_{\text{task}^j} (Exe_{Time}) \tag{14}$$

4.15 | Waiting time

It is a difference between the submission time and the starting time of the cloudlet or task [154, 155].

$$\text{Waiting time} = task_j(Str_{Time} - Sb_{Time}) \tag{15}$$

where Str_{Time} represents the starting time and Sb_{Time} signifies the submission time of the j^{th} cloudlet or task.

4.16 | Workload

It is the volume of handled processing within a specific time. In a simple way, it is the capability to regulate or process work in IaaS clouds. The term degree of imbalance (DI) is applied to calculate the workload in data centres [156, 157].

$$\text{Degree of imbalance} = \frac{max_{task^j}\ (Exe_{Time}) - min_{task^j}\ (Exe_{Time})}{Avg_{task^j}\ (Exe_{Time})} \tag{16}$$

The performance metrics also play an essential role in achieving optimal resource scheduling in IaaS Clouds. Figure 3 shows the analysis of resource scheduling parameters from the years 2010 to 2023. These are some criteria through which cloud providers and cloud user can acquire their satisfaction level with the help of these parameters in performance. Cost 12.71% execution time 12.08%, makespan 11.04%, workload (degree of imbalance) 9.79%, energy consumption 8.33%, utilization of resources 7.50%, response time 5.21%, reliability 4.79%, bandwidth/speed 3.54%, throughput 3.33%, and (others availability, fault tolerance, SLA and recovery time need to be more focus in further research) 5.25% used by researchers for resource scheduling for IaaS clouds in their research.

These algorithms are executed in various simulation tools and real test-bed environments including CloudSim 54.88% [158, 159], GridSim 0% [160], SimGrid 1.22% [161], MATLAB 7.32% and real cloud environment (practically implementation 12.20%), and provide the near-optimum solution for the resource scheduling in IaaS cloud computing. Hence, 18.29% researchers have not mentioned their simulation tools and real cloud environment for the evaluation.

CloudSim supports only the standard workload format (SWF) for the dataset. Parallel workload archives are designed for the workloads on parallel machines. PWA contains two types of workloads. Raw workload logs from several devices all over the world and workload models. The objective is to create this information free of cost and accessible to researchers concerned with the appraisal of parallel systems and precise schedulers for such systems. PWA is also implemented in the research of IaaS Clouds High-Performance Computing Centre North [162]. The aim of the Grid Workload Archive is to make available a virtual meeting place where researchers can use and update the grid workload traces for the research. The Grid Workload Archive has a standard Grid Workloads Format (GWF). Based on GWF, high accessibility, easily available database in the effect of grid workload traces can be modified and created for the IaaS clouds

NASA Ames iPCS/860 [163]. So simulation or real implementation has a different significant role in the performance of the algorithms for resource scheduling in IaaS large-scale clouds.

For the comparison of results, most researchers have compared the algorithms (which are mostly considered meta-heuristics) with the traditional or basic algorithms. In this case, how can the optimal solutions be obtained, so researchers should also focus on the comparison criteria in resource scheduling for IaaS clouds. Quality of the solution is also necessary for the enhancement of resource scheduling for IaaS clouds and mostly statistical analysis is used to check the quality of the solutions. Furthermore, convergence rate (two approaches are used to either reduce the number of iterations or enhance the performance metrics), failure rate (in terms of reliability), multi-objective optimization (most considered two objectives and need to be worked for more than two objectives) and load balancing (tasks and VM load distribution) are the most critical issues are found in resource scheduling for IaaS clouds. All these critical issues and challenges need to be focused on for further research in IaaS clouds and other domains like fog computing and IoT [164–170].

5 | OPEN ISSUES AND CHALLENGES

Through reviewing the literature and analysing the IaaS cloud resource scheduling problems and existing solutions, we identified the limitations of the existing work and the remaining challenges, as illustrated in Figure 4. The problem leading to the research, as illustrated in Figure 4, highlights the main requirements that should be considered during the design of resource scheduling optimization algorithms or algorithms for IaaS clouds. These requirements include overcoming the local search and global search convergence, ensuring minimum execution cost and time as the user's QoS expectations, and handling dynamic fluctuations of cloud resources and demands while maintaining reliability for better performance.

Based on the various resource scheduling algorithms, several literature reviews and surveys have been published. We observed that there is still a need for additional enhancements and improvements in resource scheduling for IaaS clouds. The critical issues of resource scheduling in IaaS clouds depend upon cost, execution time, fault tolerance, makespan, response time, throughput, and overall performance based on the results and outcomes of algorithms. The energy consumption efficiency without degrading performance and avoiding SLA violence is also a critical issue that requires special attention in green computing. Hence, resource scheduling is considered one of the crucial problems in IaaS cloud resource management. There are also some open issues listed that highlight several potential directions.

- Cloud-IoT scheduling: Newly, IoT technology has brought the revolution into a more ingenious world with a huge amount of data and various services. Cloud computing is a widely sustainable technology, particularly in IoT environments, as it offers an adaptable mass of computing, storage, and software services at a much-decreased cost. Nevertheless, the junction of IoT and cloud computing can bring about innovative prospects in the respective technologies with efficient scheduling.
- Dynamic scheduling: Cloud users have variable demands with different QoS requirements. Efficient scheduling methods or techniques must be able to dynamically modify the scheduling process to meet the variable demands of users and adapt to changes in the IaaS cloud environment.
- Energy-aware scheduling: Low power consumption is also an essential issue for the service providers. For this purpose, methods or algorithms must be designed to minimize energy consumption and maximize income trade-offs for IaaS clouds.
- Green scheduling: It is necessary to use the computing resources in a user-friendly and environmentally clean way. For this purpose, scheduling algorithms need to use less energy consumption, hazardous materials, heat generation, and resource wastage during the resource scheduling for IaaS clouds.
- Large-scale scheduling: With the quick advancement of cloud computing, the number of resources, users, tasks, providers, data centres, and workflows is continuously growing. A significantly larger number of cloudlets/tasks will be processed in the future, and a substantial amount of resources need to be scheduled and organized in IaaS clouds. Managing the large-scale cloud resources and cloud users' demands will be challenging for the current scheduling algorithms.
- Load balancing aware scheduling: Workflows amongst the various computing nodes are also critical for IaaS cloud computing. Overloading and underloading of resources must be avoided by the optimal distribution of loads amongst the distributed nodes. The multi-kernel modules should perform monitoring of load distribution on physical as well as virtual nodes to balance the workload and performance of heavily loaded clouds.
- Multi-objective scheduling: A resource scheduling problem is a multi-objective problem, but mostly, these are handled with single-objective methods or algorithms. Every cloud user wants the minimum cost with high performance, while every cloud provider wants the high profit or revenue with maximum utilization of resources. Therefore, a multi-objective resource scheduling problem might be more significant in the upcoming IaaS cloud research.
- Real-time scheduling: In IaaS clouds, the cloud resources and the user demands fluctuate dynamically. Therefore, the scheduling methods or algorithms need to be smart enough to make real-time responses to the changing environment.
- Security and privacy-aware scheduling: Security and privacy-aware scheduling is another area of research in IaaS clouds that needs to be explored using heuristic and meta-heuristic algorithms. Investigations must be performed to protect sensitive or private information related to the tasks and cloud users [171–173]. However, these algorithms still suffer from some weaknesses and shortcomings that significantly affect the success of privacy. In certain situations, the privacy preservation of published datasets has faced limitations.

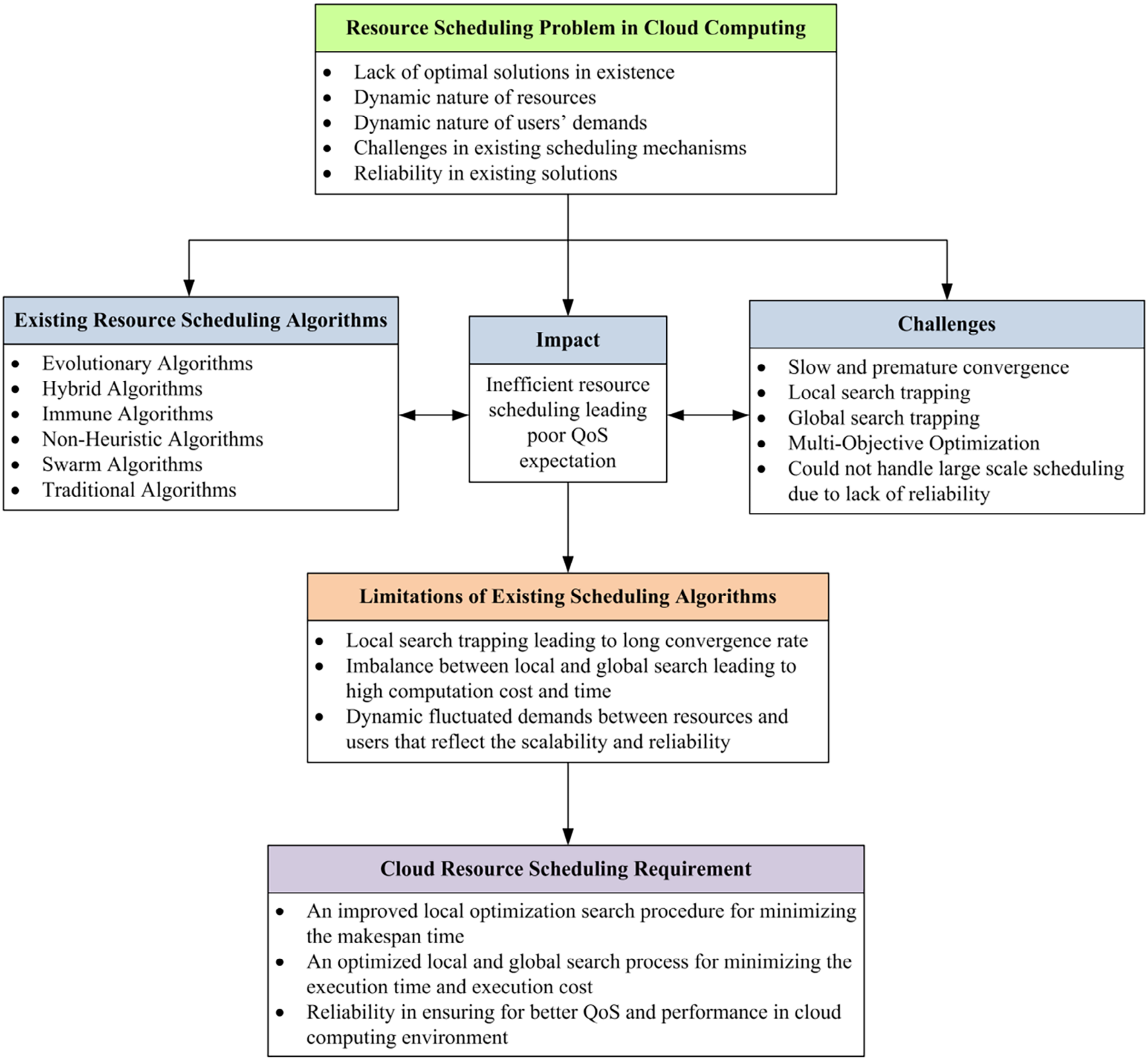

FIGURE 4 Cloud resource scheduling problems, existing solutions and their limitations, challenges, and design requirements.

Breaching positive information is the main drawback, leading to the failure of privacy preservation for the published dataset, especially if the privacy process relies on it.

6 | CONCLUSION

Cloud computing is becoming more attractive for many organizations because it provides multiple computing services like cloud storage, hosting, cloud servers etc. Cloud computing's scope is the largest and shrinks the development efforts and overcomes the maintenance, installation, and hurdles. This critical review presents an in-depth analysis of resource scheduling for IaaS clouds. It is supposed that the numerous resource scheduling issues are yet to be tackled concerning solved algorithms and algorithms. This is the era of various security attacks, so detection and avoidance factors of serious viruses must be included in the designed schemes. In most of the analysed procedures, security factors are neglected or lightly covered. Therefore, future research on security factors will resolve cloud service providers' security and resource availability problems. Numerous weaknesses include user input constraints, deadlines, energy efficiency, execution cost, performance issues, makespan, and transmission cost. It is also recommended that designed algorithms should be less complicated as they consume less energy for processing. Hence, there is a requisite to execute scheduling algorithms considering the performance metrics and comparison criteria to a more near-optimal solution for IaaS clouds. With the emergence of fog/edge computing technology, distributed processing across cloud and edge will play a significant role in future networks. Consequently, distributed and collaborative resource scheduling among edge and cloud resources is essential to exploit the benefits of load distribution and faster processing at the network edge. Although processing at the network edge reduces delays and mitigates the loads on cloud computing centres, scheduling algorithms should take into consideration the limited resources of fog/edge computing in IoT environments.

AUTHOR CONTRIBUTIONS

Syed Hamid Hussain Madni: Conceptualization; data curation; formal analysis; methodology; project administration; resources; writing—original. **Muhammad Faheem**: Conceptualization; data curation; formal analysis; methodology; software; writing—review and editing. **Muhammad Younas**: Investigation; resources; software; writing—review and editing. **Maidul Hasan Masum**: Formal analysis; investigation; validation; visualization; writing—original draft; writing—review and editing. **Sajid Shah**: Methodology; software; validation; visualization; writing—original draft; writing—review and editing.

ACKNOWLEDGEMENTS

The research work of Muhammad Faheem is supported in part by the Academy of Finland and the University of Vaasa, Finland.

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

DATA AVAILABILITY STATEMENT

Data will be available upon request.

ORCID

Syed Hamid Hussain Madni https://orcid.org/0000-0002-3816-1382
Muhammad Faheem https://orcid.org/0000-0003-4628-4486
Sajid Shah https://orcid.org/0009-0006-9980-101X

REFERENCES

1. Foster, I., Zhao, Y., Raicu, I., Lu, S.: Cloud computing and grid computing 360-degree compared. In: Grid Computing Environments Workshop, GCE'08, 2008, pp. 1–10. IEEE, Piscataway, NJ (2008).
2. Vaquero, L.M., Rodero-Merino, L., Caceres, J., Lindner, M.: A break in the clouds: towards a cloud definition. ACM SIGCOMM Comput. Commun. Rev. 39(1), 50–55 (2008)
3. Puthal, D., Sahoo, B., Mishra, S.,Swain, S.: Cloud computing features, issues, and challenges: a big picture. In: International Conference on Computational Intelligence and Networks (CINE), pp. 116–123. IEEE, Piscataway, NJ (2015)
4. Faheem, M., Raza, B., Bhutta, M.S., Madni, S.H.H.: A blockchain-based resilient and secure framework for events monitoring and control indistributed renewable energy systems. IET Blockchain. 4, 1–15 (2024)
5. John, A., Isnin, I.F.B., Madni, S.H.H., Faheem, M.: Cluster-based wireless sensor network framework for denial-of-service attack detection based on variable selection ensemble machine learning algorithms. Intell. Syst. Appl. 22, 200381 (2024)
6. Ma, T., Chu, Y., Zhao, L., Ankhbayar, O.: Resource allocation and scheduling in cloud computing: Policy and algorithm. IETE Tech. Rev. 31(1), 4–16 (2014)
7. Gul, B. et al.: CPU and RAM energy-based SLA-aware workload consolidation techniques for clouds. IEEE Access 8, 62990–63003 (2020)
8. Zhang, Y., Su, Y.M.: Research on resource scheduling algorithm in cloud computing data center. Adv. Mater. Res. 926, 2050–2053 (2014)
9. Liaqat, M., Naveed, A., Ali, R.L., Shuja, J., Ko, K.-M.: Characterizing dynamic load balancing in cloud environments using virtual machine deployment models. IEEE Access 7, 145767–145776 (2019)
10. Manvi, S.S., Shyam, G.K.: Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey. J. Netw. Comput. Appl. 41, 424–440 (2014)
11. Huang, L., Chen, H.-S., Hu, T.-T.: Survey on resource allocation policy and job scheduling algorithms of cloud computing. J. Softw. 8(2), 480–487 (2013)
12. Gill, P.E., Murray, W., Wright, M.H.: Practical optimization. SIAM. 8(3), 1–28 (2019)
13. Yang, X.-S.: Optimization Techniques and Applications with Examples. John Wiley & Sons, Hoboken, NJ (2018)
14. Wang, L. et al.: Cloud computing: a perspective study. New Gener. Comput. 28(2), 137–146 (2010)
15. Madni, S.H.H., Latiff, M.S.A., Coulibaly, Y.: Resource scheduling for infrastructure as a service (IaaS) in cloud computing: Challenges and opportunities. J. Netw. Comput. Appl. 68, 173–200 (2016)
16. Singh, S., Chana, I.: A survey on resource scheduling in cloud computing: Issues and challenges. J. Grid Comput. 14(2), 217–264 (2016)
17. Cao, Y., Ro, C., Yin, J.: Comparison of job scheduling policies in cloud computing. In: Future Information Communication Technology and Applications, pp. 81–87. Springer, Berlin (2013)
18. Abdulhamid, S.M., Latiff, M.S.A., Bashir, M.B.: Scheduling techniques in on-demand grid as a service cloud: A review. J. Theor. Appl. Inf. Technol. 63(1), 1–32 (2014)
19. Huang, M.G., Ou, Z.Q.: Review of task scheduling algorithm research in cloud computing. Adv. Mater. Res. 926, 3236–3239 (2014)
20. Mathew, T., Sekaran, K.C., Jose, J.: Study and analysis of various task scheduling algorithms in the cloud computing environment. In: International Conference on Advances in Computing, Communications and Informatics (ICACCI, 2014), pp. 658–664. IEEE, Piscataway, NJ (2014)
21. Tsai, C.-W., Rodrigues, J.J.: Metaheuristic scheduling for cloud: A survey. IEEE Syst. J. 8(1), 279–291 (2014)
22. Kalra, M., Singh, S.: A review of metaheuristic scheduling techniques in cloud computing. Egypt. Inf. J. 16, 275–295 (2015)
23. Zhan, Z.-H., Liu, X.-F., Gong, Y.-J., Zhang, J., Chung, H.S.-H., Li, Y.: Cloud computing resource scheduling and a survey of its evolutionary approaches. ACM Comput. Surv. 47(4), 1–26 (2015)
24. Masdari, M., Salehi, F., Jalali, M., Bidaki, M.: A survey of PSO-based scheduling algorithms in cloud computing. J. Netw. Syst. Manag. 25(1), 122–158 (2017)
25. Liu, L., Qiu, Z.: A survey on virtual machine scheduling in cloud computing. In: 2nd IEEE International Conference on Computer and Communications (ICCC), pp. 2717–2721. IEEE, Piscataway, NJ (2016)
26. Babu, A.A., Rajam, V.M.A.: Resource scheduling algorithms in cloud environment-A survey. In: 2017 Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM), pp. 25–30. IEEE, Piscataway, NJ (2017)
27. Singh, P., Dutta, M., Aggarwal, N.: A review of task scheduling based on meta-heuristics approach in cloud computing. Knowl. Inf. Syst. 52(1), 1–51 (2017)
28. Kumar, M., Sharma, S.C., Goel, A., Singh, S.P.: A comprehensive survey for scheduling techniques in cloud computing. J. Netw. Comput. Appl. 143, 1–33 (2019)
29. Arunarani, A., Manjula, D., Sugumaran, V.: Task scheduling techniques in cloud computing: A literature survey. Future Gener. Comput. Syst. 91, 407–415 (2019)
30. Mahmoud, H., Thabet, M., Khafagy, M.H., Omara, F.A.: A comparative study of heterogenous task-based scheduling techniques in a cloud environment. In: 2020 International Conference on Innovative Trends

in Communication and Computer Engineering (ITCE), pp. 1–6. IEEE, Piscataway, NJ (2020)
31. George, S.S., Pramila, R.S.: A review of different techniques in cloud computing. Mater. Today: Proc. 46, 8002–8008 (2021)
32. Houssein, E.H., Gad, A.G., Wazery, Y.M., Suganthan, P.N.: Task Scheduling in cloud computing based on meta-heuristics: review, taxonomy, open challenges, and future trends. Swarm Evol. Comput. 62, 100841 (2021)
33. Singh, H., Tyagi, S., Kumar, P., Gill, S.S., Buyya, R.: Metaheuristics for scheduling of heterogeneous tasks in cloud computing environments: Analysis, performance evaluation, and future directions. Simul. Modell. Pract. Theory 111, 102353 (2021)
34. Rahimikhanghah, A., Tajkey, M., Rezazadeh, B., Rahmani, A.M.: Resource scheduling methods in cloud and fog computing environments: a systematic literature review. Cluster Comput. 25, 1–35 (2022)
35. Chaurasia, N., Kumar, M., Chaudhry, R., Verma, O.P.: Comprehensive survey on energy-aware server consolidation techniques in cloud computing. J. Supercomput. 77, 11682–11737 (2021)
36. Brownlee, J., Clever algorithms: nature-inspired programming recipes. Jason Brownlee, (2011)
37. Xing, B., Gao, W.-J.: Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms. Springer, Heidelberg (2014)
38. Storn, R., Price, K.: Differential Evolution-a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. International Computer Science Institute, Berkeley, CA (1995)
39. Goldberg, D.E., Holland, J.H.: Genetic algorithms and machine learning. Mach. Learn. 3(2), 95–99 (1988)
40. Zain, A.M., Yousif, A.: Chemical reaction optimization (CRO) for cloud job scheduling. SN Appl. Sci. 2(1), 1–12 (2020)
41. Cui, Y.F., Li, X.M., Dong, K.W., Zhu, J.L.: Cloud computing resource scheduling method research based on improved genetic algorithm. Adv. Mater. Res. 271-273, 552–557 (2011)
42. Chen, S., Wu, J., Lu, Z.: A cloud computing resource scheduling policy based on genetic algorithm with multiple fitness. In: 2012 IEEE 12th International Conference on Computer and Information Technology (CIT), pp. 177–184. IEEE, Piscataway, NJ (2012)
43. Zhao, C., Zhang, S., Liu, Q., Xie, J., Hu, J.: Independent tasks scheduling based on genetic algorithm in cloud computing. In: WiCom'09. 5th International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1–4. IEEE, Piscataway, NJ (2009)
44. Wu, G., Tang, M., Tian, Y.-C., Li, W.: Energy-efficient virtual machine placement in data centers by genetic algorithm. In: Neural Information Processing, pp. 315–323. Springer, Heidelberg (2012)
45. Sindhu, S., Mukherjee, S.: A genetic algorithm based scheduler for cloud environment. In: 2013 4th International Conference on Computer and Communication Technology (ICCCT), 6, pp. 23–27. IEEE, Piscataway, NJ (2013)
46. Ma, J., Li, W., Fu, T., Yan, L., Hu, G.: A novel dynamic task scheduling algorithm based on improved genetic algorithm in cloud computing. In: Wireless Communications, Networking and Applications. pp. 829–835. Springer, Heidelberg (2016)
47. Agarwal, M., Srivastava, G.M.S.: A genetic algorithm inspired task scheduling in cloud computing. In: 2016 International Conference on Computing, Communication and Automation (ICCCA), pp. 364–367. IEEE, Piscataway, NJ (2016)
48. Gawanmeh, A., Parvin, S., Alwadi, A.: A genetic algorithmic method for scheduling optimization in cloud computing services. Arab. J. Sci. Eng. 43, 1–10 (2017)
49. Mahmood, A., Khan, S.A.: Hard real-time task scheduling in cloud computing using an adaptive genetic algorithm. Computers 6(2), 15 (2017)
50. Zhou, Z., Li, F., Zhu, H., Xie, H., Abawajy, J.H., Chowdhury, M.U.: An improved genetic algorithm using greedy strategy toward task scheduling optimization in cloud environments. Neural. Comput. Appl. 32(6), 1531–1541 (2020)
51. Devi, K.L., Valli, S.: Multi-objective heuristics algorithm for dynamic resource scheduling in the cloud computing environment. J. Supercomput. 77, 8252–8280 (2021)
52. Dorigo, M., Blum, C.: Ant colony optimization theory: A survey. Theor. Comput. Sci. 344(2), 243–278 (2005)
53. Yang, X.-S. and Deb, S.: Cuckoo search via Lévy flights. In: World Congress on Nature & Biologically Inspired Computing, 2009. NaBIC 2009, pp. 210–214. IEEE, Piscataway, NJ (2009)
54. Kennedy, J.: Particle swarm optimization. In: Encyclopedia of Machine Learning. pp. 760–766. Springer, Heidelberg (2010)
55. Wen, X., Huang, M., Shi, J.: Study on resources scheduling based on ACO allgorithm and PSO algorithm in cloud computing. In: 2012 11th International Symposium on Distributed Computing and Applications to Business, Engineering & Science (DCABES), pp. 219–222. IEEE, Piscataway, NJ (2012)
56. Yang, H.: Improved ant colony algorithm based on PSO and its application on cloud computing resource scheduling. Adv. Mater. Res. 989, 2192–2195 (2014)
57. Cho, K.-M., Tsai, P.-W., Tsai, C.-W., Yang, C.-S.: A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing. Neural. Comput. Appl. 26, 1–13 (2014)
58. Shojafar, M., Javanmardi, S., Abolfazli, S., Cordeschi, N.: FUGE: A joint meta-heuristic approach to cloud job scheduling algorithm using fuzzy theory and a genetic method. Cluster Comput. 18,1–16 (2015)
59. Javanmardi, S., Shojafar, M., Amendola, D., Cordeschi, N., Liu, H., Abraham, A.: Hybrid job scheduling algorithm for cloud computing environment. In: Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA, pp. 43–52. Springer, Berlin (2014)
60. Tsai, J.-T., Fang, J.-C., Chou, J.-H.: Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm. Comput. Oper. Res. 40(12), 3045–3055 (2013)
61. Liu, C.-Y., Zou, C.-M., Wu, P.: A task scheduling algorithm based on genetic algorithm and ant colony optimization in cloud computing. In: 2014 13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES), pp. 68–72. IEEE, Piscataway, NJ (2014)
62. Raju, I.R.K., Varma, P.S., Sundari, M.R., Moses, G.J.: Deadline aware two stage scheduling algorithm in cloud computing. J. Sci. Technol. 9(4), 1–10 (2016)
63. Jacob, T.P., Pradeep, K.: A multi-objective optimal task scheduling in cloud environment using cuckoo particle swarm optimization. Wirel. Pers. Commun. 109(1), 315–331 (2019)
64. Pradeep, K., Jacob, T.P.: A hybrid approach for task scheduling using the cuckoo and harmony search in cloud computing environment. Wireless Pers. Commun. 101(4), 2287–2311 (2018)
65. Kumar, N.J., Balasubramanian, C.: Hybrid gradient descent golden eagle optimization (HGDGEO) algorithm-based efficient heterogeneous resource scheduling for big data processing on clouds. Wireless Pers. Commun. 129, 1–21 (2023)
66. Gajera, V., Gupta, R., Jana, P.K.: An effective multi-objective task scheduling algorithm using min-max normalization in cloud computing. In: 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), pp. 812–816. IEEE, Piscataway, NJ (2016)
67. Mansouri, N., Zade, B.M.H., Javidi, M.M.: Hybrid task scheduling strategy for cloud computing by modified particle swarm optimization and fuzzy theory. Comput. Ind. Eng. 130, 597–633 (2019)
68. Xue, S., Shi, W., Xu, X.: A heuristic scheduling algorithm based on PSO in the cloud computing environment. Int. J. u- e- Serv., Sci. Technol. 9(1), 349–362 (2016)
69. Madni, S.H.H., Abd Latiff, M.S., Abdullah, A.H.: Reliability aware resource scheduling based on fuzzy cuckoo search (FCS) technique for IaaS cloud. IOP Conf. Ser.: Mater. Sci. Eng. 884(1), 012053 (2020)
70. Khan, M.S.A., Santhosh, R.: Task scheduling in cloud computing using hybrid optimization algorithm. Soft Comput. 26(23), 13069–13079 (2022)
71. Mori, K., Tsukiyama, M., Fukuda, T.: Immune algorithm with searching diversity and its application to resource allocation problem. Trans. Jpn. Soc. Mech. Eng. C 113, 872–872 (1993)

72. Jiao, H., Zhang, J., Li, J., Shi, J., Li, J.: Immune optimization of task scheduling on multidimensional QoS constraints. Cluster Comput. 18(2), 909–918 (2015)
73. Lin, W., Liang, C., Wang, J.Z., Buyya, R.: Bandwidth-aware divisible task scheduling for cloud computing. Softw.: Pract. Exper. 44(2), 163–174 (2014)
74. Tang, J.-M., Luo, L., Wei, K.-M., Guo, X., Ji, X.-Y.: A heuristic resource scheduling algorithm of cloud computing based on polygons correlation calculation. In: IEEE 12th International Conference on e-Business Engineering (ICEBE), pp. 365–370. IEEE, Piscataway, NJ (2015)
75. Kapur, R.: A cost effective approach for resource scheduling in cloud computing. In: 2015 International Conference on Computer, Communication and Control (IC4), pp. 1–6. IEEE, Piscataway, NJ (2015)
76. Panda, S.K., Jana, P.K.: An energy-efficient task scheduling algorithm for heterogeneous cloud computing systems. Cluster Comput. 22(2), 509–527 (2019)
77. Dong, Z., Liu, N., Rojas-Cessa, R.: Greedy scheduling of tasks with time constraints for energy-efficient cloud-computing data centers. J. Cloud Comput. 4(1), 1–14 (2015)
78. Li, F.F., Xie, D.Q., Gao, Y., Xie, G.W., Guo, Q.Y.: Research on multi-QoS and trusted task scheduling in cloud computing environment. Appl. Mech. Mater. 263, 1892–1896 (2013)
79. Kumar, N., Chilamkurti, N., Zeadally, S., Jeong, Y.-S.: Achieving quality of service (QoS) using resource allocation and adaptive scheduling in cloud computing with grid support. Comput. J. 57(2), 281–290 (2014)
80. Wu, X., Deng, M., Zhang, R., Zeng, B., Zhou, S.: A task scheduling algorithm based on QoS-driven in Cloud Computing. Procedia Comput. Sci. 17, 1162–1169 (2013)
81. Bansal, N., Maurya, A., Kumar, T., Singh, M., Bansal, S.: Cost performance of QoS Driven task scheduling in cloud computing. Procedia Comput. Sci. 57, 126–130 (2015)
82. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Global Optim. 39(3), 459–471 (2007)
83. Yang, X.-S.: A new metaheuristic bat-inspired algorithm. In: Nature inspired cooperative strategies for optimization (NICSO 2010), pp. 65–74. Springer, Berlin (2010)
84. Yang, X.-S.: Firefly algorithms for multimodal optimization. In: Stochastic Algorithms: Foundations and Applications. pp. 169–178. Springer, Heidelberg (2009)
85. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department. 200(6), 1–10 (2005)
86. Kashan, A.H.: League championship algorithm: a new algorithm for numerical function optimization. In: SOCPAR'09. International Conference of Soft Computing and Pattern Recognition, pp. 43–48. IEEE, Piscataway, NJ (2009)
87. Kenndy, J., Eberhart, R.: Particle swarm optimization. Proc. IEEE Int. Conf. Neural Networks 4, 1942–1948 (1995)
88. Cheng, M.-Y., Prayogo, D.: Symbiotic organisms search: A new metaheuristic optimization algorithm. Comput. Struct. 139, 98–112 (2014)
89. Tawfeek, M., El-Sisi, A., Keshk, A.E., Torkey, F.: Cloud task scheduling based on ant colony optimization. In: 2013 8th International Conference on Computer Engineering and Systems (ICCES), pp. 64–69. IEEE, Piscataway, NJ (2013)
90. Moon, Y., Yu, H., Gil, J.-M., Lim, J.: A slave ants based ant colony optimization algorithm for task scheduling in cloud computing environments. Hum.-centric Comput. Inf. Sci. 7(1), 28 (2017)
91. Xue, S., Li, M., Xu, X., Chen, J., Xue, S.: An ACO-LB algorithm for task scheduling in the cloud environment. J. Softw. 9(2), 466–473 (2014)
92. Li, K., Xu, G., Zhao, G., Dong, Y., Wang, D.: Cloud task scheduling based on load balancing ant colony optimization. In: 2011 Sixth Annual Chinagrid Conference (ChinaGrid), pp. 3–9. IEEE, Piscataway, NJ (2011)
93. Liu, X.-F., Zhan, Z.-H., Du, K.-J., Chen, W.-N.: Energy aware virtual machine placement scheduling in cloud computing based on ant colony optimization approach. In: Proceedings of the 2014 Conference on Genetic and Evolutionary Computation, pp. 41–48. Association for Computing Machinery, New York (2014)
94. Zuo, L., Shu, L., Dong, S., Zhu, C., Hara, T.: A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing. IEEE Access 3, 2687–2699 (2015)
95. Zhu, Y., Liang, H.: Research for the virtual machine-oriented cloud resource scheduling algorithm. In: 2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering (ICIII), pp. 133–136. IEEE, Piscataway, NJ (2013)
96. Pan, J.-S., Wang, H., Zhao, H., Tang, L.: Interaction artificial bee colony based load balance method in cloud computing. In: Genetic and Evolutionary Computing. pp. 49–57. Springer, Berlin (2015)
97. Raghavan, S., Marimuthu, C., Sarwesh, P., Chandrasekaran, K.: Bat algorithm for scheduling workflow applications in cloud. In: 2015 International Conference on Electronic Design, Computer Networks & Automated Verification (EDCAV), pp. 139–144. IEEE, Piscataway, NJ (2015)
98. Madni, S.H.H., Latiff, M.S.A.: Optimal resource scheduling for IaaS cloud computing using cuckoo search algorithm. Sains Humanika 9(1–3), (2017)
99. Agarwal, M., Srivastava, G.M.S.: A cuckoo search algorithm-based task scheduling in cloud computing. In: Advances in Computer and Computational Sciences. pp. 293–299. Springer, Heidelberg (2018)
100. Aloboud, E., Kurdi, H.: Cuckoo-inspired job scheduling algorithm for cloud computing. Procedia Comput. Sci. 151, 1078–1083 (2019)
101. Madni, S.H.H., Abd Latiff, M.S., Abdulhamid, S.M., Ali, J.: Hybrid gradient descent cuckoo search (HGDCS) algorithm for resource scheduling in IaaS cloud computing environment. Cluster Comput. 22, 301–334 (2019)
102. Zhao, L.F., Zhou, S.H., Chang, W.B.: Task scheduling in cloud computing with improved firefly algorithm. Appl. Mech. Mater. 602, 3189–3193 (2014)
103. Gupta, P., Ghrera, S.P.: Load and fault aware honey bee scheduling algorithm for cloud infrastructure. In: Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA), pp. 135–143. Springer, Berlin (2014)
104. Babu, K.R., Samuel, P.: Enhanced bee colony algorithm for efficient load balancing and scheduling in cloud. In: Innovations in Bio-Inspired Computing and Applications. pp. 67–78. Springer, Heidelberg (2016)
105. Liu, J., Luo, X.G., Zhang, X.M., Zhang, F.: Job scheduling algorithm for cloud computing based on particle swarm optimization. Adv. Mater. Res. 662, 957–960 (2013)
106. Kumar, D., Raza, Z.: A PSO based VM resource scheduling model for cloud computing. In: IEEE International Conference on Computational Intelligence and Communication Technology (CICT), pp. 213–219. IEEE, Piscataway, NJ (2015)
107. Zhong, Z., Chen, K., Zhai, X., Zhou, S.: Virtual machine-based task scheduling algorithm in a cloud computing environment. Tsinghua Sci. Technol. 21(6), 660–667 (2016)
108. Abdi, S., Motamedi, S.A., Sharifian, S.: Task scheduling using modified PSO algorithm in cloud computing environment. In: International Conference on Machine Learning, Electrical and Mechanical Engineering. Springer, 4(1), p. 8 (2014)
109. Xiong, Y.Y., Wu, Y.Y.: Cloud computing resource schedule strategy based on PSO algorithm. Appl. Mech. Mater. 513-517, 1332–1336 (2014)
110. Al-Olimat, H.S., Alam, M., Green, R., Lee, J.K.: Cloudlet scheduling with particle swarm optimization. In: 2015 Fifth International Conference on Communication Systems and Network Technologies (CSNT). pp. 991–995. IEEE, Piscataway, NJ (2015)
111. Zhang, H., Li, P., Zhou, Z., Yu, X.: A PSO-based hierarchical resource scheduling strategy on cloud computing. Trustworthy Computing and Services. pp. 325–332. Springer, Heidelberg (2013)
112. Ramezani, F., Lu, J., Hussain, F.: Task scheduling optimization in cloud computing applying multi-objective particle swarm optimization. In: Service-Oriented Computing. pp. 237–251. Springer, Heidelberg (2013)
113. Jena, R.: Multi objective task scheduling in cloud environment using nested PSO framework. Procedia Comput. Sci. 57, 1219–1227 (2015)
114. Li, H.-H., Fu, Y.-W., Zhan, Z.-H., Li, J.-J.: Renumber strategy enhanced particle swarm optimization for cloud computing resource scheduling. In:

2015 IEEE Congress on Evolutionary Computation (CEC), pp. 870–876. IEEE, Piscataway, NJ (2015)
115. Beegom, A.A., Rajasree, M.: Integer-pso: a discrete pso algorithm for task scheduling in cloud computing systems. Evol. Intell. 12(2), 227–239 (2019)
116. Kumar, M., Sharma, S., Goel, S., Mishra, S.K., Husain, A.: Autonomic cloud resource provisioning and scheduling using meta-heuristic algorithm. Neural. Comput. Appl. 32(24), 18285–18303 (2020)
117. Alsaidy, S.A., Abbood, A.D., Sahib, M.A.: Heuristic initialization of PSO task scheduling algorithm in cloud computing. J. King Saud Univ. 34(6), 2370–2382 (2022)
118. Abdulhamid, S.M., Latiff, M.S.A., Idris, I.: Tasks scheduling technique using league championship algorithm for makespan minimization in IaaS cloud. ARPN J. Eng. Appl. Sci. 9(12), 2528–2533 (2015)
119. Abdulhamid, S.I.M., Latiff, M.S.A., Abdul-Salaam, G., Madni, S.H.H.: Secure scientific applications scheduling technique for cloud computing environment using global league championship algorithm. PLoS One 11(7), e0158102 (2016)
120. Abdulhamid, S.M., Latiff, M.S.A., Madni, S.H.H., Abdullahi, M.: Fault tolerance aware scheduling technique for cloud computing environment using dynamic clustering algorithm. Neural. Comput. Appl. 26(7), 1–15 (2016)
121. Xavier, V.A., Annadurai, S.: Chaotic social spider algorithm for load balance aware task scheduling in cloud computing. Cluster Comput. 22(1), 287–297 (2019)
122. Abdullahi, M., Ngadi, M.A.: Symbiotic organism search optimization based task scheduling in cloud computing environment. Future Gener. Comput. Syst. 56, 640–650 (2016)
123. Kumar, K.P., Kousalya, K., Vishnuppriya, S.: DSOS with local search for task scheduling in cloud environment. In: 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 1–4. IEEE, Piscataway, NJ (2017)
124. Abdullahi, M., Ngadi, M.A., Dishing, S.I., Abdulhamid, S.M., Ahmad, B.I.: An efficient symbiotic organisms search algorithm with chaotic optimization strategy for multi-objective task scheduling problems in cloud computing environment. J. Netw. Comput. Appl. 133, 60–74 (2019)
125. Chen, X. et al.: A woa-based optimization approach for task scheduling in cloud computing systems. IEEE Syst. J. 14(3), 3117–3128 (2020)
126. Natesan, G., Chokkalingam, A.: An improved grey wolf optimization algorithm based task scheduling in cloud computing environment. Int. Arab J. Inf. Technol. 17(1), 73–81 (2020)
127. Gill, S.S., Buyya, R., Chana, I., Singh, M., Abraham, A.: BULLET: Particle swarm optimization based scheduling technique for provisioned cloud resources. J. Netw. Syst. Manag. 26(2), 361–400 (2018)
128. Awad, A., El-Hefnawy, N., Abdel_kader, H.: Enhanced particle swarm optimization for task scheduling in cloud computing environments. Procedia Comput. Sci. 65, 920–929 (2015)
129. Aissi, H., Bazgan, C., Vanderpooten, D.: Complexity of the min–max and min–max regret assignment problems. Oper. Res. Lett. 33(6), 634–640 (2005)
130. Wu, M.-Y., Shu, W., Zhang, H.: Segmented min-min: A static mapping algorithm for meta-tasks on heterogeneous computing systems. in hcw, IEEE 1(1), 1–11 (2000)
131. Sindhu, S., Mukherjee, S.: Efficient task scheduling algorithms for cloud computing environment. In: High Performance Architecture and Grid Computing. pp. 79–83. Springer, Heidelberg (2011)
132. Mao, Y., Chen, X., Li, X.: Max–min task scheduling algorithm for load balance in cloud computing. In: Proceedings of International Conference on Computer Science and Information Technology, pp. 457–465. Springer, Berlin (2014)
133. Devipriya, S., Ramesh, C.: Improved max-min heuristic model for task scheduling in cloud. In: 2013 International Conference on Green Computing, Communication and Conservation of Energy (ICGCE), pp. 883–888. IEEE, Piscataway, NJ (2013)
134. Li, X., Mao, Y., Xiao, X., Zhuang, Y.: An improved max-min task-scheduling algorithm for elastic cloud. In: 2014 International Symposium on Computer, Consumer and Control (IS3C). pp. 340–343. IEEE, Piscataway, NJ (2014)
135. Wang, G., Yu, H.C.: Task scheduling algorithm based on improved min-min algorithm in cloud computing environment. J. Appl. Mech. 303, 2429–2432 (2013)
136. Al-Maytami, B.A., Fan, P., Hussain, A., Baker, T., Liatsis, P.: A task scheduling algorithm with improved makespan based on prediction of tasks computation time algorithm for cloud computing. IEEE Access 7, 160916–160926 (2019)
137. Nabi, M., Toeroe, M., Khendek, F.: Availability in the cloud: State of the art. J. Netw. Comput. Appl. 60, 54–67 (2016)
138. Faheem, M., Al-Khasawneh, M.A.: Multilayer cyberattacks identification and classification using machine learning in internet of blockchain (IoBC)-based energy networks. Data Brief 54, 110461 (2024)
139. Faheem, M., Al-Khasawneh, M.A., Khan, A.A., Madni, S.H.H.: Cyberattack patterns in blockchain-based communication networks for distributed renewable energy systems: a study on big datasets. Data Brief 53, 110212 (2024)
140. Madni, S.H.H., Latiff, M.S.A., Coulibaly, Y., Abdulhamid, S.I.M.: An appraisal of meta-heuristic resource allocation techniques for IaaS cloud. J. Sci. Technol. 9(4), 1–14 (2016)
141. Raza, B., Kumar, Y.J., Malik, A.K., Anjum, A., & Faheem, M.: Performance prediction and adaptation for database management system workload using case-based reasoning approach. Inf. Syst. 76, 46–58 (2018)
142. Faheem, M., Kuusniemi, H.,Eltahawy, B., Bhutta, M.S., Raza, B.: A lightweight smart contracts framework for blockchain-based secure communication in smart grid applications. IET Gener. Transm. Distrib. 18, 625–638 (2024)
143. Butt, R.A., Ashraf, M.W., Faheem, M., Idrus, S.M.: A survey of dynamic bandwidth assignment schemes for TDM-based passive optical network. J. Opt. Commun. 41(3), 279–293 (2020)
144. Ali, G., Dastgir, A., Iqbal, M.W., Anwar, M., Faheem, M.: A hybrid convolutional neural network model for automatic diabetic retinopathy classification from fundus images. IEEE J. Transl. Eng. Health Med. 11, 341–350 (2023)
145. Malik, H., Anees, T., Faheem, M., Chaudhry, M.U., Ali, A., Asghar, M.N.: Blockchain and Internet of Things in smart cities and drug supply management: Open issues, opportunities, and future directions. Internet Things 23, 100860 (2023)
146. Bashir, M.B., Abd Latiff, M.S., Ahmed, A.A., Yousif, A., Eltayeb, M.E.: Content-based information retrieval techniques based on grid computing: A review. IETE Tech. Rev. 30(3), 223–232 (2013)
147. Armbrust, M. et al.: A view of cloud computing. Commun. ACM 53(4), 50–58 (2010)
148. Shaukat, M., Alasmary, W., Alanazi, E., Shuja, J., Madani, S.A., Hsu, C.-H.: Balanced energy-aware and fault-tolerant data center scheduling. Sensors 22(4), 1482 (2022)
149. Beloglazov, A., Buyya, R.: Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. Concurr. Comput. Pract. Exp. 24(13), 1397–1420 (2012)
150. Younge, A.J., Von Laszewski, G., Wang, L., Lopez-Alarcon, S., Carithers, W.: Efficient resource management for cloud computing environments. In: 2010 International Green Computing Conference, pp. 357–364. IEEE, Piscataway, NJ (2010)
151. Faragardi, H.R., Shojaee, R., Tabani, H., Rajabi, A.: An analytical model to evaluate reliability of cloud computing systems in the presence of QoS requirements. In: 2013 IEEE/ACIS 12th International Conference on Computer and Information Science (ICIS), pp. 315–321. IEEE, Piscataway, NJ (2013)
152. Zhang, Q., Cheng, L., Boutaba, R.: Cloud computing: state-of-the-art and research challenges. J. Internet Serv. Appl. 1(1), 7–18 (2010)
153. Burhan, M., Alam, H., Arsalan, A., Rehman, R.A., Anwar, M., Faheem, M., Ashraf, M.W.: A comprehensive survey on the cooperation of fog computing paradigm-based IoT applications: layered architecture, real-time security issues, and solutions. IEEE Access. 11(6), 73303–73329 (2023)
154. Lawler, E.L., Lenstra, J.K., Kan, A.H.R., Shmoys, D.B.: Sequencing and scheduling: Algorithms and complexity. In: Handbooks in

Operations Research and Management Science. vol. 4, pp. 445–522. Elsevier, Amsterdam (1993)

155. Panwalkar, S.S., Iskander, W.: A survey of scheduling rules. Oper. Res. 25(1), 45–61 (1977)
156. Raza, B., Aslam, A., Sher, A., Malik, A.K., Faheem, M.: Autonomic performance prediction framework for data warehouse queries using lazy learning approach. Appl. Soft Comput. 91, 106216 (2020)
157. Ashraf, Waqar, M., Idrus, SM., Iqbal, F., Butt, R.A., Faheem, M.: Disaster-resilient optical network survivability: a comprehensive survey. Photonics 5(4), 35 (2018)
158. Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A., Buyya, R.: CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw.: Pract. Exper. 41(1), 23–50 (2011)
159. Buyya, R., Ranjan, R., Calheiros, R.N.: Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. In: International Conference on High Performance Computing & Simulation, 2009. HPCS'09, pp. 1–11. IEEE, Piscataway, NJ (2009)
160. Buyya, R., Murshed, M.: Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. Concurr. Comput. Pract. Exp. 14(13-15), 1175–1220 (2002)
161. Casanova, H.: Simgrid: A toolkit for the simulation of application scheduling. In: Proceedings of First IEEE/ACM International Symposium on Cluster Computing and the Grid, pp. 430–437. IEEE, Piscataway, NJ (2001)
162. HPC2N. The HPC2N Seth log (2016). http://www.cs.huji.ac.il/labs/parallel/workload/l_hpc2n/. Accessed 12 Jan 2023
163. NASA. The NASA Ames iPCS/860 log (2016). http://www.cs.huji.ac.il/labs/parallel/workload/l_nasa_ipsc/. Accessed 2 Jan 2023
164. Alarood, A.A., Faheem, M., Al-Khasawneh, M.A., Alzahrani, A.I., Alshdadi, A.A.: Secure medical image transmission using deep neural network in e-health applications. Healthcare Technol. Lett. 10(4), 87–98 (2023)
165. Faheem, M., Butt, R.A., Raza, B., Alquhayz, H., Abbas, M.Z., Ngadi, M.A., Gungor, V.C.: A multiobjective, lion mating optimization inspired routing protocol for wireless body area sensor network based healthcare applications. Sensors 19(23), 5072 (2019)
166. Bashir, R.N., Saeed, M., Al-Sarem, M., Marie, R., Faheem, M., Karrar, A.E., Elhussein, B.: Smart reference evapotranspiration using Internet of Things and hybrid ensemble machine learning approach. Internet Things 24, 100962 (2023)
167. Shah, S., Madni, S.H.H., Hashim, S.Z.B.M., Ali, J., Faheem, M., Ali, N.A.: Assessing the prioritization of key influencing factors for industrial IoT readiness in SMEs. In: International Conference of Reliable Information and Communication Technology, pp. 252–262. Springer Nature Switzerland, Cham (2023)
168. Blose, M., Akinyemi, L., Ojo, S., Faheem, M., Imoize, A.L., Khan, A.A.: Scalable hybrid switching-driven software defined networking issue: From the perspective of reinforcement learning. IEEE Access 12(3), 63334–63350 (2024)
169. Sivakumar, S., Logeshwaran, J., Kannadasan, R., Faheem, M., Ravikumar, D.: A novel energy optimization framework to enhance the performance of sensor nodes in Industry 4.0. Energy Sci. Eng. 12(3), 835–859 (2024)
170. Shah, S., Madni, S.H.H., Hashim, S.Z.B.M., Ali, J., Faheem, M.: Factors influencing the adoption of industrial internet of things for the manufacturing and production small and medium enterprises in developing countries. IET Collab. Intell. Manuf. 6(1), e12093 (2024)
171. Ali, G., et al.: A hybrid convolutional neural network model for automatic diabetic retinopathy classification from fundus images. IEEE J. Transl. Eng. Health Med. 11(2), 341–350 (2023).
172. Kawoosa, A.I., Prashar, D., Faheem, M., Jha, N., Khan, A.A.: Using machine learning ensemble method for detection of energy theft in smart meters. IET Gener. Transm. Distrib. 17(21), 4794–4809 (2023).
173. Al-Khasawneh, M.A.S., Faheem, M., Aldhahri, E.A., Alzahrani, A., Alarood, A.A.: A MapReduce based approach for secure batch satellite image encryption. IEEE Access 11(4), 62865–62878 (2023).

How to cite this article: Madni, S.H.H., Faheem, M., Younas, M., Masum, M.H., Shah, S.: Critical review on resource scheduling in IaaS clouds: Taxonomy, issues, challenges, and future directions. J. Eng. 2024, e12420 (2024). https://doi.org/10.1049/tje2.12420