Cluster Computing (2023) 26:3037–3067
https://doi.org/10.1007/s10586-023-04090-y

A review of task scheduling in cloud computing based on nature-inspired optimization algorithm

Farida Siddiqi Prity[1] · Md. Hasan Gazi[2] · K. M. Aslam Uddin[1]

Received: 10 May 2023 / Revised: 30 May 2023 / Accepted: 11 June 2023 / Published online: 29 June 2023

Abstract

The advent of the cloud computing paradigm allowed multiple organizations to move, compute, and host their applications in the cloud environment, enabling seamless access to a wide range of services with minimal effort. An efficient and dynamic task scheduler is required to handle concurrent user requests for cloud services using various heterogeneous and diversified resources. Improper scheduling can lead to challenges with under or over-utilization of resources, which could waste cloud resources or degrade service performance. Nature-inspired optimization techniques have been proven effective at solving scheduling problems. This paper accomplishes a review of nature-inspired optimization techniques for scheduling tasks in cloud computing. A novel classification taxonomy and comparative review of these techniques in cloud computing are presented in this research. The taxonomy of nature-inspired scheduling techniques is categorized as per the scheduling algorithms, nature of the scheduling problem, type of tasks, the primary objective of scheduling, task-resource mapping scheme, scheduling constraint, and testing environment. Additionally, guidelines for future research issues are also provided, which should undoubtedly benefit researchers and practitioners as well as open the door for newcomers eager to pursue their glory in the field of cloud task scheduling.

Keywords Cloud computing · Resource · Task scheduling · Nature-inspired optimization · Taxonomy

1 Introduction

The ever-expanding scope of the transnational corporate atmosphere has prompted the establishment of sophisticated technology to handle it [1]. One such innovation that can expedite software deployment processes and save time and effort is cloud computing [2]. Cloud computing is the practice of accessing and conserving information and applications via the internet rather than locally stored systems at the consumer's disposal [3, 4]. It uses distant servers connected to the internet to store, administer, and provide online access to information. Organizations must establish sophisticated cloud management to manage and govern the utilization of cloud resources while assuring appropriate, reliable, and adaptable hardware and software that optimizes costs. The early phases of preparing should include developing the cloud deployment approach to the management of clouds [5, 6]. Due to the low probability of infrastructure breakdowns in the cloud, servers are consistently and widely accessible. Cloud computing allows numerous individuals to utilize various applications efficiently and at a lower cost due to the collaboration of shared infrastructures [7].

In cloud computing, a scheduler (broker) is designed to identify possible approaches for allocating a collection of readily accessible scarce resources to new applications to maximize scheduling targets such as makespan, computational cost, monetary cost, reliability, availability, resource utilization, response time, energy consumption, etc. [8, 9]. The scheduler devises strategies for allocating the proper assignments to constrained resources to maximize scheduling outcomes [10]. The scheduling algorithm aims to decide which task will be carried out based on which resource [11]. This approach encourages accessibility to

✉ K. M. Aslam Uddin
aslam@nstu.edu.bd

[1] Department of Information and Communication Engineering, Noakhali Science and Technology University, Noakhali 3814, Bangladesh

[2] High Court Division, Supreme Court of Bangladesh, Dhaka 1000, Bangladesh

the collaborative resource stream while providing the appropriate consumers with a QOS assurance. The primary objective of a novel scheduling approach is to determine the best combination of resources that can be employed to accomplish an incoming task so that a scheduling algorithm can be used to maximize a variety of QOS factors, including expenses, makespan, flexibility, trustworthiness, task disapproval proportion, the efficiency of resources, utilization of energy, etc., and meet restrictions, such as the deadline and budgetary constraints, etc., to prevent the issue of load imbalance. The solution to user satisfaction is achieving the intended efficiency, which may be accomplished by rendering the primary objective of cloud computing services [12]. Today, a wide range of applications have been developed that employ the scheduling principle, including power system management, scheduling of multimodal contents on the global web, and electronic circuit board production [13].

Task scheduling in cloud computing is a complicated topic that necessitates tackling a number of effectiveness, resource utilization, and workload administration obstacles while additionally considering diverse assets, task dependencies, QoS specifications, assurances, and confidentiality issues. The majority of task scheduling challenges are either NP-complete or NP-hard. As a result, it takes a very long time to develop an optimal solution compared to other alternatives [14]. There are no particular techniques for obtaining polynomial-time responses to these issues. Taillard suggested a scenario in which 0.02 to 1.01 percent of candidate solutions require the total amount of time necessary to reach the best solution [15]. This illustration demonstrates how challenging it can be to identify the optimum solution to a complex situation. Most researchers have therefore been motivated to pursue an appropriate scheduling algorithm to identify a quick yet effective solution to this scheduling challenge. The cloud service provider must offer an effective and optimal scheduling procedure that considers a number of factors, including cost, time, and SLA standards that must be followed as established by the users.

1.1 Motivation of the research

The two primary scheduling techniques modern computer systems use are exhaustive algorithms and Deterministic Algorithms (DAs) [16]. In terms of efficiency, DAs are substantially better than both conventional (exhaustive) and heuristic methods for scheduling problems. However, DAs have two key drawbacks: first, they were designed to handle some data distributions, and second, only some DAs can handle complex scheduling issues. Meta-heuristic algorithms, also known as approximation algorithms, use iterative techniques to obtain optimal solutions faster than 7DAs and exhaustive algorithms [16–19]. Meta-heuristic algorithms are classified into nature-inspired and non-nature-inspired [20]. Numerous study findings show that nature-inspired optimization algorithms produce superior scheduling outcomes than conventional and heuristic ones [21, 22]. While numerous nature-inspired optimization scheduling techniques have been successfully used in various computing contexts, including grid and clustering computing, they have yet to be specifically designed for the cloud. As a result, nature optimization algorithms may initially appear inappropriate as a scheduling option for cloud tasks to the general public. Encouraged by this misunderstanding, this study not only offers a systematic overview of scheduling methods used in the cloud environment from a nature-inspired perspective but also establishes a connection between conventional/heuristic scheduling methods and nature-inspired meta-heuristic ones so that cloud researchers who are still enthusiastic about conventional/heuristic scheduling can transit to scheduling based on nature-inspired optimization swiftly. The purpose of discussing both conventional and heuristic algorithms before discussing nature-inspired optimization algorithms is to clarify the differentiation between them easier.

We are also inspired by ideas from peer surveys in earlier works of literature. Task scheduling is a crucial component of cloud computing, which aims to increase VM utilization while lowering data center operating costs, leading to appreciable advancements in QoS metrics and overall performance. With an effective task scheduling technique, a high number of user requests may be processed appropriately and assigned to suitable VMs, which helps to meet the needs of cloud users and service providers more effectively. We carefully reviewed numerous nature-inspired scheduling approaches in the literature, and we discovered that most of the studies of literature do not cover all the aspects such as QoS-based comparative analysis, state-of-the-art, taxonomy, graphical representations, open issues and comparison of simulation tools of task scheduling as shown in Table 1. Because of this, conducting a thorough assessment of task scheduling utilizing nature-inspired optimization is vitally necessary to keep up with the field's continuing, expanding research.

1.2 Contributions of the present study

Although nature-inspired scheduling methods considerably impact cloud services, essential methodologies and backgrounds of this sector still need to be thoroughly and methodically evaluated. As a result, this study aims to compare the prior strategies and assess and critique current cloud scheduling systems in light of nature-inspired algorithms. A unique classification scheme (taxonomy) and a

Table 1 Comparison between existing works and the present study

Publication	Taxonomy	State-of-the-art	QoS-based comparative analysis	Graphical representations	Open issues	Comparison of simulation tools
[23]	No	No	No	No	No	No
[24]	Yes	Yes	No	No	Yes	No
[25]	No	Yes	No	No	Yes	No
[26]	Yes	Yes	No	No	Yes	No
[27]	Yes	No	Yes	No	No	Yes
Present Study	Yes	Yes	Yes	Yes	Yes	Yes

thorough analysis of contemporary nature-inspired scheduling approaches in cloud computing are provided. The effectiveness of existing techniques is evaluated based on qualitative QoS parameter-based criteria. The comparison of numerous simulation tools frequently used in the cloud is also presented. Through extensive investigation and discussion, results related to the pertinent aspects are validated. Finally, research issues are compiled to create a research roadmap that could include potential future study areas and existing trends.

2 Categorization of cloud task scheduling schemes

Cloud task scheduling scheme has been divided into three groups: traditional scheduling, heuristics scheduling, and meta-heuristics scheduling. This section has compared the traditional and heuristics strategies and evaluates and assesses the cloud computing system in light of nature-inspired optimization algorithms.

2.1 Traditional scheduling

Scheduling can be defined as assigning tasks to a set of provided machines under the constraints of objective function optimization. The scheduling challenge is called a single-processor scheduling issue when only one machine exists. The scheduling issue is seen as a multiprocessor scheduling when more than one machine is involved. Based on resource consumption costs, makespan, load balancing, and QoS and its variants, scheduling algorithm performance can be measured. In the context of traditional scheduling, a hierarchy of difficulties was established through numerous investigations [28, 29] concerning the characteristics of tasks (i.e., weight, due date, release date, and processing time), machines (i.e., single or multiple), as well as many other details, such as online vs. offline, batch vs. non-batch, precedence vs. non-precedence, sequence-dependent vs. sequence-independent. Scheduling problems are typically distinguished from one another and described using these limitations. All scheduling issues discussed in [28] have been explained using a three-fold notation. In this approach, a single or parallel machine designates the machine type, sequence-dependent or sequence-independent specifies the processing features and limitations, and makespan indicates the measure's value. According to the length of the scheduling time, the schedule was divided into four categories [29]: short-range, middle-range, long-range, and reactive control/scheduling.

2.2 Heuristics scheduling

Heuristic algorithms vary in performance depending on the problem they are used for; although they work well for some situations, they struggle with others. Heuristics typically deliver a precise response in a reasonable amount of time for a particular type of problem, but they need to improve when it comes to challenging optimization problems. Numerous heuristic techniques have been developed, such as Min-Min [30], Max–Min [30], First Come First Serve (FCFS) [31], Shortest Job First (SJF) [32], Round Robin (RR) [33], Heterogeneous Earliest Finish Time (HEFT) [34], Minimum Completion Time (MCT) [35], and Sufferage [36] in cloud environments to address the scheduling issues relating to workflow and independent tasks/applications.

In the Min-Min heuristics algorithm, the shortest task that can be completed in the shortest amount of time is chosen from among all tasks provided, and it is mapped to a virtual machine that will take the shortest amount of time to complete. When all tasks are successfully scheduled, the procedure is repeated, lengthening the overall makespan as each task's completion time lengthens. Large jobs must wait until smaller ones are completed before being handled by this method, which handles minor chores efficiently. This approach dramatically increases system throughput overall; however, heavy workloads may cause a starvation problem. Chen et al. [37] and Amalarethinam and Kavitha [38] used Min-Min in their proposed work. A set of issues, such as slow task execution, difficulties in the deadline, and priority problems, arise during the execution of the Min-Min algorithm. The max–Min algorithm created the problem of potential resource over-utilization and under-utilization in [39] and [40]. Non-continuous monitoring of nodes, load imbalance, cost, and time overheads due to unmanaged communication and storage, and static task scheduling problems are introduced while implementing Heterogeneous Earliest Finish Time (HEFT) algorithms [41, 42]. SJF can not solve the problem of starvation and load imbalance [43–45]. Round Robin needs to improve its ability to balance the load [46, 47]. Sufferage performs flawlessly in many situations; however, this technique has drawbacks if numerous tasks have identical suffrage values [48]. In this scenario, the first arriving task is chosen and run without considering other tasks, which may lead to a starving problem.

2.3 Meta-heuristic scheduling

Due to their efficiency in resolving complex and extensive computational challenges, meta-heuristic algorithms have significantly increased in popularity. Task scheduling is explained by applying heuristic and meta-heuristic methods to obtain optimal or nearly optimal solutions because traditional methods frequently fail to comprehend situations to their optimality. Heuristic solutions frequently become caught in a local minima dilemma, and meta-heuristic algorithms are the most effective way to escape this condition, as mentioned in [24, 26]. Meta-heuristic algorithms efficiently explore the search space to find a sub- or near-optimal solution to NP-complete problems. It also doesn’t depend on the problem being solved; it typically uses approximation rather than determinism. Because of their intuitive independence from the problem to be addressed, meta-heuristic algorithms are valuable for tackling problems in various domains with highly acceptable performance. Researchers have systematically evaluated the application of meta-heuristic algorithms on scheduling features in cloud and grid settings [49–52]. Meta-heuristic techniques are frequently employed as very efficient solutions to NP-hard optimization issues. Meta-heuristic algorithms are classified into two types [53]:

- Nature-inspired
- Non-nature-inspired.

3 Nature-inspired meta-heuristics scheduling

Nature-inspired algorithms are optimization methods that handle challenging optimization issues by emulating the behavior of natural systems [54]. These algorithms effectively locate optimal solutions for multi-dimensional and multi-modal issues. This paper conducts a schematic review of the nature-inspired optimization techniques of task scheduling in the cloud computing application.

4 Taxonomy of nature-inspired task scheduling optimization algorithm in cloud computing

A novel, rigorous taxonomy is presented in Fig. 1 using a number of principal methodologies used in the literature to more thoroughly and clearly comprehend the nature-inspired task scheduling approaches in cloud computing. This taxonomy divides the methods into seven main divisions based on the type of scheduling algorithm (scheduler), nature of the scheduling problem, nature of the task, primary objectives of scheduling, task-resource mapping schemes, scheduling constraints, and testing environment.

4.1 Scheduling algorithm

Nature-inspired scheduling algorithms can be divided into four major categories [55] such as:

- Evolutionary-based
- Swarm-based
- Physics-based
- Hybrid approaches

This study has discussed various algorithms of these four categories of nature-inspired optimization algorithms for task scheduling in cloud computing. Figure 2 illustrates the overall summary of all the nature-inspired optimization algorithms discussed in this section.

4.1.1 Evolutionary-based algorithm

The principles of natural evolution serve as the foundation for evolutionary approaches [56]. An initially produced population that evolved over several generations serves as the basis for the search procedure. The best individuals are constantly brought together to create the next generation, one of these approaches’ most vital points. As a result, the population can be improved over several generations. Genetic Algorithm (GA) [57], Memetic Algorithm (MA) [58], Evolution Strategy (ES) [59], Probability-Based

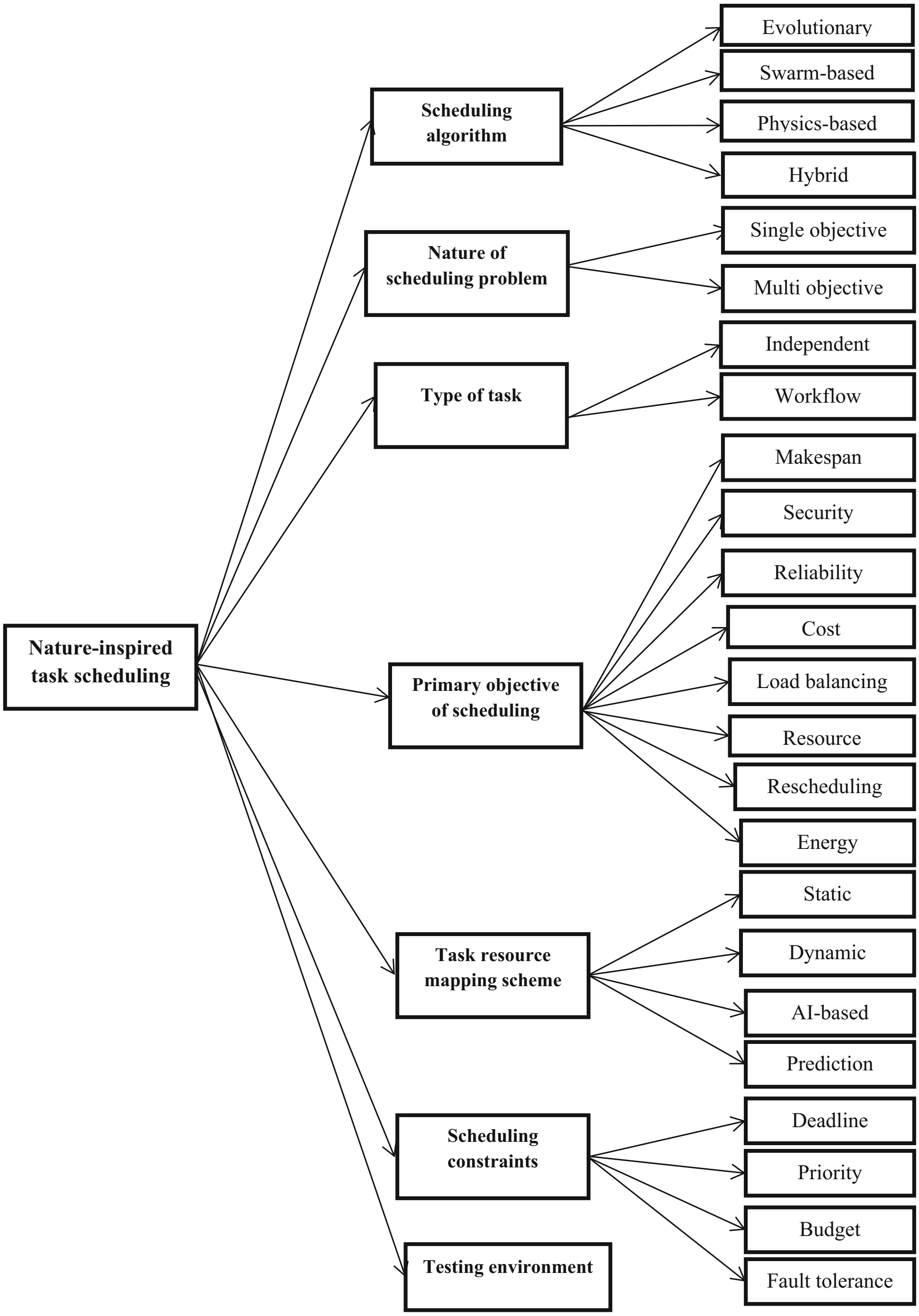

Fig. 1 Taxonomy of nature-inspired task scheduling approaches

Incremental Learning (PBIL) [60], Genetic Programming (GP) [61], Lion Optimization Algorithm (LOA) [62], Imperialist Competitive Algorithm (ICA) [63], Sun Flower Optimization Alg (SFO) [64], and Biogeography-Based Optimizer (BBO) [65] are some of the well-known algorithms. In this section, some evolutionary nature-inspired algorithms that are used in task scheduling of cloud computing are briefly summarized.

4.1.1.1 Genetic algorithm The Darwinian notion of "survival of the fittest" served as the foundation for the development of the genetic algorithm (GA), which is bio-inspired in that fitness is increased through the process of evolution through reproduction [66]. A Genetic Algorithm analyzes new regions of the solution space while utilizing the best solutions from completed searches. Chromosomes, composed of a collection of components called genes, can represent any solution to a particular problem. Crossover or

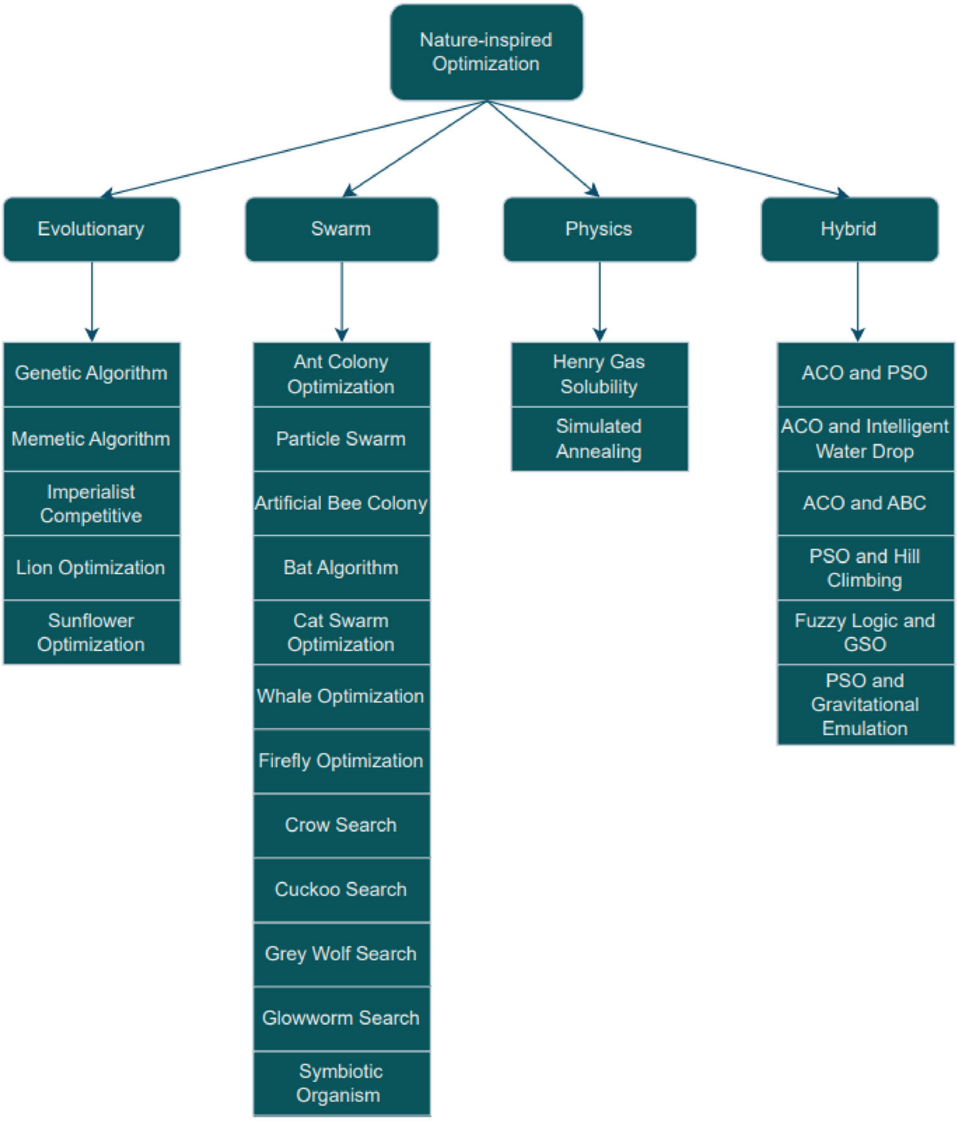

Fig. 2 Nature-inspired optimization algorithms discussed in the proposed study

mutation operators produce new offspring chromosomes once the population has been initialized with randomly produced solutions [67]. The generation of the children is repeated until enough of the best offspring are produced to find the best outcome. The scheduling problem response can be expressed using a variety of representation strategies. Some researchers [68] have encoded the solutions using fixed-length binary strings. A chromosomal matrix is utilized to depict the mapping of tasks on resources in direct representation [69–73]. Paper [74] employed the min-min heuristic and minimum execution time to create the initial population. When populating the population, the order of the tasks was also considered [74–76], and genetic operations were then used to solve the workflow scheduling issue. In [76], Round Robin and best-fit techniques identify potential solutions for allocating jobs to resources. According to [77], crossover and mutation operations are carried out based on the level-wise representation after tasks have been organized according to the order of their workflow level. In reference [78], the authors used crossover and mutation operators. The chromosomes are depicted as 2D strings in [79] based on the timeframe and financial constraints. HEFT is paired with a genetic algorithm [80] to create a scheduling plan. Authors [81] have suggested two chromosomes: an ordering chromosome that specifies the execution order following the scientific workflow representation and an allocation chromosome that contains the assignment of tasks to nodes. A task scheduling technique based on the Shadow Price-guided Genetic Algorithm (SGA) was presented by [82]. The DVFS technique is applied to the grid environment to reduce energy consumption and maximize makespan [83]. The authors of [84] describe a multi-agent evolutionary method for distributing load among virtual computers.

A VM allocation methodology with appropriate load balancing and system resource utilization is provided by the method utilized by [85]. The migration cost is optimized through elitism selection and tree structure encoding.

4.1.1.2 Memetic algorithm

Dawkins' Memetic Algorithm (MA) [86] is based on the term meme. Memes are concepts like rumors and stories that spread across the community of meme carriers. MAs use a specific local search procedure to enhance or improve individual fitness. Many issues in the real world have been addressed using MA. In [87], the authors used a memetic method to solve a hybrid flow shop scheduling problem involving multiprocessor activities. Researchers have employed MA to address the traveling salesperson problem and quadratic assignment [88]. The memetic has been used with hill climbing, tabu search, and simulated annealing for task scheduling [89, 90]. The best schedule for a workflow application running on a multiprocessor system uses MA for multiprocessor scheduling [91]. The authors of [92] used this algorithm to shorten the schedule. A global search optimization based on the Particle Swarm Optimization (PSO) technique produces candidate solutions.

4.1.1.3 Imperialist competitive algorithm

The mathematical description of the imperialist competition, which improves optimization outcomes, inspired the Imperialist Competitive Algorithm (ICA) [93]. Imperialism refers to using and controlling another nation's economic, political, and human resources by direct legislation. Numerous academics have used ICA-based optimization techniques to address the scheduling issue. An ICA-based technique is used for the flow shop scheduling problem [94]. By reducing waiting time in [95], identical work is extended to reduce maximum completion time. In [96], a bi-objective parallel machine scheduling problem with an emphasis on minimizing mean task completion time was addressed. A power-aware load balancing technique reduces energy consumption in cloud computing data centers based on ICA [97]. A cost-effective resource provisioning mechanism is provided in [98] to assign virtual machines with reservations on an on-demand basis. Makespan is optimized for independent tasks in grid computing [99] utilizing the imperialist competition method. A hybrid technique is used for job shop scheduling in [100] to reduce the makespan. A Gravitational Attraction Search (GAS) technique was introduced in [101] and was integrated with ICS in a different study to optimize the service composition problem in cloud computing more quickly. The authors have expanded their work by combining a PROCLUS classifier with an ICA algorithm to choose the best service provider for particular services [102]. In [103], authors developed an ICA-based scheduling method that takes execution time and execution cost parameters into account for activities that depend on scheduling. In [104], the authors offered an online scheduler based on the ICA and established a dependability model for cloud systems.

4.1.1.4 Lion optimization algorithm

The Lion Optimization Algorithm (LOA) is a meta-heuristic optimization algorithm inspired by the social behavior of lions. The social structure of lions, in which young are born from resident males and females, is called pride. The lion's territorial defense and territorial takeover behavior have been used to solve optimization problems [105]. The lion's behaviors of hunting, going toward a safe area, wandering, and migration were included by the authors in [106]. The proposed work has been contrasted with algorithms for Invasive Weed Optimization (IWO), Biogeography-based Optimization (BBO), Gravitational Search (GSA), Hunting Search (HuS), Bat Algorithm (BA), and Water Wave Optimization (WWO). The simulation outcomes show how effective the LOA algorithm is in solving many other optimization issues.

4.1.1.5 Sunflower optimization algorithm

The Sunflower Optimization Algorithm (SFO) is a population-based algorithm modeled after a natural process that moves sunflowers toward the sun. The SFO is a new meta-heuristic algorithm motivated by sunflowers traveling toward the sun while considering neighboring sunflower pollination. The performance of the current task scheduling is improved by the Enhanced Sunflower Optimization (ESFO) algorithm introduced in the paper [107]. It discovers the best scheduling strategy in polynomial time. An effective hybrid optimization algorithm named Sunflower Whale Optimization Algorithm (SFWOA) is proposed in [108]. The authors of the paper [109] suggest an Opposition-based Sunflower Optimization (OSFO) algorithm improve the efficiency of the task schedulers that are already in use in terms of cost, energy, and makespan. Paper [110] suggests a sunflower optimization algorithm with a sine–cosine algorithm (SFOA-SCA) for enhancing the effectiveness of load balancing in cloud networks.

4.1.2 Swarm-based algorithm

Swarm Intelligence (SI) is a relatively recent method of problem-solving that draws its inspiration from the social behavior of insects and other animals and the collective intelligence of swarms of biological populations [111]. A computational and behavioral paradigm called SI uses the interplay of small information processing units to address a dispersed problem. The most widely used algorithms are Particle Swarm Optimization (PSO) [112], Marriage in Honey Bees Optimization Algorithm (MBO) [113], Whale

Optimization [114], Firefly Optimization [115], Artificial Bee Colony (ABC) [116], Ant Colony Optimization (ACO) [117], Artificial Fish-Swarm Algorithm (AFSA) [118], Bat algorithm (BA) [119], Cat Swarm Optimization [120], Termite Algorithm [121], Wasp Swarm Algorithm [122], Monkey Search [123], Wolf Pack Search Algorithm [124], Bee Collecting Pollen Algorithm (BCPA) [125], Symbiotic organism search optimization algorithm [126], Crow Search, Cuckoo Search [127], Dolphin Partner Optimization (DPO) [128], Grey Wolf Search Algorithm [129], Glow worm optimization [130], etc. In this section, some swarm intelligence nature-inspired algorithms that are used in task scheduling of cloud computing are briefly summarized.

4.1.2.1 Ant colony optimization

Marco Dorigo created the ant colony optimization (ACO). The foraging behavior of several ant species inspired it. The pheromone that the ants deposit on the ground instructs the other ants to follow the path. Pheromone values are employed to seek solution space, and the ants record their positions and the caliber of their solutions to identify an optimum solution [131]. Paper [132] suggests using the ACO technique to specify the task and resource selection criteria in clusters. The user specifies QoS limitations in [133] to get the appropriate quality for the scheduling workflow. The paper [134] presents an Ant Colony System (ACS)-based workflow scheduling algorithm that has been enhanced with several additional features. The ACO algorithm and the knowledge matrix notion are combined in [135]. The researchers adopted the ACO approach [136] to track the historical desirability of putting them in the same physical machine. The paper [137] has developed an ant colony-based energy-efficient scheduling algorithm. The updated pheromone schemes are presented in [138, 139]. The population is created using the concept of biased starting ants from [140], where the expected time to complete a task and the standard deviations of tasks are considered. An infinite number of ants are employed in [141] for the grid or cloud-based scheduling of interdependent jobs or workflows. Researchers in [142] suggest independent work scheduling based on the ACO approach for cloud computing systems. Based on ACO, workflow scheduling for grid systems is suggested in [143]. The length of the schedule is kept to a minimum, and tests with time-varying workflow are run in a grid setting. When scheduling workflows in hybrid clouds, reference [144] considers deadline and cost.

4.1.2.2 Particle swarm optimization

Particle swarm optimization (PSO) was created by Eberhart and Kennedy [145] and is based on the social behavior of particles like flocking birds. The particles adjust their path based on their optimal position and the optimal position of the best particle across the board for each generation. The position and speed of the particles are initialized before the population of the particles is created at random [146–150]. Discrete PSO is paired with the Min-Min approach [151] to decrease the execution time of scheduling activities on computational grids. A combination of PSO and Gravitational Emulation Local Search (GELS) will improve the utilization of searching space [152]. Tasks are scheduled using a Hybrid Particle Swarm Optimization (HPSO) to reduce turnaround time and increase resource effectiveness [153]. In paper [154], Tabu Search (TS) and PSO are combined to use TS to create a local search mechanism. PSO reduces the completion time and improves resource utilization when paired with another local search approach called Cuckoo Search [155]. Task scheduling in grid contexts has been done via particle swarm optimization [156, 157]. A PSO-based hyper-heuristic for resource scheduling in the grid context was presented by R. Aron et al. [158]. In [159], a load rebalancing algorithm utilizing PSO and the least position value technique is used for task scheduling. PSO suggests a Task-based System Load Balancing method (TBSLB-PSO) in [160]. The method reduces transfer and task execution times. Improved makespan and resource utilization are found in [161]. Energy consumption is decreased to 67.5% using the particle swarm optimized Tabu search mechanism (PSOTBM) [162].

4.1.2.3 Artificial bee colony optimization

The Artificial Bee Colony (ABC) algorithm is based on how honey bees intelligently forage for food. Dervis Karaboga created this strategy in 2005 to address real-world issues [163]. Every bee in a colony cooperates to find food sources, and this knowledge is used to guide decision-making regarding the search for space exploration. Numerous combinatorial problems, including flow shop scheduling [164], on-shop scheduling [165], project scheduling [166], and traveling salesman [167], have been solved using this technique. [168, 169] provide examples of how bee colony optimization is used for task scheduling in distributed grid systems. For task mapping on the resources, authors have used bees' foraging behavior as a model. The strategy is described in [170], which also distributes the workload of parallel programs across the resources available in grid computing systems. Load balancing [171, 172] presents non-preemptive independent task scheduling based on ABC optimization. Compared to the Min-Min algorithm, resource utilization is enhanced by an average of 5.0383% when bee colony and PSO are combined [173]. ABC is integrated with a Memetic method to reduce the makespan and balance the load [174]. Many writers have used the ABC algorithm to schedule different tasks in cloud computing [175, 176]. The ABC algorithm was made possible for scheduling dependent tasks in a cloud context by several distinctive characteristics, such as modularity and parallelism [177, 178]. Energy-aware scheduling is suggested by [179] to effectively manage the resources and improve their usage in the cloud.

4.1.2.4 Bat algorithm

A swarm-based method called the Bat Algorithm (BA) imitates bats' echolocation activity [180]. Bats emit a sound pulse, and an echo is created when nearby objects reflect that sound pulse. Bats use the lag time between the signal's emission and return to determine the prey's precise location, distance, and speed. Paper [181] uses the BA algorithm in a cloud context for resource scheduling, which optimizes the makespan more effectively than the GA method. In [182], a hybrid technique for BA-Harmony search is suggested for cloud computing work scheduling. The Gravitational Scheduling Algorithm (GSA) [183] is a further extension of BA that considers time restrictions and a trust model. This approach chooses resources for task mapping based on their trust value. To reduce execution costs, the Authors used BA to resolve the workflow scheduling issue in the cloud [184]. The approach performs better in terms of processing costs comparing the algorithm to the best resource selection algorithm. PSO and the bat algorithm are combined in the paper [185] for cloud profit maximization.

4.1.2.5 Cat swarm optimization

The author [186] introduced the Cat Swarm Optimization (CSO) algorithm for continuous optimization issues. Based on the social behavior of cats, a heuristic algorithm for optimization is proposed. It is based on the seeking and tracking behavior modes of cats. CSO is used in the paper [187]. Researchers in [188, 189] provide many modified versions of CSO for resolving discrete optimization issues in various fields. For the zero–one knapsack issue and the traveling salesperson's problem, DBCSO [190] is a binary variant of CSO. In cloud computing, CSO has been utilized to address the scheduling of workflows while considering single and multiple objectives [191]. Makespan, computation expense, and CPU idle time were considered optimization criteria for mapping the dependent tasks. The traditional genetic algorithm is supplemented with CSO and DBCSO to initialize the population [192].

4.1.2.6 Whale optimization algorithm

An innovative method for handling optimization issues is the Whale Optimization Algorithm (WOA). Three operators are used in this algorithm to replicate how humpback whales hunt by searching for prey, circling prey, and using bubble nets. The bubble-net feeding method is the name of this foraging technique [193]. By optimizing the allocation of tasks to resources, the WOA algorithm can reduce the makespan and improve the overall performance of the cloud computing system. Mangalampalli et al. created a Multi-objective Trust-Aware Scheduler with Whale Optimization (MOTSWO) that prioritizes jobs and virtual machines and schedules them to the best virtual resources while consuming the least amount of time and energy possible [194]. The task scheduling technique presented by [195] allocates tasks to the appropriate VMS based on determining task and VM priorities. It is modeled using the WOA to reduce data center energy use and electricity costs. The multi-objective model and WOA are the foundation for the task scheduling algorithm proposed in this research, known as W-Scheduler [196]. To further expand the WOA-based method's capacity to find the best solutions, the authors offer Improved WOA for Cloud Task Scheduling (IWC) [197]. The suggested IWC offers superior convergence speed and accuracy in searching for the optimal task scheduling plans than the present meta-heuristic algorithms, according to simulation-based studies and comprehensive IWC implementation. Another IWC has been presented in [198]. The Vocalization of the Humpback Whale Optimization Algorithm (VWOA) is used to optimize task scheduling in a cloud computing environment [199]. It reduces time, cost, and energy consumption while maximizing resource use in terms of makespan, cost, degree of imbalance, resource utilization, and energy consumption.

4.1.2.7 Firefly algorithm

The flashing behavior of fireflies inspired the nature-inspired optimization method known as the Firefly Algorithm (FA) [200]. A population-based algorithm mimics the flashing behavior of fireflies to choose the best answer. An efficient Trust-Aware Task Scheduling algorithm using Firefly optimization has been presented in [201], which has shown a significant impact over the conventional approaches by minimizing the makespan, availability, success rate, and turnaround efficiency. An acceptable improvement for makespan and resource utilization using the FA has been presented in [202]. The Crow Search algorithm and FA are integrated to enhance global search capability [203]. A hybrid Firefly-Genetic combination is propounded for scheduling tasks in [204]. An intelligent meta-heuristic algorithm based on the combination of ICA and FA has been presented in [205] to show dramatic improvements in makespan, CPU time, load balancing, stability, and planning speed. Another hybrid method using Firefly and SA algorithms has been proposed in [206]. The Cat Swarm Optimization and the FA were combined into a hybrid multi-objective scheduling algorithm [207]. An original modified Firefly Algorithm (mFA) is used to construct and successfully optimize the operational cost minimization problem for DGDCs [208].

4.1.2.8 Crow search algorithm A novel kind of swarm intelligence optimization algorithm known as the "Crow Search Algorithm" was developed by imitating the intelligent behavior of crows in hiding and finding food [209]. The Crow Search method has been applied to optimization problems. The Crow Search is proposed in [210] for task scheduling in the cloud. Enhanced Crow Search Algorithm (ECSA) is proposed in [211] to improve the random selection of tasks. A technique called the Crow-Penguin Optimizer for Multi-objective Task Scheduling Strategy in Cloud Computing (CPO-MTS) is suggested in [212]. The suggested algorithm quickly determines how to best use the given tasks and cloud resources to complete them. In a multi-objective task scheduling environment, Singh et al. propose the Crow Search-based Load Balancing Algorithm (CSLBA), which focuses on allocating the best resources for the task to be implemented while considering different factors like the Average Makespan Time (AMT), Average Waiting Time (AWT), and Average Data Center Processing Time (ADCPT) [213]. The task resource mapping problem is addressed using a Crow Search-based load balancing method for enhanced optimization [214]. To fix the issue with the resource allocation model, a new optimization technique known as Grey Wolf Optimization and Crow Search Algorithm (GWO-CSA) is created [215]. The Crow Search algorithm and the Sparrow Search Algorithm (SSA) are combined to minimize energy in the cloud computing environment [216]. Mangalampalli et al. proposed a multi-objective task scheduling method using the Crow Search Algorithm to schedule tasks to the appropriate virtual machines while considering the cost per energy unit in data centers [217].

4.1.2.9 Cuckoo search algorithm Cuckoo Search is one of the most recent algorithms to draw inspiration from nature. Cuckoo Search is based on some cuckoo species' brood parasitism [218]. Additionally, the so-called Lévy flights improve this technique. The Cuckoo Search algorithm is proposed in [219, 220] to optimize task scheduling in cloud computing. A combination of two optimization algorithms, Cuckoo Search and PSO (CPSO), has been proposed in [221] paper to reduce the makespan, cost, and deadline violation rate. The Cuckoo Crow Search Algorithm (CCSA) is an effective hybridized scheduling method developed in [222] to find an appropriate VM for task scheduling. The cuckoo search-based task scheduling method suggested in [223] assists in efficiently allocating tasks among the available virtual machines and maintains a low overall response time (QoS). For better resource management in the Smart Grid, a load-balancing strategy based on the Cuckoo Search is suggested in [224]. A Standard Deviation-based Modified Cuckoo Optimization Algorithm (SDMCOA) is described to schedule tasks and manage resources [225] efficiently. The Multi-objective Cuckoo Search Optimization (MOCSO) algorithm is suggested to resolve multi-objective resource scheduling issues in an IaaS cloud computing context [226]. A combination of Cuckoo Search and oppositional-based learning (OBL) has created a new hybrid algorithm called the Oppositional Cuckoo Search Algorithm (OCSA) [227]. Madni et al. developed the Hybrid Gradient Descent Cuckoo Search (HGDCS) algorithm based on the Gradient Descent (GD) approach and Cuckoo Search algorithm to optimize and address issues with resource scheduling in IaaS cloud computing [228]. The CHSA algorithm, a mix of the Cuckoo Search and Harmony Search (HS) algorithms, is used to optimize the scheduling process [229]. A group technology-based model and Cuckoo Search algorithm are proposed for resource allocation [230]. A Hybridized Optimization algorithm that is the combination of the 'Shuffled Frog Leaping Algorithm' (SFLA) and 'Cuckoo Search' (CS) Algorithm for resource allocation is proposed in [231].

4.1.2.10 Grey wolf search optimization algorithm A meta-heuristic optimization algorithm called grey wolf optimization (GWO) was developed after studying the social interactions of grey wolves [232]. The GWO algorithm is modeled on how grey wolves hunt, which involves working together as a pack to catch prey. Natesan et al. propose simulating the performance cost grey wolf optimization (PCGWO) algorithm to optimize allocating resources and tasks in cloud computing [233]. The publication [234] offers the modified fractional grey wolf optimizer for multi-objective task scheduling (MFGMTS), a multi-objective optimization technique. Grey wolf optimizer is also proposed in [235]. A hybrid algorithm Genetic Gray Wolf optimization algorithm (GGWO), is proposed by combining gray wolf optimizer (GWO) and genetic algorithm [236]. In the study [237], a mean GWO algorithm has been developed to enhance the system performance of task scheduling in the heterogeneous cloud environment. A multi-objective GWO technique has been developed in [238] for task scheduling to achieve the best possible use of cloud resources while minimizing the data center's energy consumption and the scheduler's overall makespan for the given list of tasks. Using the hill-climbing approach and chaos theory, Mohammadzadeh et al. [239] devised IGWO, an enhanced version of the GWO algorithm that speeds up convergence and avoids getting caught in the local optimum. Particle Swarm Optimization and Grey Wolf Optimization, two well-known meta-heuristic algorithms, have been combined to form the PSO-GWO algorithm, which has been proposed in [240]. The experiment's findings indicate that, compared to the traditional Particle Swarm Optimization and Grey

Wolf Optimization techniques, the PSO-GWO methodology reduces the average total execution cost and time.

4.1.2.11 Glowworm swarm optimization

The glowworm swarm optimization (GSO) method simulates the movement of the glowworms in a swarm depending on the distance between them and on a luminous substance called luciferin [241]. It is a relatively recent swarm intelligence system. The article [242] proposes a hybrid glowworm swarm optimization (HGSO) based on GSO to achieve more effective scheduling with affordable costs. The suggested HGSO speeds up convergence and makes it easier to escape from local optima by reducing unnecessary computation and dependence on GSO initialization. GSO is used in paper [243] to address the task scheduling issue in cloud computing in order to reduce the overall cost of job execution while maintaining on-time task completion.

4.1.2.12 Symbiotic organism search optimization algorithm

The symbiotic organism search (SOS) optimization technique is a nature-inspired optimization technique that can be used for cloud computing task schedules. The algorithm is founded on the idea that organisms can coexist harmoniously and use one another's advantages to survive. The discrete symbiotic organism search (DSOS) technique is presented in the publication [244] for optimal task scheduling on cloud resources. The discrete symbiotic organism search (DSOS) method has improved, but because the makespan and response time parameters are so significant, it still gets stuck in local optima. As a result, a quicker convergent technique for enhanced Discrete Symbiotic Organism Search (eDSOS) is suggested in [245] when the search space is more extensive or more prominent due to diversification. The chaotic symbiotic organisms search (CMSOS) technique is developed to resolve the multi-objective large-scale task scheduling optimization problem in the IaaS cloud computing environment [246]. The Adaptive Benefit Factors-based Symbiotic Organisms Search (ABFSOS) method is presented to balance local and global search techniques for a faster convergence speed [247]. An energy–aware Discrete Symbiotic Organism Search (E-DSOS) Optimization algorithm has been proposed in [248] for task scheduling in a cloud environment. A modified Symbiotic Organisms Search Algorithm (G_SOS) is suggested to reduce task execution time (makespan), cost, reaction time, and degree of imbalance and speed up convergence for an ideal solution in an IaaS cloud [249].

4.1.3 Physics-based algorithm

Methods based on physics mimic the laws of physics that govern the cosmos [250]. The most widely used algorithms are Henry Gas Solubility Optimization [251], Simulated Annealing (SA) [252], Gravitational Local Search (GLSA) [253], Big-Bang Big-Crunch (BBBC) [254], Gravitational Search Algorithm (GSA) [255], Charged System Search (CSS) [256], Central Force Optimization (CFO) [257], Artificial Chemical Reaction Optimization Algorithm (ACROA) [258], Black Hole (BH) [259], Ray Optimization (RO) [260]. In this section, some physics-based nature-inspired algorithms that are used in task scheduling of cloud computing are briefly summarized.

4.1.3.1 Henry gas solubility optimization

Henry's law is a fundamental gas law that describes how much of a given gas dissolves in a specific kind and amount of liquid at a specific temperature. The Henry Gas Solubility Optimization (HGSO) algorithm mimics the huddling behavior of gas to balance exploitation and exploration in the search space and prevent local optima. A modified Henry gas solubility optimization for the best task scheduling is provided in the paper [261] and is based on the WOA and Complete Opposition-Based Learning (COBL). Henry Gas Solubility Whale Cloud (HGSWC) is the name of the proposed technique. A set of 36 optimization benchmark functions is used to validate HGSWC, compared to traditional HGSO and WOA.

4.1.3.2 Simulated annealing

Simulated Annealing (SA) is a technique for resolving bound- and unconstrained optimization issues. The technique simulates the physical procedure of raising a material's temperature and gradually decreasing it to reduce flaws while conserving system energy. A combination of Firefly and SA has been presented in [206].

4.1.4 Hybrid algorithm

In hybrid scheduling algorithms, two or more scheduling algorithms are merged to address the task scheduling issue in a cloud context. The fundamental concept behind hybrid algorithms is to combine the benefits of various methods into a single algorithm to improve performance in terms of computation time, result in quality, or both. [262] discusses combining ACO and PSO techniques to improve resource scheduling. The performance and quality of the solution are optimized using a similar hybrid technique by the authors in [263] employing ACO and Intelligent Water Drop algorithm. ACO with Particle Swarm (ACOPS) [264] is presented to schedule VMs more efficiently. In this method, the burden of user requests is dynamically forecasted and mapped onto the available VMs. In [265], authors introduced an algorithm that optimizes work scheduling in cloud environments using fundamental notions from ACO and ABC. A hybrid Gravitational

Emulation Local Search approach and PSO is used in [152] to enhance the results. In [148], the hill climbing local search heuristic is integrated with PSO, and in [266], fuzzy logic and GSO are described.

4.2 Nature of scheduling problem

Developing an optimization model that satisfies the objectives by locating the best optimal solution is necessary because there is always an agreement between optimization objectives. Therefore, two types of scheduling objectives are considered for nature-inspired task scheduling. Table 2 represents the categorization of techniques reviewed based on the nature of the scheduling problem.

4.2.1 Single objective

It is feasible to evaluate the optimality of a specific solution in contrast to another one already existing in a single objective optimization. For predetermined objectives, a single best solution is chosen. Regarding task scheduling in cloud computing, most approaches only consider the CPU and memory requirements.

4.2.2 Multi-objective

It is possible to characterize task scheduling in a distributed heterogeneous computing system as a non-linear, multi-objective, and NP-hard optimization problem that aims to maximize cloud resource consumption while meeting QoS standards. It is impossible to directly compare one solution's optimality to another already existing in multi-objective optimization. Multi-objective scheduling often uses a Pareto dominance relation technique to replace a single optimal solution with various possibilities and provide numerous, varied trade-offs between the objectives.

Table 2 Nature of scheduling problem

	References	Nature-inspired algorithm
Single objective	[267]	FA, CSA, and SA
Multi-objective	[268]	PSO
	[269]	PSO
	[270]	ACO
	[271]	PSO
	[226]	CSA

4.3 Nature of task

There are two scheduling techniques: independent and dependent scheduling (workflow scheduling). When dependent scheduling is used, a workflow connects the tasks to one another. The tasks are autonomous in independent scheduling because they are not dependent on one another. Several authors used both scheduling in their proposed system. Table 3 depicts the categorization of techniques reviewed based on the nature of task.

4.4 Primary objectives of scheduling

Based on specific scheduling criteria, the scheduling process distributes the tasks inside the workflow onto the appropriate resources. The scheduling criteria, such as execution time, cost, reliability, and load balancing influence the success of the scheduling challenge. Tables 4, 5, 6, and 7 illustrate the categorization of techniques based on primary objectives of scheduling using the evolutionary algorithm, swarm intelligence-based algorithm, physics-base algorithm, and hybrid algorithm, respectively.

4.4.1 Makespan

Makespan is described as the overall time required to execute the whole workflow by considering the time when the tasks finished their execution and the time when it has been submitted [290]. In the literature, most of the scheduling algorithms have focused on optimizing makespan [79, 291]. However, minimizing the total execution time reduces the execution cost while mapping the tasks to the resources.

4.4.2 Security

Due to the heterogeneous and scattered nature of cloud computing resources, security is a significant concern. Because of virtualization and multi-tenancy capabilities, providing data security and privacy in a cloud environment is more challenging than in traditional systems.

4.4.3 Reliability

Failures during the execution of a workflow can occur for various reasons, including resource unavailability, resource failure, and network infrastructure. As a result, the scheduling mechanism should consider resource failure and ensure reliable executions even when there is concurrency and failure. The likelihood that the tasks will be carried out successfully and the workflow will be completed is known as reliability. For an application to run smoothly, all

Table 3 Nature of task

	Reference	Algorithm
Independent	[267]	FA, CSA, and SA
	[268]	PSO
	[269]	PSO
	[271]	PSO
	[226]	CSA
	[272]	Hybrid (ACO and CSA)
	[273]	PSO
	[274]	CSO
	[275]	GA
	[276]	chaotic ACO and GA
	[277]	GA
	[278]	FA
	[279]	ABC and PSO
Workflow	[280]	GA
	[281]	ACO
	[282]	ACO, PSO, and GA
	[283]	Hybrid (SA with CSO)
	[284]	PSO
	[285]	PSO
	[286]	GA
Independent and workflow scheduling	[270]	ACO
	[287]	GA
	[288]	IE-ABC
	[289]	PSO and CSO

resources must be reliable. It is possible to calculate the failure rate; thus, the mapping should be carried out to increase reliability and lower failure rates. Task-level, VM-level, and workflow-level failures are the three levels of failure in a workflow application.

4.4.4 Cost

The cost of an application is determined by two factors depending on two fundamental cloud resources: the cost of computing and the cost of data transit and storage. By supplying a lot of resources, the overall execution time can be reduced, but execution costs, scheduling overheads, and resource underutilization may all rise as a result.

4.4.5 Load balancing

In cloud computing settings, virtual computers predominate as the processing components. There may be instances during scheduling where multiple tasks are allocated to virtual machines (VMs) for simultaneous execution. The loads on the VMs become imbalanced as a result. The scheduler should be able to spread the burden to the available ones to prevent overloading resources. Load balancing over the resource enhances resource consumption and the efficiency of the scheduling process.

4.4.6 Resource utilization

The service provider benefits from higher resource utilization to maximize profit by renting out scarce resources to users to ensure full utilization.

4.4.7 Rescheduling

Because rescheduling necessitates re-evaluating the plan and the cost of data transportation among the dependent activities over the multiple machines, it is generally considered an overhead to the scheduling process [292]. A heavy load on the server can need rescheduling. Rescheduling the jobs is also necessary in the event of failures such as VM shutdowns or system faults [293]. Only some tasks are chosen for rescheduling because it lengthens execution time overall and causes performance to suffer.

Table 4 Primary objectives of scheduling for the evolutionary algorithm

Algorithm	Reference	Reliability	Makespan	Cost	Security	Load Balance	Energy Efficient	Rescheduling	Resource Utilization
GA	[297]	Yes	Yes	Yes	–	–	–	–	–
	[79]	–	Yes	Yes	Yes	–	–	–	–
	[298]	–	Yes	–	–	–	–	–	–
	[75]	–	Yes	–	–	Yes	–	–	–
	[76]	–	Yes	-	Yes	–	–	–	–
	[294]	–	–	–	–	–	Yes	–	Yes
	[68]	–	Yes	–	–	–	–	–	Yes
	[69]	–	–	-	–	Yes	–	–	–
	[70]	–	–	–	–	–	–	–	Yes
	[71]	–	Yes	–	–	–	–	–	-
	[72]	–	–	–	–	–	–	–	Yes
MA	[88]		Yes	Yes	–	–	–	–	–
	[90]		Yes	Yes	–	–	–	–	–
	[91]	–	Yes	–	–	–	–	–	–
ICA	[99]	Yes	–	–	–	–	–	–	–
	[103]	–	Yes	–	–	–	–	–	–
	[104]	Yes	–	–	–	–	Yes	–	–
LOA	[299]	–	Yes	Yes	–	Yes	–	–	Yes
SFO	[107]	–	Yes	–	–	–	Yes	–	
	[108]	–	–	–	–	–	–	–	Yes

4.4.8 Energy efficiency

The amount of CPU and resources used directly influences how much energy a task uses. When CPUs are not appropriately utilized, idle power is not utilized effectively, resulting in significant energy consumption. Due to the high resource demand, it might occasionally consume much energy, reducing performance [294]. Scheduling decisions are crucial, to limit the energy consumption of the allocated resource. They help determine the best order in which tasks should be completed. An energy-efficient storage service is one of the strategies for reducing power usage addressed in [295]. This service may create a predictive model to forecast how users will use the files and data stored there. A few scheduling algorithms might even be created to lower energy usage [296].

4.5 Scheduling constraints

The possibility of the SLA being negatively impacted if many applications cannot meet the deadline, priority, budget, and fault tolerance limits are essential concerns in cloud scheduling. The suggested mechanisms in this respect are detailed in the subsections that follow. Table 8 illustrates the categorization of techniques based on scheduling constraints.

4.5.1 Budget

Budget is a limitation the user imposes on using the cloud service provider's resources. Scheduling decisions are made utilizing the budget constraint to minimize the workflow's overall execution time and guarantee that it is completed within the budget.

4.5.2 Deadline

Applications that depend on timing must finish running in a specific amount of time. These applications are made to give results before the deadline using deadline-constrained scheduling. When scheduling jobs, deadline-constrained scheduling must also consider the associated costs. In time-sensitive applications, robust scheduling with deadlines is essential since it increases the dependability of the program.

4.5.3 Fault tolerant

A cost-effective fault tolerant (CEFT) scheduling strategy that should adhere to a predetermined deadline in cloud systems was discussed in the paper [257]. This method of handling jobs, known as the primary/backup (P/B) strategy, contains two duplicate copies of each task and offers

Table 5 Primary objectives of scheduling for swarm intelligence-based algorithm

	Reference	Reliability	Makespan	Cost	Security	Load Balance	Energy Efficient	Rescheduling	Resource Utilization
ACO	[300]	–	Yes	–	–	Yes	–	–	–
	[301]	–	Yes	–	–	–	–	–	–
	[133]	–	Yes	Yes	–	–	–	–	–
	[134]	–	Yes	Yes	–	–	–	–	–
	[138]	–	Yes	–	–	–	–	–	–
PSO	[302]	–	Yes	–	Yes	–	–	–	–
	[295]	–	Yes	Yes	Yes	Yes	–	–	–
	[146]	–	Yes	Yes	–	–	–	–	–
	[147]	–	–	Yes	–	–	–	–	–
ABC	[168]	–	Yes	–	–	–	–	–	–
	[169]	–	Yes	–	–	–	–	–	–
	[170]	–	–	–	–	Yes	Yes	–	Yes
	[171]	–	–	–	–	Yes	–	–	–
	[172]	–	–	–	–	Yes	–	–	Yes
BA	[181]	–	Yes	Yes	–	–	–	–	–
	[182]	–	Yes	-	–	–	–	–	–
	[184]	–	Yes	Yes	–	–	–	–	–
CSO	[191]	–	–	Yes	–	Yes	–	–	–
	[192]	–	–	Yes	–	–	–	–	–
WOA	[198]	–	Yes	Yes	–	–	–	–	–
	[196]	–	Yes	Yes	–	–	–	–	–
	[199]	–	Yes	Yes	–	–	Yes	–	Yes
	[195]	–	–	Yes	–	–	Yes	–	–
	[194]	–	Yes	Yes	–	–	–	–	–
FA	[201]	–	Yes	Yes	–	–	–	–	–
	[207]	–	Yes	Yes	–	Yes	–	–	–
	[204]	–	Yes	–	–	–	–	–	–
Crow Search	[217]	–	Yes	–	–	–	Yes	–	
	[212]	–	Yes	Yes	–	Yes	–	–	Yes
	[214]	–	–	Yes	–	–	Yes	–	–
	[210]	–	Yes	–	–	–	–	–	–
	[213]	–	–	–	–	–	–	–	Yes
Cuckoo Search	[223]	–	–	–	–	Yes	–	–	–
	[224]	–	Yes	–	–	–	–	–	–
	[225]	–	Yes	Yes	–	Yes	–	–	–
	[226]	–	Yes	Yes	–	–	–	–	–
	[227]	–	Yes	–	–	–	–	–	–
	[228]	–	Yes	Yes	–	–	–	–	–
	[229]	–	Yes	–	–	Yes	–	–	–
GWO	[236]	–	Yes	Yes	–	Yes	Yes	–	–
	[241]	–	Yes	Yes	–	–	–	–	–
	[240]	–	Yes	Yes	–	–	–	–	–
	[239]	–	Yes	Yes	–	–	Yes	–	–
GSO	[242]	–	Yes	Yes	–	–	–	–	–
SOS	[244]	–	Yes	–	–	–	–	–	–
	[245]	–	Yes	–	–	–	–	–	–
	[246]	–	–	–	–	–	Yes	–	–

Table 6 Primary objectives of scheduling for physics-based algorithm

	Reference	Reliability	Makespan	Cost	Security	Load Balance	Energy Efficient	Rescheduling	Resource Utilization
HGSA	[261]	–	Yes	Yes	–	–	–	–	–
SA	[206]	–	Yes	–	–	–	–	–	–

Table 7 Primary objectives of scheduling for hybrid algorithm

		Reliability	Makespan	Cost	Security	Load balance	Energy efficient	Rescheduling	Resource utilization
ABC and PSO	[279]	–	Yes	–	–	–	Yes	–	–
ACO with CRO	[281]	–	Yes	Yes	–	–	–	–	–
ACO and CSO	[282]	–	Yes	–		–	–	–	–
ACO and GA	[272]	–	Yes	–		–	–	–	–
ACO with GSA	[277]	–	Yes	–	Yes	–	-	–	Yes
CSA with HGSA	[283]	–	Yes	Yes	–	–	Yes	–	–
GA with ANN	[287]	–	Yes	–	–	–	Yes	–	–
GA with fuzzy	[303]	–	Yes	–	–	–	–	–	–

Table 8 Scheduling constraints

	References	Nature-inspired algorithm
Deadline	[304]	PSO
	[282]	ACO
	[289]	PSO and CSO
Priority	[305]	PSO
	[306]	BFO
Budget	[307]	PSO
Fault tolerance	[308]	PSO

permanent or temporary fault tolerance in the event of hardware failure. The tasks in this strategy are independent and do not follow a hierarchy. The iterative method is used to optimize the suggested resource allocation technique more successfully. The PSO method chooses VM for the next task in each iteration.

4.5.4 Priority

Verma and Kaushal suggested a Bi-Criteria priority-based PSO (BPSO), lowering the makespan and execution costs while scheduling cloud workflow activities [305]. This PSO would take into consideration deadlines and financial restrictions. Each workflow task is prioritized at the bottom level specified by the HEFT algorithm so that the PSO can carry out the tasks according to these priorities.

Table 9 Task resource mapping scheme

	Reference	Nature-inspired algorithm
Static	[171]	ABC
Dynamic	[309]	ACO and PSO
	[312]	GA
	[313]	PSO
	[314]	GA
AI	[310]	PSO
	[315]	PSO
Prediction	[311]	ACO
	[285]	GA

4.6 Task-resource mapping scheme

To effectively utilize the available resources based on the cloud environment and the submitted workload, static, dynamic, AI-based, and prediction-based mapping of cloud resources to incoming tasks are carried out. Table 9 illustrates the categorization of techniques based on the task resource mapping scheme.

4.6.1 Static

Static scheduling involves prior knowledge of the tasks to decide on a schedule before a task begins to execute. By enhancing load balancing among VMs and balancing the priority of the activities on those VMs, the paper [171] suggested a resource provisioning technique inspired by the ABC algorithm's behavior to boost resource utilization, increase system throughput, and decrease queuing time.

4.6.2 Dynamic

Dynamic scheduling may occur while a task is being executed and doesn't need to be aware of every task property. When maximizing resource use is more critical than reducing execution time, this helps manage changing requirements of cloud users [219]. All meta-heuristic scheduling techniques are, in fact, dynamic [220]. However, some research is discussed in this subsection based on the dynamic cloud environment and the dynamic scheduling system specified in the primary keyword of the selected articles. Islam and Habiba have proposed dynamic scheduling methods based on ACO and Variable Neighborhood PSO (VNPSO) [309].

4.6.3 Artificial intelligence (AI)-based

The creation of an intelligent method that works and responds, like humans, to schedule and assign resources with various aspects, such as intelligent and autonomous systems, nature-inspired intelligent systems, operational research systems, agent-based systems, neural networks, machine learning, and expert systems, is supported by the highly technical and specialized methodology known as AI-based scheduling [226]. Greater accuracy and precision are ensured for resource allocation and scheduling in the cloud framework with AI, and failure and error rates are nearly nonexistent. ML algorithms have been used in cloud computing to forecast the status of the resources based on their anticipated future load and security. Incoming requests were mapped to resources using an ANN model by paper [310]; to complete the task more quickly.

4.6.4 Prediction-based

Prediction-based scheduling relates to how various techniques and measurements behave when allocating resources. When it comes to efficient task scheduling and optimum resource allocation in the cloud environment, it can be essential to estimate the critical resource requirements and users' demand for the future using automatic resource allocation or resource reservation approaches [231, 232]. Paper [311] introduced a Prediction-based ACO classification algorithm (PACO), which operates based on task prioritization and considers various QoS criteria to minimize the workflow's overall execution time and guarantee that it is completed within the budget.

4.7 Testing tools and their comparison

Testing new strategies in an authentic setting is almost impossible because some trials could harm the end-user QoS. There are numerous well-known simulation tools for examining novel scheduling algorithms and judging their efficiency in various cloud environments. The most well-known tool for task scheduling is the CloudSim simulation toolkit. Its existing programmatic classes can be expanded by the algorithm needed to assess a variety of QoS parameters, including makespan, financial cost, computational cost, reliability, availability, scalability, energy consumption, security, and throughput, as well as includible constraints, such as deadline, priority, budget, and fault tolerance. iCloud, GridSim, CloudSim, CloudAnalyst, NetworkCloudSim, Work flowSim, GreenCloud, and more well-known simulation tools are available on the cloud. Table 10 illustrates the categorization of techniques based on testing tools.

5 Analysis and discussions

This section comprehensively summarizes and examines the nature-inspired meta-heuristics techniques for task scheduling in cloud computing in terms of the scheduling algorithm, type of tasks, the primary objective of scheduling, task-resource mapping scheme, scheduling constraint, and testing environment.

5.1 Scheduling algorithm

Figure 3 shows how various task scheduling methods for clouds inspired by nature can be divided into swarm, evolutionary, physics, and hybrid approaches, along with the proportions of each most commonly used approach. The majority of the algorithms used in the literature are swarm-based optimization methods. The most active algorithm among them is PSO, as seen in Fig. 4. Sometimes, those algorithms are used with additional meta-heuristic or heuristic techniques. Because a hybrid version of the system can further improve overall performance, the search strategy and convergence speed of meta-heuristic techniques vary. Therefore, the hybrid algorithm achieves the second rank in the list. Most researchers used the GA in evolutionary approaches, as seen in Fig. 5.

5.2 Scheduling objective

Multi-objective techniques have more scheduling strategies than single-objective, as shown in Fig. 6. Consequently, it encourages the ongoing incorporation of multi-objective strategies for more dependable task scheduling in the cloud environment.

Table 10 Testing environment

	Reference	Nature-inspired algorithm
GCC compiler	[267]	FA, CSA, and SA
CloudSim	[268]	PSO
	[269]	PSO and GA
	[270]	PSO
	[226]	CSA
	[273]	PSO
	[274]	PSO
Real environment	[272]	ACO and CSA
	[280]	GA
CloudSim and real environment	[270]	ACO
	[285]	PSO
Matlab	[288]	PSO
	[276]	chaotic ACO and GA
	[277]	GA
CloudSim with MATLAB	[287]	GA
GridSim	[312]	GA
Java	[282]	ACO
	[310]	PSO
	[307]	PSO

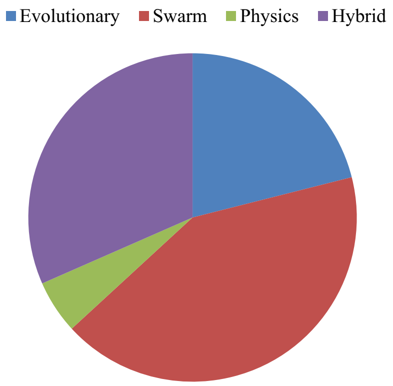

Fig. 3 Usage measurement of nature-inspired algorithm

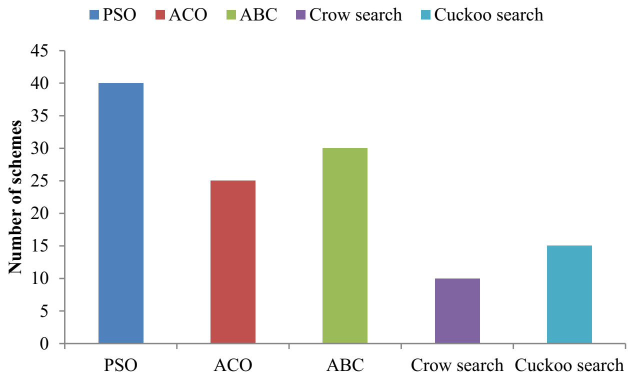

Fig. 4 Usage measurement of swarm algorithm

5.3 Type of task

The most used type of task scheduling is independent rather than workflow scheduling. Some literature reviews also use the combination of independent and workflow scheduling. The usage measurement of the type of task scheduling for nature-inspired algorithms is illustrated in Fig. 7.

5.4 Primary objectives of scheduling

The algorithms are examined in light of the different scheduling objectives in Fig. 8. The many QoS parameters that different scheduling algorithms take into consideration are shown in Fig. 8. The makespan is the most often employed scheduling objective. The second most crucial element is cost. Only a few studies, as shown in Fig. 8, have utilized reliability into consideration while scheduling tasks. Reliable scheduling can lessen the impact of resource breakdowns. Making scheduling decisions using the application's failure probability as a minimum consideration is recommended. The service providers must consider resource consumption. Proper resource usage will be achievable if the submitted applications have the necessary resources. However, there are connections between the use of resources and energy consumption. The fundamental concept behind enhancing resource usage is to combine the load on the virtual machines (VMs) such that the spare VMs can be turned off or repurposed for new application demands. In this situation, predicting future resource demand can help cloud service providers turn a profit.

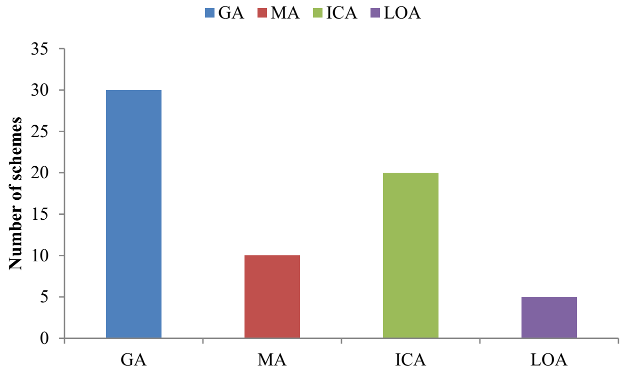

Fig. 5 Usage measurement of evolutionary algorithm

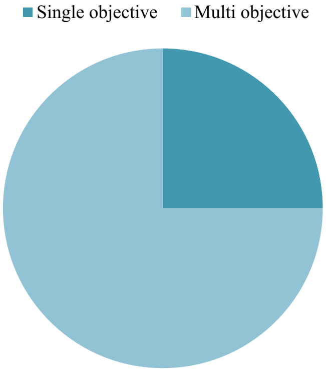

Fig. 6 Usage measurement of scheduling objective algorithm

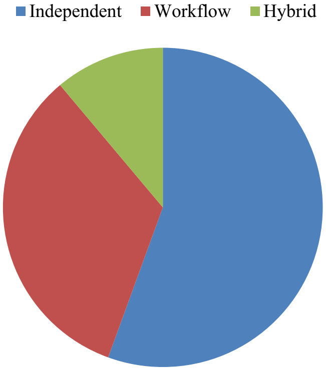

Fig. 7 Usage measurements of types of task

Reducing VM migrations might be made more accessible with the estimation.

5.5 Scheduling constraints

Deadline is the constraint most emphasized by nature-inspired techniques. Fault tolerance is also a widely adopted constraint. Most studies also concentrated on budget and priority at the time of scheduling.

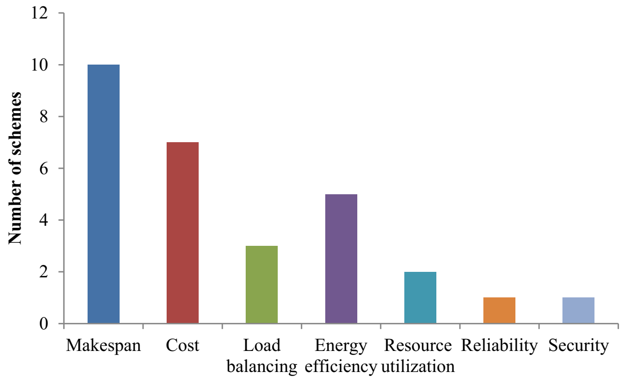

Fig. 8 Frequency of scheduling objectives

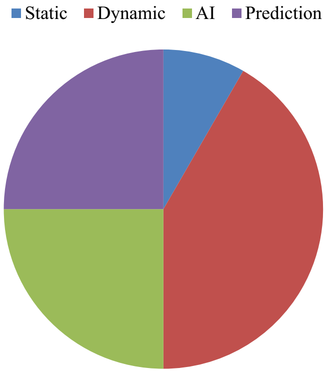

Fig. 9 Usage measurements of task resource management

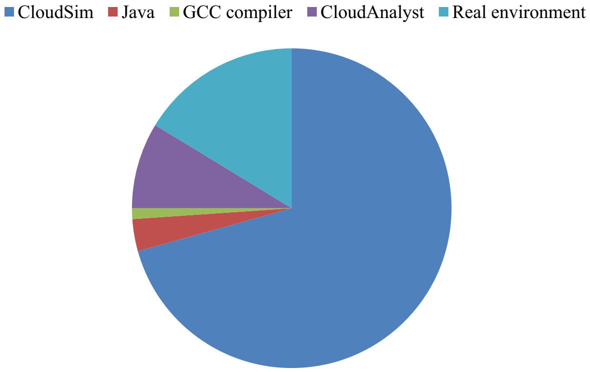

Fig. 10 Usage measurements of the testing environment

5.6 Task resource mapping scheme

Dynamic scheduling has the most excellent use in nature-inspired optimization problems. AI and prediction-based scheduling are also more widely used than static scheduling, as depicted in Fig. 9.

5.7 Testing environment

CloudSim is the most popular and widely used testing tool in implementing nature-inspired task scheduling algorithms, as depicted in Fig. 10. A few studies also used GridSim, java, Matlab, etc. in some works.

5.8 Discussion

In this study, a novel taxonomy has been used to comprehensively and meticulously categorize different popular nature-inspired scheduling approaches in the cloud in terms of the scheduling algorithm, the nature of the scheduling problem, the type of task, the main scheduling goal, the task-resource mapping scheme, the scheduling constraint, and the testing environment.

Swarm-based optimization techniques are used in most of the scheduling algorithms used in the literature. The most popular swarm-based technique for scheduling issues in clouds is Particle Swarm Optimization. The majority of researchers use Genetic Algorithms in evolutionary-based techniques as well. These algorithms are occasionally integrated with different heuristics or meta-heuristic methods. Most of the studies are multi-objective and independent. In maximal meta-heuristic algorithms, makespan is employed as the scheduling objective. As scheduling objectives, cost and energy efficiency are also given preference. Security and reliability parameters receive less focus. The deadline is the constraint that nature-inspired techniques highlight most. Additionally, fault tolerance is a widely used restriction. Dynamic scheduling achieves higher popularity than static, AI, and prediction-based approaches. The most widely used task scheduling tool is the CloudSim simulation toolkit, whose existing programmatic classes can be expanded in accordance with the algorithm requirements to evaluate a variety of QoS parameters, including makespan, monetary cost, computational cost, reliability, availability, scalability, energy consumption, security, and throughput.

6 Future research issues of nature-optimization task scheduling algorithms

Although nature-inspired algorithms have been used in many applications successfully, there is still plenty of space for analysis and discussion regarding many current problems and topics. Several areas need to be addressed in this field. Cloud task scheduling presents a variety of difficulties, including heterogeneity, uncertainty, and resource dispersion, which conventional resource management techniques need to address. Therefore, to increase the dependability of cloud applications and services, a lot of attention and significance should be given to these cloud qualities. The workload should occupy the fewest resources that will be effectively employed to achieve the shortest job duration (maximize system throughput) and maintain a desired QoS level. Developing novel practical solutions in this field still requires additional research effort. There must be a minimum amount of contact between the computing environment and the cloud consumer to fulfill the QoS standards specified by users while maintaining the SLA. Performance degradation could therefore be prevented through research into an efficient autonomic infrastructure-based technique that, in advance, detects SLA violations.

Increasing energy efficiency in cloud computing is one of the main issues. According to estimates, 53% of all operational expenses go into cooling and powering data centers. As a result, IaaS providers have been given the urgent responsibility of reducing energy use. The goal of data center design should not only be to save energy costs; it also needs to comply with environmental regulations and legal requirements. Energy-aware server consolidation and energy-efficient task scheduling can be used to lower power usage by turning off idle systems. Since cloud service providers today seek to offer their services to end customers with high performance, high quality, and minimal processing time while making maximum profit, reliability is one of the most demanding cloud computing concerns. The scheduling strategy should be used to safeguard and protect the private and sensitive data contained in the submitted applications.

As can be observed from the numerous state-of-the-art scheduling methods chosen for this study, not all problems (challenges) can be solved by a single algorithm. For instance, while an algorithm considers different factors, such as resource utilization, availability, response time, scalability, etc., it may entirely neglect the energy, cost, time, and quality of service (QoS) parameters that are the emphasis of the other algorithm. Hybrid algorithms can be more inventively modified to individually or collectively maximize a variety of scheduling goals, such as energy optimization, load balancing, scalable VM migration, etc.

7 Conclusion

Many scientific applications can now migrate to the cloud because of the adoption of the new distributed computing paradigm and cloud computing infrastructure. With the advantages of cloud computing, including virtualization and shared resource pools, it is possible to manage and run large-scale workflow applications without maintaining physical computing infrastructure. The use of numerous resources accessible via the cloud needs to be optimized. Nature-inspired scheduling algorithms give better optimization results than traditional and heuristic. Over the past few years, nature-inspired optimization has developed rapidly, establishing a significant trend in cloud task scheduling. Several existing studies have employed nature-inspired optimization algorithms to solve scheduling problems. However, their studies do not concentrate on all of the parameters required to analyze the importance of nature-inspired optimization algorithms in task scheduling. Most studies only considered a small number of parameters (such as state-of-the-art, QoS parameters, etc.). Still, they need to simultaneously assess taxonomy, graphical representation, and research difficulties for a deeper understanding of task scheduling techniques based on nature-inspired optimization. To the best of our knowledge, a thorough, systematic, taxonomic assessment of nature-inspired optimization scheduling approaches in the cloud is required to keep up with the ever-increasing growth of nature-inspired algorithms. Therefore, a comprehensive study of several nature-inspired meta-heuristics-based strategies has been conducted in this paper. Conventional and heuristic scheduling approaches were presented to distinguish them from nature-inspired optimization algorithms. A novel taxonomy has been demonstrated to extensively and methodically classify various popular nature-inspired scheduling approaches in the cloud regarding the scheduling algorithm, the nature of the scheduling problem, the type of task, the primary scheduling goal, the task-resource mapping scheme, the scheduling constraint, and the testing environment. The results of each classification stage of taxonomy are analyzed and comprehensively examined to provide a roadmap for researchers working in the area of scheduling for enhancing cloud service.

In most literature, researchers have employed PSO and GA algorithms to carry out most cloud computing tasks. The most active study areas are makespan, cost, and resource utilization which are concentrated by most studies. Single objective scheduling, reliability, rescheduling, budget, priority, AI-based, and prediction-based mapping schemes have attracted less attention from researchers; as a result, they must be urgently included in ongoing research. The solutions produced by many nature-inspired methods that are already in use can be improved by considering the hybrid approach. The advantages of existing nature-inspired algorithms can be combined in a hybrid manner to address various issues brought on by workflow applications that optimize across several criteria. At the end of the discussion, a few simulation tools used in cloud computing are briefly addressed and contrasted for implementing and testing new algorithms. This study also presented some outstanding research issues and upcoming modern trends. This study primarily endeavored to strongly encourage the depth of the fundamental ideas of nature-inspired task scheduling techniques in the cloud computing area, which should give investigators and users advice on how to identify the nature of their scheduling problem, pinpoint their primary QoS parameters, identify the best task-resource mapping scheme, and establish the scheduling constraints that are most suitable for the challenge, without violating the SLA.

Author contribution FSP: writing original draft, literature surveys, writing—review and editing; MdHG: draft manuscript preparation, writing—review, and editing; KMAU: study conception, supervision, and investigation on challenges.

Funding The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data Availability Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest The authors have no conflict of interest to disclose.

References

1. Kaur, R., Laxmi, V.: Performance evaluation of task scheduling algorithms in virtual cloud environment to minimize makespan. Int. J. Inf. Technol. (2022). https://doi.org/10.1007/s41870-021-00753-4
2. Gawali, M.B., Shinde, S.K.: Task scheduling and resource allocation in cloud computing using a heuristic approach. J. Cloud Comput. **7**(1), 1–16 (2018)
3. Singh, S., Chana, I.: A survey on resource scheduling in cloud computing: issues and challenges. J. Grid Comput. **14**, 217–264 (2016)
4. Mathew, T., Sekaran, K.C. and Jose, J., 2014, September. Study and analysis of various task scheduling algorithms in the cloud computing environment. In 2014 International Conference on Advances in Computing, Communications and Informatics *(ICACCI)* (pp. 658–664). IEEE.
5. Xu, L., Qiao, J., Lin, S., Zhang, W.: Dynamic task scheduling algorithm with deadline constraint in heterogeneous volunteer computing platforms. Future Internet **11**(6), 121 (2019)
6. Damodaran, P., Chang, P.Y.: Heuristics to minimize makespan of parallel batch processing machines. Int. J. Adv. Manuf. Technol. **37**, 1005–1013 (2008)
7. Kim, S.I., Kim, J.K.: A method to construct task scheduling algorithms for heterogeneous multi-core systems. IEEE Access **7**, 142640–142651 (2019)
8. Pinedo, M. and Hadavi, K., 1992. Scheduling: theory, algorithms and systems development. In Operations Research Proceedings 1991: Papers of the 20th Annual Meeting/Vorträge der 20. Jahrestagung (pp. 35–42). Springer, Berlin
9. 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)
10 Singh, H., Tyagi, S., Kumar, P.: Scheduling in cloud computing environment using metaheuristic techniques: a survey. In: Shal, V. (ed.) Emerging technology in modelling and graphics: proceedings of IEM graph 2018, pp. 753–763. Springer Singapore, Singapore (2020)

11. Liu, Y., Zhang, C., Li, B., Niu, J.: DeMS: A hybrid scheme of task scheduling and load balancing in computing clusters. J. Netw. Comput. Appl. **83**, 213–220 (2017)
12. Kumar, D.: Review on task scheduling in ubiquitous clouds. J. ISMAC **1**(01), 72–80 (2019)
13. Allahverdi, A., Ng, C.T., Cheng, T.E., Kovalyov, M.Y.: A survey of scheduling problems with setup times or costs. Eur. J. Oper. Res. **187**(3), 985–1032 (2008)
14. Remesh Babu, K.R. and Samuel, P., 2016. Enhanced bee colony algorithm for efficient load balancing and scheduling in cloud. In *Innovations in Bio-Inspired Computing and Applications: Proceedings of the 6th International Conference on Innovations in Bio-Inspired Computing and Applications (IBICA 2015) held in Kochi, India during December 16–18, 2015* (pp. 67–78). Springer International Publishing.
15. Taillard, E.: Some efficient heuristic methods for the flow shop sequencing problem. Eur. J. Oper. Res. **47**(1), 65–74 (1990)
16. Morton, T., Pentico, D.W.: Heuristic scheduling systems: with applications to production systems and project management. John Wiley, Hoboken (1993)
17. Bissoli, D.C., Altoe, W.A., Mauri, G.R. and Amaral, A.R., 2018, August. A simulated annealing metaheuristic for the bi-objective flexible job shop scheduling problem. In *2018 International Conference on Research in Intelligent and Computing in Engineering (RICE)* (pp. 1–6). IEEE.
18. Gong, G., Chiong, R., Deng, Q., Gong, X.: A hybrid artificial bee colony algorithm for flexible job shop scheduling with worker flexibility. Int. J. Prod. Res. **58**(14), 4406–4420 (2020)
19. Zarrouk, R., Bennour, I.E., Jemai, A.: A two-level particle swarm optimization algorithm for the flexible job shop scheduling problem. Swarm Intell. **13**, 145–168 (2019)
20. Sörensen, K., Glover, F.: Metaheuristics. Encycl. Operations Res. Manag. Sci. **62**, 960–970 (2013)
21. Garg, D. and Kumar, P., 2019. A survey on metaheuristic approaches and its evaluation for load balancing in cloud computing. In *Advanced Informatics for Computing Research: Second International Conference, ICAICR 2018, Shimla, India, July 14–15, 2018, Revised Selected Papers, Part I 2* (pp. 585–599). Springer Singapore.
22. Kaur, N. and Chhabra, A., 2016, March. Analytical review of three latest nature inspired algorithms for scheduling in clouds. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)* (pp. 3296–3300). IEEE.
23. Garg, D. and Kumar, P., 2019. A survey on metaheuristic approaches and its evaluation for load balancing in cloud computing. In Advanced Informatics for Computing Research: Second International Conference, ICAICR 2018, Shimla, India, July 14–15, 2018, Revised Selected Papers, Part I 2 (pp. 585–599). Springer Singapore.
24. Kalra, M., Singh, S.: A review of metaheuristic scheduling techniques in cloud computing. Egypt. Inf. J. **16**(3), 275–295 (2015)
25. Kaur, N. and Chhabra, A., 2016, March. Analytical review of three latest nature inspired algorithms for scheduling in clouds. In 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) (pp. 3296–3300). IEEE.
26. Tsai, C.W., Rodrigues, J.J.: Metaheuristic scheduling for cloud: a survey. IEEE Syst. J. **8**(1), 279–291 (2013)
27. Nandhakumar, C. and Ranjithprabhu, K., 2015, January. Heuristic and meta-heuristic workflow scheduling algorithms in multi-cloud environments—a survey. In 2015 International Conference on Advanced Computing and Communication Systems (pp. 1–5). IEEE.
28. Hatchuel, A., Saidi-Kabeche, D., Sardas, J.C.: Towards a new planning and scheduling approach for multistage production systems. Int. J. Prod. Res. **35**(3), 867–886 (1997)
29. Lawler, E.L., Lenstra, J.K. and Rinnooy Kan, A.H.G., 1982. Recent developments in deterministic sequencing and scheduling: a survey. In *Deterministic and Stochastic Scheduling: Proceedings of the NATO Advanced Study and Research Institute on Theoretical Approaches to Scheduling Problems held in Durham, England, July 6–17, 1981* (pp. 35–73). Springer Netherlands.
30. Madni, S.H.H., Abd Latiff, M.S., Abdullahi, M., Abdulhamid, S.I.M., Usman, M.J.: Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment. PLoS ONE **12**(5), e0176321 (2017)
31. Mazumder, A.M.R., Uddin, K.A., Arbe, N., Jahan, L. and Whaiduzzaman, M., 2019, June. Dynamic task scheduling algorithms in cloud computing. In *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)* (pp. 1280–1286). IEEE.
32. Chowdhury, N., M Aslam Uddin, K., Afrin, S., Adhikary, A., Rabbi, F.: Performance evaluation of various scheduling algorithm based on cloud computing system. Asian J. Res. Comput. Sci. **2**(1), 1–6 (2018)
33. Balharith, T. and Alhaidari, F., 2019, May. Round robin scheduling algorithm in CPU and cloud computing: a review. In *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)* (pp. 1–7). IEEE.
34. Zhao, H. and Sakellariou, R., 2003. An experimental investigation into the rank function of the heterogeneous earliest finish time scheduling algorithm. In *Euro-Par 2003 Parallel Processing: 9th International Euro-Par Conference Klagenfurt, Austria, August 26-29, 2003 Proceedings 9* (pp. 189-194). Springer, Berlin
35. Li, B., Niu, L., Huang, X., Wu, H. and Pei, Y., 2018, December. Minimum completion time offloading algorithm for mobile edge computing. In *2018 IEEE 4th International Conference on Computer and Communications (ICCC)* (pp. 1929–1933). IEEE.
36. Krishnaveni, H., Sinthujanitaprakash, V.: Execution time based sufferage algorithm for static task scheduling in cloud. In: Advances in big data and cloud computing: Proceedings of ICBDCC18, pp. 61–70. Springer Singapore, Singapore (2019)
37. Chen, H., Wang, F., Helian, N. and Akanmu, G., 2013, February. User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing. In *2013 national conference on parallel computing technologies (PARCOMPTECH)* (pp. 1–8). IEEE.
38. George Amalarethinam, D.I. and Kavitha, S., 2019. Rescheduling enhanced Min-Min (REMM) algorithm for meta-task scheduling in cloud computing. In *International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018* (pp. 895–902). Springer International Publishing.
39. Mao, Y., Chen, X. and Li, X., 2014. Max–min task scheduling algorithm for load balance in cloud computing. In *Proceedings of International Conference on Computer Science and Information Technology: CSAIT 2013, September 21–23, 2013, Kunming, China* (pp. 457–465). Springer India.
40. Sandana Karuppan, A., Meena Kumari, S.A. and Sruthi, S., 2019. A priority-based max-min scheduling algorithm for cloud environment using fuzzy approach. In *International Conference on Computer Networks and Communication Technologies: ICCNCT 2018* (pp. 819–828). Springer Singapore.
41. Zhou, X., Zhang, G., Sun, J., Zhou, J., Wei, T., Hu, S.: Minimizing cost and makespan for workflow scheduling in cloud using fuzzy dominance sort based HEFT. Futur. Gener. Comput. Syst. **93**, 278–289 (2019)

42. Tong, Z., Deng, X., Chen, H., Mei, J., Liu, H.: QL-HEFT: a novel machine learning scheduling scheme base on cloud computing environment. Neural Comput. Appl. **32**, 5553–5570 (2020)
43. Nazar, T., Javaid, N., Waheed, M., Fatima, A., Bano, H. and Ahmed, N., 2019. Modified shortest job first for load balancing in cloud-fog computing. In *Advances on Broadband and Wireless Computing, Communication and Applications: Proceedings of the 13th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA-2018)* (pp. 63–76). Springer International Publishing.
44. Alworafi, M.A., Dhari, A., Al-Hashmi, A.A. and Darem, A.B., 2016, December. An improved SJF scheduling algorithm in cloud computing environment. In *2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT)* (pp. 208–212). IEEE.
45. Seth, S., Singh, N.: Dynamic heterogeneous shortest job first (DHSJF): a task scheduling approach for heterogeneous cloud computing systems. Int. J. Inf. Technol. **11**(4), 653–657 (2019)
46. Devi, D.C., Uthariaraj, V.R.: Load balancing in cloud computing environment using improved weighted round robin algorithm for nonpreemptive dependent tasks. Sci. World J. (2016). https://doi.org/10.1155/2016/3896065
47. Venkataraman, N.: Threshold based multi-objective memetic optimized round robin scheduling for resource efficient load balancing in cloud. Mobile Netw. Appl. **24**, 1214–1225 (2019)
48. Krishnaveni, H., Janita, V.S.: Completion time based sufferage algorithm for static task scheduling in cloud environment. Int. J. Pure Appl. Math. **119**(12), 13793–13797 (2018)
49. Dutta, M. and Aggarwal, N., 2016. Meta-heuristics based approach for workflow scheduling in cloud computing: a survey. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems: Proceedings of ICAIECES 2015* (pp. 1331–1345). Springer India.
50. Wu, F., Wu, Q., Tan, Y.: Workflow scheduling in cloud: a survey. J. Supercomput. **71**, 3373–3418 (2015)
51. Alkhanak, E.N., Lee, S.P., Khan, S.U.R.: Cost-aware challenges for workflow scheduling approaches in cloud computing environments: Taxonomy and opportunities. Futur. Gener. Comput. Syst. **50**, 3–21 (2015)
52. Masdari, M., ValiKardan, S., Shahi, Z., Azar, S.I.: Towards workflow scheduling in cloud computing: a comprehensive analysis. J. Netw. Comput. Appl. **66**, 64–82 (2016)
53. Fister, I., Jr., Yang, X.S., Fister, I., Brest, J., Fister, D.: A brief review of nature-inspired algorithms for optimization. Neural Evol. Comput. (2013). https://doi.org/10.48550/arXiv.1307.4186
54 Yang, X.S., He, X.: Nature-inspired optimization algorithms in engineering: overview and applications. Nat. -Inspired Comput. Eng. (2016). https://doi.org/10.1007/978-3-319-30235-5_1
55. Nanda, S.J., Panda, G.: A survey on nature inspired metaheuristic algorithms for partitional clustering. Swarm Evol. Comput. **16**, 1–18 (2014)
56. Ss, V.C., Hs, A.: Nature inspired meta heuristic algorithms for optimization problems. Computing **104**(2), 251–269 (2022)
57 Mirjalili, S., Mirjalili, S.: Genetic algorithm. Evol. Algorithms Neural Netw.: Theory Appl. **780**, 43–55 (2019)
58. Wang, Z., Tang, K., Yao, X.: A memetic algorithm for multilevel redundancy allocation. IEEE Trans. Reliab. **59**(4), 754–765 (2010)
59. Tilahun, S.L., Kassa, S.M. and Ong, H.C., 2012. A new algorithm for multilevel optimization problems using evolutionary strategy, inspired by natural adaptation. In *PRICAI 2012: Trends in Artificial Intelligence: 12th Pacific Rim International Conference on Artificial Intelligence, Kuching, Malaysia, September 3–7, 2012. Proceedings 12* (pp. 577–588). Springer Berlin Heidelberg.
60. Yang, S., Yao, X.: Experimental study on population-based incremental learning algorithms for dynamic optimization problems. Soft. Comput. **9**, 815–834 (2005)
61. Gandomi, M., Kashani, A.R., Farhadi, A., Akhani, M., Gandomi, A.H.: Spectral acceleration prediction using genetic programming based approaches. Appl. Soft Comput. **106**, 107326 (2021)
62. Hussain, I., Ullah, I., Ali, W., Muhammad, G., Ali, Z.: Exploiting lion optimization algorithm for sustainable energy management system in industrial applications. Sustain. Energy Technol. Assess. **52**, 102237 (2022)
63. Hosseini, S., Al Khaled, A.: A survey on the imperialist competitive algorithm metaheuristic: implementation in engineering domain and directions for future research. Appl. Soft Comput. **24**, 1078–1094 (2014)
64. Gomes, G.F., da Cunha, S.S., Ancelotti, A.C.: A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates. Eng. Comput. **35**, 619–626 (2019)
65. Guo, W., Chen, M., Wang, L., Mao, Y., Wu, Q.: A survey of biogeography-based optimization. Neural Comput. Appl. **28**, 1909–1926 (2017)
66. Aguilar-Rivera, R., Valenzuela-Rendón, M., Rodríguez-Ortiz, J.J.: Genetic algorithms and Darwinian approaches in financial applications: a survey. Expert Syst. Appl. **42**(21), 7684–7697 (2015)
67. Zames, G.: Genetic algorithms in search, optimization and machine learning. Inf Tech J **3**(1), 301 (1981)
68. Dasgupta, K., Mandal, B., Dutta, P., Mandal, J.K., Dam, S.: A genetic algorithm (ga) based load balancing strategy for cloud computing. Procedia Technol. **10**, 340–347 (2013)
69. Ge, Y. and Wei, G., 2010, October. GA-based task scheduler for the cloud computing systems. In *2010 International Conference on Web Information Systems and Mining* (Vol. 2, pp. 181–186). IEEE.
70. Zheng, Z., Wang, R., Zhong, H. and Zhang, X., 2011, March. An approach for cloud resource scheduling based on Parallel Genetic Algorithm. In *2011 3rd International Conference on Computer Research and Development* (Vol. 2, pp. 444–447). IEEE.
71. Wang, T., Liu, Z., Chen, Y., Xu, Y. and Dai, X., 2014, August. Load balancing task scheduling based on genetic algorithm in cloud computing. In *2014 IEEE 12th international conference on dependable, autonomic and secure computing* (pp. 146–152). IEEE.
72. Jang, S.H., Kim, T.Y., Kim, J.K., Lee, J.S.: The study of genetic algorithm-based task scheduling for cloud computing. Int. J. Cont. Autom. **5**(4), 157–162 (2012)
73. Liu, J., Luo, X.G., Zhang, X.M., Zhang, F., Li, B.N.: Job scheduling model for cloud computing based on multi-objective genetic algorithm. Int. J. Comput. Sci. Issues (IJCSI) **10**(1), 134 (2013)
74. Kaur, K., Chhabra, A., Singh, G.: Heuristics based genetic algorithm for scheduling static tasks in homogeneous parallel system. Int. J. Comput. Sci. Security (IJCSS) **4**(2), 183–198 (2010)
75. Ghorbannia Delavar, A., Aryan, Y.: HSGA: a hybrid heuristic algorithm for workflow scheduling in cloud systems. Clust. Comput. **17**, 129–137 (2014)
76. Yu, J., Buyya, R.: Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms. Sci. Program. **14**(3–4), 217–230 (2006)
77. Khajemohammadi, H., Fanian, A. and Gulliver, T.A., 2013, August. Fast workflow scheduling for grid computing based on a multi-objective genetic algorithm. In *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)* (pp. 96–101). IEEE.

78. Gu, J., Hu, J., Zhao, T., Sun, G.: A new resource scheduling strategy based on genetic algorithm in cloud computing environment. J. Comput. **7**(1), 42–52 (2012)
79. Huang, J.: The workflow task scheduling algorithm based on the GA model in the cloud computing environment. J. Softw. **9**(4), 873–880 (2014)
80. Nasonov, D., Butakov, N., Balakhontseva, M., Knyazkov, K. and Boukhanovsky, A.V., 2014. Hybrid evolutionary workflow scheduling algorithm for dynamic heterogeneous distributed computational environment. In *International Joint Conference SOCO'14-CISIS'14-ICEUTE'14: Bilbao, Spain, June 25th-27th, 2014, Proceedings* (pp. 83–92). Springer International Publishing.
81. Szabo, C., Sheng, Q.Z., Kroeger, T., Zhang, Y., Yu, J.: Science in the cloud: allocation and execution of data-intensive scientific workflows. J. Grid Comput. **12**, 245–264 (2014)
82. Shen, G. and Zhang, Y.Q., 2011. A shadow price guided genetic algorithm for energy aware task scheduling on cloud computers. In *Advances in Swarm Intelligence: Second International Conference, ICSI 2011, Chongqing, China, June 12-15, 2011, Proceedings, Part I 2* (pp. 522-529). Springer Berlin Heidelberg.
83. Kolodziej, J., Khan, S.U. and Xhafa, F., 2011, October. Genetic algorithms for energy-aware scheduling in computational grids. In *2011 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing* (pp. 17–24). IEEE.
84. Zhu, K., Song, H., Liu, L., Gao, J. and Cheng, G., 2011, December. Hybrid genetic algorithm for cloud computing applications. In *2011 IEEE Asia-Pacific Services Computing Conference* (pp. 182–187). IEEE.
85. Sawant, S., 2011. A genetic algorithm scheduling approach for virtual machine resources in a cloud computing environment.
86. Moscato, P.: On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms. Caltech Concurr. Comput. Program **826**, 37 (1989)
87. Jouglet, A., Oğuz, C., Sevaux, M.: Hybrid flow-shop: a memetic algorithm using constraint-based scheduling for efficient search. J. Mathe. Model. Algorithms **8**, 271–292 (2009)
88. Moscato, P., Norman, M.G.: A memetic approach for the traveling salesman problem implementation of a computational ecology for combinatorial optimization on message-passing systems. Parallel Comput. Trans. Appl. **1**, 177–186 (1992)
89. Kashani, M.H., Jahanshahi, M.: A new method based on memetic algorithm for task scheduling in distributed systems. Int. J. Simul. Syst. Sci. Technol. **10**(5), 26–32 (2009)
90. Padmavathi, S., Shalinie, S.M., Abhilaash, R.: A memetic algorithm based task scheduling considering communication cost on cluster of workstations. Int. J. Adv. Soft Comput. Appl. **2**, 174–190 (2010)
91. Sutar, S., Sawant, J. and Jadhav, J., 2006. Task scheduling for multiprocessor systems using memetic algorithms. In *4th International Working Conference Performance Modeling and Evaluation of Heterogeneous Networks (HET-NETs '06).*
92. Zhao, F., Tang, J.: A memetic algorithm combined particle swarm optimization with simulated annealing and its application on multiprocessor scheduling problem. Prz Elektrotechniczny **88**, 292–296 (2012)
93. Atashpaz-Gargari, E. and Lucas, C., 2007, September. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In *2007 IEEE Congress on Evolutionary Computation* (pp. 4661–4667). Ieee.
94. Behnamian, J., Zandieh, M.: A discrete colonial competitive algorithm for hybrid flowshop scheduling to minimize earliness and quadratic tardiness penalties. Expert Syst. Appl. **38**(12), 14490–14498 (2011)
95. Attar, S.F., Mohammadi, M., Tavakkoli-Moghaddam, R.: A novel imperialist competitive algorithm to solve flexible flow shop scheduling problem in order to minimize maximum completion time. Int. J. Comput. Appl. **28**(10), 27–32 (2011)
96. Madani-Isfahani, M., Ghobadian, E., Tekmehdash, H., Tavakkoli-Moghaddam, R., Naderi-Beni, M.: An imperialist competitive algorithm for a bi-objective parallel machine scheduling problem with load balancing consideration. Int. J. Ind. Eng. Comput. **4**(2), 191–202 (2013)
97. Yakhchi, S., Ghafari, S.M., Yakhchi, M., Fazeli, M. and Patooghy, A., 2015, March. ICA-MMT: A load balancing method in cloud computing environment. In *2015 2nd World Symposium on Web Applications and Networking (WSWAN)* (pp. 1–7). IEEE.
98. Yousefyan, S., Dastjerdi, A.V. and Salehnamadi, M.R., 2013, May. Cost effective cloud resource provisioning with imperialist competitive algorithm optimization. In *The 5th Conference on Information and Knowledge Technology* (pp. 55–60). IEEE.
99. Pooranian, Z., Shojafar, M., Javadi, B., Abraham, A.: Using imperialist competition algorithm for independent task scheduling in grid computing. J. Intell. Fuzzy Syst. **27**(1), 187–199 (2014)
100. Piroozfard, H. and Wong, K.Y., 2014, December. An imperialist competitive algorithm for the job shop scheduling problems. In *2014 IEEE International Conference on Industrial Engineering and Engineering Management* (pp. 69–73). IEEE.
101. Jula, A., Othman, Z. and Sundararajan, E., 2013, April. A hybrid imperialist competitive-gravitational attraction search algorithm to optimize cloud service composition. In *2013 IEEE workshop on memetic computing (MC)* (pp. 37–43). IEEE.
102. Jula, A., Othman, Z., Sundararajan, E.: Imperialist competitive algorithm with PROCLUS classifier for service time optimization in cloud computing service composition. Expert Syst. Appl. **42**(1), 135–145 (2015)
103. Fatemipour, F. and Fatemipour, F., 2012. Scheduling scientific workflows using imperialist competitive algorithm. In *International conference on industrial intelligent information (ICIII 2012)* (pp. 218–225).
104. Faragardi, H.R., Rajabi, A., Shojaee, R. and Nolte, T., 2013, November. Towards energy-aware resource scheduling to maximize reliability in cloud computing systems. In *2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing* (pp. 1469–1479). IEEE.
105. Rajakumar, B.R.: The Lion's Algorithm: a new nature-inspired search algorithm. Procedia Technol. **6**, 126–135 (2012)
106. Yazdani, M., Jolai, F.: Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm. J. Comput. Design Eng. **3**(1), 24–36 (2016)
107. Emami, H.: Cloud task scheduling using enhanced sunflower optimization algorithm. Ict Express **8**(1), 97–100 (2022)
108. Subhash, L.S., Udayakumar, R.: Sunflower whale optimization algorithm for resource allocation strategy in cloud computing platform. Wireless Pers. Commun. **116**, 3061–3080 (2021)
109. Chandrashekar, C., Krishnadoss, P.: Opposition based sunflower optimization algorithm using cloud computing environments. Mater. Today: Proc. **62**, 4896–4902 (2022)
110. Jena, U.K., Kumar Das, P., Kabat, M.R., Kuanar, S.K.: Dynamic load balancing in cloud network through sunflower optimization algorithm and sine-cosine algorithm. In: Next generation of internet of things: proceedings of ICNGIoT 2022, pp. 609–621. Springer Nature Singapore, Singapore (2022)
111. Mahale, R.A., Chavan, S.D.: A survey: evolutionary and swarm based bio-inspired optimization algorithms. Int. J. Sci. Res. Publ. **2**(12), 1–6 (2012)
112. Juneja, M. and Nagar, S.K., 2016, October. Particle swarm optimization algorithm and its parameters: A review. In *2016*

International Conference on Control, Computing, Communication and Materials (ICCCCM) (pp. 1–5). IEEE.

113. Yuce, B., Packianather, M.S., Mastrocinque, E., Pham, D.T., Lambiase, A.: Honey bees inspired optimization method: the bees algorithm. Insects **4**(4), 646–662 (2013)
114. Mirjalili, S., Lewis, A.: The whale optimization algorithm. Adv. Eng. Softw. **95**, 51–67 (2016)
115. Senthilnath, J., Omkar, S.N., Mani, V.: Clustering using firefly algorithm: performance study. Swarm Evol. Comput. **1**(3), 164–171 (2011)
116. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. Appl. Soft Comput. **8**(1), 687–697 (2008)
117. Blum, C.: Ant colony optimization: Introduction and recent trends. Phys. Life Rev. **2**(4), 353–373 (2005)
118. Neshat, M., Sepidnam, G., Sargolzaei, M., Toosi, A.N.: Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications. Artif. Intell. Rev. **42**(4), 965–997 (2014)
119. Yang, X.S., He, X.: Bat algorithm: literature review and applications. Int. J. Bio-inspired Comput. **5**(3), 141–149 (2013)

120 Ahmed, A.M., Rashid, T.A., Saeed, S.A.M.: Cat swarm optimization algorithm: a survey and performance evaluation. Comput. Intell. Neurosci. (2020). https://doi.org/10.1155/2016/3896065

121. Ajith, A., Crina, G., Vitorino, R., Martin, R., Stephen, W.: Termite: a swarm intelligent routing algorithm for mobilewireless Ad-Hoc networks, pp. 155–184. Springer, Berlin (2006)
122. Pinto, P., Runkler, T.A. and Sousa, J.M., 2005. Wasp swarm optimization of logistic systems. In *Adaptive and Natural Computing Algorithms: Proceedings of the International Conference in Coimbra, Portugal, 2005* (pp. 264–267). Springer Vienna.

123 Chen, X., Zhou, Y., Luo, Q.: A hybrid monkey search algorithm for clustering analysis. Sci. World J. (2014). https://doi.org/10.1155/2014/938239

124. YongBo, C., YueSong, M., JianQiao, Y., XiaoLong, S., Nuo, X.: Three-dimensional unmanned aerial vehicle path planning using modified wolf pack search algorithm. Neurocomputing **266**, 445–457 (2017)
125. Lu, X. and Zhou, Y., 2008. A novel global convergence algorithm: bee collecting pollen algorithm. In *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence: 4th International Conference on Intelligent Computing, ICIC 2008 Shanghai, China, September 15–18, 2008 Proceedings 4* (pp. 518–525). Springer Berlin Heidelberg.
126. Kenan Dosoglu, M., Guvenc, U., Duman, S., Sonmez, Y., Tolga Kahraman, H.: Symbiotic organisms search optimization algorithm for economic/emission dispatch problem in power systems. Neural Comput. Appl. **29**, 721–737 (2018)
127. Meraihi, Y., Gabis, A.B., Ramdane-Cherif, A., Acheli, D.: A comprehensive survey of crow search algorithm and its applications. Artif. Intell. Rev. **54**(4), 2669–2716 (2021)
128. Dhanya, D., Arivudainambi, D.: Dolphin partner optimization based secure and qualified virtual machine for resource allocation with streamline security analysis. Peer-to-Peer Netw. Appl. **12**, 1194–1213 (2019)
129. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Softw. **69**, 46–61 (2014)
130. Pilla, R., Botcha, N., Gorripotu, T.S. and Azar, A.T., 2020. Fuzzy PID controller for automatic generation control of interconnected power system tuned by glow-worm swarm optimization. In *Applications of Robotics in Industry Using Advanced Mechanisms: Proceedings of International Conference on Robotics and Its Industrial Applications 2019 1* (pp. 140–149). Springer International Publishing.
131. Dorigo, M., Blum, C.: Ant colony optimization theory: a survey. Theoret. Comput. Sci. **344**(2–3), 243–278 (2005)
132. Chiang, C.W., Lee, Y.C., Lee, C.N., Chou, T.Y.: Ant colony optimisation for task matching and scheduling. IEE Proc. – Comput. Digital Tech. **153**(6), 373–380 (2006)
133. Chen, W.N., Zhang, J. and Yu, Y., 2007, September. Workflow scheduling in grids: an ant colony optimization approach. In *2007 IEEE Congress on Evolutionary Computation* (pp. 3308–3315). IEEE.
134. Chen, W.N., Shi, Y. and Zhang, J., 2009, May. An ant colony optimization algorithm for the time-varying workflow scheduling problem in grids. In *2009 IEEE Congress on Evolutionary Computation* (pp. 875–880). IEEE.
135. Pacini, E., Mateos, C., Garino, C.G.: Balancing throughput and response time in online scientific clouds via ant colony optimization (SP2013/2013/00006). Adv. Eng. Softw. **84**, 31–47 (2015)
136. Liu, X.F., Zhan, Z.H., Du, K.J. and Chen, W.N., 2014, July. Energy aware virtual machine placement scheduling in cloud computing based on ant colony optimization approach. In *Proceedings of the 2014 annual conference on genetic and evolutionary computation* (pp. 41–48).
137. Sivaraju, S.S., Kumar, C.: Energy enhancement of WSN with deep learning based SOM scheduling algorithm. J. Inf. Technol. Digital World **4**(3), 238–249 (2022)
138. Mathiyalagan, P., Suriya, S., Sivanandam, S.N.: Modified ant colony algorithm for grid scheduling. Int. J. Comput. Sci. Eng. **2**(02), 132–139 (2010)
139. Liu, A. and Wang, Z., 2008, October. Grid task scheduling based on adaptive ant colony algorithm. In *2008 International conference on management of e-commerce and e-government* (pp. 415–418). IEEE.
140. Bagherzadeh, J. and MadadyarAdeh, M., 2009, October. An improved ant algorithm for grid scheduling problem. In *2009 14th International CSI Computer Conference* (pp. 323–328). IEEE.
141. Chen, W.N., Zhang, J.: An ant colony optimization approach to a grid workflow scheduling problem with various QoS requirements. IEEE Trans. Syst. Man Cybernetics Part C **39**(1), 29–43 (2008)
142. Tawfeek, M.A., El-Sisi, A., Keshk, A.E. and Torkey, F.A., 2013, November. Cloud task scheduling based on ant colony optimization. In *2013 8th international conference on computer engineering & systems (ICCES)* (pp. 64–69). IEEE.
143. Khambre, P.D., Deshpande, A., Mehta, A., Sain, A.: Modified pheromone update rule to implement ant colony optimization algorithm for workflow scheduling algorithm problem in grids. Int. J. Adv. Res. Comput. Sci. Technol. **2**(2), 424–429 (2014)
144. Singh, L., Singh, S.: Deadline and cost based ant colony optimization algorithm for scheduling workflow applications in hybrid cloud. J. Sci. Eng. Res. **5**(10), 1417–1420 (2014)
145. Eberhart, R. and Kennedy, J., 1995, November. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks* (Vol. 4, pp. 1942–1948).
146. Pandey, S., Wu, L., Guru, S.M. and Buyya, R., 2010, April. A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In *2010 24th IEEE international conference on advanced information networking and applications* (pp. 400–407). IEEE.
147. Wu, Z., Ni, Z., Gu, L. and Liu, X., 2010, December. A revised discrete particle swarm optimization for cloud workflow scheduling. In *2010 international conference on computational intelligence and security* (pp. 184–188). IEEE.
148. Xue, S.J., Wu, W.: Scheduling workflow in cloud computing based on hybrid particle swarm algorithm. Indonesian J. Electr. Eng. Comput. Sci. **10**(7), 1560–1566 (2012)

149. Tavakkoli-Moghaddam, R., Azarkish, M., Sadeghnejad-Barkousaraie, A.: A new hybrid multi-objective Pareto archive PSO algorithm for a bi-objective job shop scheduling problem. Expert Syst. Appl. **38**(9), 10812–10821 (2011)
150. Beegom, A.A. and Rajasree, M.S., 2014. A particle swarm optimization based pareto optimal task scheduling in cloud computing. In *Advances in Swarm Intelligence: 5th International Conference, ICSI 2014, Hefei, China, October 17–20, 2014, Proceedings, Part II 5* (pp. 79–86). Springer International Publishing.
151. Karimi, M., Motameni, H.: Tasks scheduling in computational grid using a hybrid discrete particle swarm optimization. Int. J. Grid Distrib. Comput. **6**(2), 29–38 (2013)
152. Pooranian, Z., Shojafar, M., Abawajy, J.H., Abraham, A.: An efficient meta-heuristic algorithm for grid computing. J. Comb. Optim. **30**, 413–434 (2015)
153. Krishnasamy, K.: Task scheduling algorithm based on hybrid particle swarm optimization in cloud computing environment. J. Theor. Appl. Inf. Technol. **55**(1), 1–3 (2013)
154. Sridhar, M. and Babu, G.R.M., 2015, June. Hybrid particle swarm optimization scheduling for cloud computing. In *2015 IEEE International Advance Computing Conference (IACC)* (pp. 1196–1200). IEEE.
155. Al-maamari, A. and Omara, F.A., 2015. Task scheduling using hybrid algorithm in cloud computing environments. *Journal of Computer Engineering (IOSR-JCE)*, *17*(3), pp.96–106.
156. Zhang, L., Chen, Y., Sun, R., Jing, S., Yang, B.: A task scheduling algorithm based on PSO for grid computing. Int. J. Comput. Intell. Res. **4**(1), 37–43 (2008)
157. Liu, H., Abraham, A., Hassanien, A.E.: Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm. Futur. Gener. Comput. Syst. **26**(8), 1336–1343 (2010)
158. Aron, R., Chana, I., Abraham, A.: A hyper-heuristic approach for resource provisioning-based scheduling in grid environment. J. Supercomput. **71**, 1427–1450 (2015)
159. Sidhu, M.S., Thulasiraman, P. and Thulasiram, R.K., 2013, April. A load-rebalance PSO heuristic for task matching in heterogeneous computing systems. In *2013 IEEE Symposium on Swarm Intelligence (SIS)* (pp. 180–187). IEEE.
160. Ramezani, F., Lu, J., Hussain, F.K.: Task-based system load balancing in cloud computing using particle swarm optimization. Int. J. Parallel Prog. **42**, 739–754 (2014)
161. Milani, F.S., Navin, A.H.: Multi-objective task scheduling in the cloud computing based on the Patrice swarm optimization. Int. J. Inf. Technol. Comput. Sci. **7**(5), 61–66 (2015)
162. Wang, Z., Shuang, K., Yang, L., Yang, F.: Energy-aware and revenue-enhancing combinatorial scheduling in virtualized of cloud datacenter. J. Converg. Inf. Technol. **7**(1), 62–70 (2012)
163. Karaboga, D., 2005. *An idea based on honey bee swarm for numerical optimization* (Vol. 200, pp. 1–10). Technical report-tr06, Erciyes university, engineering faculty, computer engineering department.
164. Liu, Y.F., Liu, S.Y.: A hybrid discrete artificial bee colony algorithm for permutation flowshop scheduling problem. Appl. Soft Comput. **13**(3), 1459–1463 (2013)
165. Huang, Y.M., Lin, J.C.: A new bee colony optimization algorithm with idle-time-based filtering scheme for open shop-scheduling problems. Expert Syst. Appl. **38**(5), 5438–5447 (2011)
166. Ziarati, K., Akbari, R., Zeighami, V.: On the performance of bee algorithms for resource-constrained project scheduling problem. Appl. Soft Comput. **11**(4), 3720–3733 (2011)
167. Karaboga, D. and Gorkemli, B., 2011, June. A combinatorial artificial bee colony algorithm for traveling salesman problem. In *2011 International Symposium on Innovations in Intelligent Systems and Applications* (pp. 50–53). IEEE.
168. Hashemi, S.M., Hanani, A.: Solving the scheduling problem in computational grid using artificial bee colony algorithm. Adv. Comput. Sci.: Int. J. **2**, 37–41 (2013)
169. Mousavinasab, Z., Entezari-Maleki, R. and Movaghar, A., 2011. A bee colony task scheduling algorithm in computational grids. In *Digital Information Processing and Communications: International Conference, ICDIPC 2011, Ostrava, Czech Republic, July 7-9, 2011, Proceedings, Part I* (pp. 200-210). Springer Berlin Heidelberg
170. de Mello, R.F., Senger, L.J. and Yang, L.T., 2006, April. A routing load balancing policy for grid computing environments. In *20th International Conference on Advanced Information Networking and Applications-Volume 1 (AINA'06)* (Vol. 1, pp. 6-pp). IEEE.
171 Dhinesh Babu, L.D., Krishna, P.V.: Honey bee behavior inspired load balancing of tasks in cloud computing environments. Appl. Soft Comput. **13**(5), 2292–2303 (2013)
172. Soni, A., Vishwakarma, G., Jain, Y.K.: A bee colony based multi-objective load balancing technique for cloud computing environment. Int. J. Comput. Appl. **114**(4), 19–25 (2015)
173. Priyadarsini, R.J., Arockiam, L.: PBCOPSO: A parallel optimization algorithm for task scheduling in cloud environment. Indian J. Sci. Technol. **8**(16), 1–5 (2015)
174. Kashani, M.H., Jamei, M., Akbari, M. and Tayebi, R.M., 2011, July. Utilizing bee colony to solve task scheduling problem in distributed systems. In *2011 Third International Conference on Computational Intelligence, Communication Systems and Networks* (pp. 298–303). IEEE.
175. Navimipour, N.J., 2015, June. Task scheduling in the cloud environments based on an artificial bee colony algorithm. In *International Conference on Image Processing* (pp. 38–44).
176. Hesabian, N., Haj, H., Javadi, S.: Optimal scheduling in cloud computing environment using the bee algorithm. Int J Comput Netw Commun Secur **3**, 253–258 (2015)
177. Udomkasemsub, O., Xiaorong, L. and Achalakul, T., 2012, May. A multiple-objective workflow scheduling framework for cloud data analytics. In *2012 Ninth International Conference on Computer Science and Software Engineering (JCSSE)* (pp. 391–398). IEEE.
178. Liang, Y.C., Chen, A.H.L. and Nien, Y.H., 2014, July. Artificial bee colony for workflow scheduling. In *2014 IEEE Congress on Evolutionary Computation (CEC)* (pp. 558–564). IEEE.
179. Kansal, N.J., Chana, I.: Artificial bee colony based energy-aware resource utilization technique for cloud computing. Concurr. Comput.: Practice Exp. **27**(5), 1207–1225 (2015)
180. Hasançebi, O., Teke, T., Pekcan, O.: A bat-inspired algorithm for structural optimization. Comput. Struct. **128**, 77–90 (2013)
181. Jacob, L.: Bat algorithm for resource scheduling in cloud computing. Population **5**(18), 23 (2014)
182. Kumar, V.S., Aramudhan, M.: Trust based resource selection in cloud computing using hybrid algorithm. Int. J. Intell. Syst. Appl. **7**(8), 59 (2015)
183. Kumar, V.S.: Hybrid optimized list scheduling and trust based resource selection in cloud computing. J. Theor. Appl. Inf. Technol. **69**(3), 434–442 (2014)
184. Raghavan, S., Sarwesh, P., Marimuthu, C. and Chandrasekaran, K., 2015, January. Bat algorithm for scheduling workflow applications in cloud. In *2015 International Conference on Electronic Design, Computer Networks & Automated Verification (EDCAV)* (pp. 139–144). IEEE.
185. George, S.: Hybrid PSO-MOBA for profit maximization in cloud computing. Int J Adv Comput Sci Appl **6**(2), 159–163 (2015)
186. Chu, S.C., Tsai, P.W. and Pan, J.S., 2006. Cat swarm optimization. In *PRICAI 2006: Trends in Artificial Intelligence: 9th Pacific Rim International Conference on Artificial Intelligence Guilin, China, August 7–11, 2006 Proceedings 9* (pp. 854–858). Springer Berlin Heidelberg.

187. Chu, S.C., Tsai, P.W.: Computational intelligence based on the behavior of cats. Int. J. Innov. Comput. Inf. Control **3**(1), 163–173 (2007)
188. Tsai, P.W., Pan, J.S., Chen, S.M., Liao, B.Y. and Hao, S.P., 2008, July. Parallel cat swarm optimization. In *2008 international conference on machine learning and cybernetics* (Vol. 6, pp. 3328–3333). IEEE.
189. Pradhan, P.M., Panda, G.: Solving multiobjective problems using cat swarm optimization. Expert Syst. Appl. **39**(3), 2956–2964 (2012)
190. Sharafi, Y., Khanesar, M.A. and Teshnehlab, M., 2013, September. Discrete binary cat swarm optimization algorithm. In *2013 3rd IEEE international conference on computer, control and communication (IC4)* (pp. 1–6). IEEE.
191. Bilgaiyan, S., Sagnika, S. and Das, M., 2014, February. Workflow scheduling in cloud computing environment using cat swarm optimization. In *2014 IEEE International Advance Computing Conference (IACC)* (pp. 680–685). IEEE.
192. Rouhi, S. and Nejad, E.B., 2015. CSO-GA: a new scheduling technique for cloud computing systems based on cat swarm optimization and genetic algorithm. *Fen Bilimleri Dergisi (CFD)*, *36*(4).
193. Hof, P.R., Van der Gucht, E.: Structure of the cerebral cortex of the humpback whale, *Megaptera novaeangliae* (Cetacea, Mysticeti, Balaenopteridae). Anat. Rec.: Adv. Integr. Anat. Evolut. Biol.: Adv. Integr. Anat. Evolut. Biol. **290**(1), 1–31 (2007)
194. Mangalampalli, S., Karri, G.R., Kose, U.: Multi Objective Trust aware task scheduling algorithm in cloud computing using whale optimization. J. King Saud Univ.-Comput. Inf. Sci. **35**(2), 791–809 (2023)
195. Mangalampalli, S., Swain, S.K., Mangalampalli, V.K.: Prioritized energy efficient task scheduling algorithm in cloud computing using whale optimization algorithm. Wireless Pers. Commun. **126**(3), 2231–2247 (2022)
196. Sreenu, K., Sreelatha, M.: W-Scheduler: whale optimization for task scheduling in cloud computing. Clust. Comput. **22**, 1087–1098 (2019)
197. Chen, X., Cheng, L., Liu, C., Liu, Q., Liu, J., Mao, Y., Murphy, J.: A WOA-based optimization approach for task scheduling in cloud computing systems. IEEE Syst. J. **14**(3), 3117–3128 (2020)
198. Jia, L., Li, K., Shi, X.: Cloud computing task scheduling model based on improved whale optimization algorithm. Wirel. Commun. Mob. Comput. **2021**, 1–13 (2021)
199. Masadeh, R., Sharieh, A., Mahafzah, B.: Humpback whale optimization algorithm based on vocal behavior for task scheduling in cloud computing. Int. J. Adv. Sci. Technol. **13**(3), 121–140 (2019)
200 Arora, S., Singh, S.: The firefly optimization algorithm: convergence analysis and parameter selection. Int. J. Comp. Appl. (2013). https://doi.org/10.5120/11826-7528
201. Mangalampalli, S., Karri, G.R., Elngar, A.A.: An efficient trust-aware task scheduling algorithm in cloud computing using firefly optimization. Sensors **23**(3), 1384 (2023)
202. Ebadifard, F., Doostali, S. and Babamir, S.M., 2018, December. A firefly-based task scheduling algorithm for the cloud computing environment: Formal verification and simulation analyses. In *2018 9th International Symposium on Telecommunications (IST)* (pp. 664–669). IEEE.
203. Malleswaran, S.K.A., Kasireddi, B.: An efficient task scheduling method in a cloud computing environment using firefly crow search algorithm (FF-CSA). Int. J. Sci. Technol. Res. **8**(12), 623–627 (2019)
204. Rajagopalan, A., Modale, D.R. and Senthilkumar, R., 2020. Optimal scheduling of tasks in cloud computing using hybrid firefly-genetic algorithm. In *Advances in Decision Sciences, Image Processing, Security and Computer Vision: International Conference on Emerging Trends in Engineering (ICETE), Vol. 2* (pp. 678–687). Springer International Publishing.
205. Kashikolaei, S.M.G., Hosseinabadi, A.A.R., Saemi, B., Shareh, M.B., Sangaiah, A.K., Bian, G.B.: An enhancement of task scheduling in cloud computing based on imperialist competitive algorithm and firefly algorithm. J. Supercomput. **76**, 6302–6329 (2020)
206 Fanian, F., Bardsiri, V.K., Shokouhifar, M.: A new task scheduling algorithm using firefly and simulated annealing algorithms in cloud computing. Int. J. Adv. Comput. Sci. Appl. (2018). https://doi.org/10.14569/IJACSA.2018.090228
207. Du, Y., Wang, J.L., Lei, L.: Multi-objective scheduling of cloud manufacturing resources through the integration of cat swarm optimization and firefly algorithm. Adv. Prod. Eng. Manag. (2019). https://doi.org/10.14743/apem2019.3.331
208. Ammari, A.C., Labidi, W., Mnif, F., Yuan, H., Zhou, M., Sarrab, M.: Firefly algorithm and learning-based geographical task scheduling for operational cost minimization in distributed green data centers. Neurocomputing **490**, 146–162 (2022)
209 Zolghadr-Asli, B., Bozorg-Haddad, O., Chu, X.: Crow search algorithm (CSA). Adv. Optim. Nat. -inspired Algorithms (2018). https://doi.org/10.1007/978-981-10-5221-7_14
210. Prasanna Kumar, K.R., Kousalya, K.: Amelioration of task scheduling in cloud computing using crow search algorithm. Neural Comput. Appl. **32**, 5901–5907 (2020)
211. Kumar, K.P., Kousalya, K., Vishnuppriya, S., Ponni, S. and Logeswaran, K., 2021, February. Enhanced Crow Search Algorithm for Task Scheduling in Cloud Computing. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1055, No. 1, p. 012102). IOP Publishing.
212. Singh, H., Tyagi, S., Kumar, P.: Crow–penguin optimizer for multiobjective task scheduling strategy in cloud computing. Int. J. Commun. Syst. **33**(14), e4467 (2020)
213. Singh, H., Tyagi, S., Kumar, P.: Crow search based scheduling algorithm for load balancing in cloud environment. Int. J. Innov. Technol. Explor. Eng. (IJITEE) **8**(9), 1058–1064 (2019)
214. Singh, H., Tyagi, S., Kumar, P.: Cloud resource mapping through crow search inspired metaheuristic load balancing technique. Comput. Electr. Eng. **93**, 107221 (2021)
215 Wang, J.: Grey wolf optimization and crow search algorithm for resource allocation scheme in cloud computing: grey wolf optimization and crow search algorithm in cloud computing. Multime'd. Res. (2021). https://doi.org/10.46253/j.mr.v4i3.a3
216 Kak, S.M., Agarwal, P., Alam, M.A., Siddiqui, F.: A hybridized approach for minimizing energy in cloud computing. Clus. Comput. (2022). https://doi.org/10.1007/s10586-022-03807-9
217. Mangalampalli, S., Mangalampalli, V.K. and Swain, S.K., A Task scheduling approach in cloud computing to minimize the power cost in datacenters using crow search.
218. Joshi, A.S., Kulkarni, O., Kakandikar, G.M., Nandedkar, V.M.: Cuckoo search optimization-a review. Mater. Today: Proc. **4**(8), 7262–7269 (2017)
219. Elnahary, M.K., Hamed, A.Y., El-Sayed, H.: Task scheduling optimization in cloud computing by cuckoo search algorithm. Sohag J. Sci. **7**(3), 29–37 (2022)
220. Navimipour, N.J., Milani, F.S.: Task scheduling in the cloud computing based on the cuckoo search algorithm. Int. J. Model. Optim. **5**(1), 44 (2015)
221. Prem Jacob, T., Pradeep, K.: A multi-objective optimal task scheduling in cloud environment using cuckoo particle swarm optimization. Wireless Pers. Commun. **109**, 315–331 (2019)
222. Krishnadoss, P., Pradeep, N., Ali, J., Nanjappan, M., Krishnamoorthy, P., Kedalu Poornachary, V.: CCSA: Hybrid cuckoo crow search algorithm for task scheduling in cloud computing. Int. J. Intell. Eng. Syst. **14**(4), 241–250 (2021)

223. Agarwal, M. and Srivastava, G.M.S., 2018. A cuckoo search algorithm-based task scheduling in cloud computing. In *Advances in Computer and Computational Sciences: Proceedings of ICCCCS 2016, Volume 2* (pp. 293–299). Springer Singapore.

224. Nazir, S., Shafiq, S., Iqbal, Z., Zeeshan, M., Tariq, S. and Javaid, N., 2019. Cuckoo optimization algorithm based job scheduling using cloud and fog computing in smart grid. In *Advances in Intelligent Networking and Collaborative Systems: The 10th International Conference on Intelligent Networking and Collaborative Systems (INCoS-2018)* (pp. 34–46). Springer International Publishing.

225. Gawali, M.B., Shinde, S.K.: Standard deviation based modified cuckoo optimization algorithm for task scheduling to efficient resource allocation in cloud computing. J. Adv. Inf. Technol. **8**, 4 (2017)

226. Madni, S.H.H., Latiff, M.S.A., Ali, J., Abdulhamid, S.I.M.: Multi-objective-oriented cuckoo search optimization-based resource scheduling algorithm for clouds. Arab. J. Sci. Eng. **44**, 3585–3602 (2019)

227. Krishnadoss, P., Jacob, P.: OCSA: task scheduling algorithm in cloud computing environment. Int. J. Intell. Eng. Syst. **11**(3), 271–279 (2018)

228. Madni, S.H.H., Abd Latiff, M.S., Abdulhamid, S.I.M., Ali, J.: Hybrid gradient descent cuckoo search (HGDCS) algorithm for resource scheduling in IaaS cloud computing environment. Clust. Comput. **22**, 301–334 (2019)

229. Pradeep, K., Prem Jacob, T.: A hybrid approach for task scheduling using the cuckoo and harmony search in cloud computing environment. Wireless Pers. Commun. **101**, 2287–2311 (2018)

230. Shahdi-Pashaki, S., Teymourian, E., Kayvanfar, V., Komaki, G.M., Sajadi, A.: Group technology-based model and cuckoo optimization algorithm for resource allocation in cloud computing. IFAC-PapersOnLine **48**(3), 1140–1145 (2015)

231. Durgadevi, P., Srinivasan, S.: Resource allocation in cloud computing using SFLA and cuckoo search hybridization. Int. J. Parallel Prog. **48**, 549–565 (2020)

232. Faris, H., Aljarah, I., Al-Betar, M.A., Mirjalili, S.: Grey wolf optimizer: a review of recent variants and applications. Neural Comput. Appl. **30**, 413–435 (2018)

233. 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)

234. Sreenu, K., Malempati, S.: MFGMTS: Epsilon constraint-based modified fractional grey wolf optimizer for multi-objective task scheduling in cloud computing. IETE J. Res. **65**(2), 201–215 (2019)

235. Bacanin, N., Bezdan, T., Tuba, E., Strumberger, I., Tuba, M. and Zivkovic, M., 2019, November. Task scheduling in cloud computing environment by grey wolf optimizer. In *2019 27th telecommunications forum (TELFOR)* (pp. 1–4). IEEE.

236. Gobalakrishnan, N., Arun, C.: A new multi-objective optimal programming model for task scheduling using genetic gray wolf optimization in cloud computing. Comput. J. **61**(10), 1523–1536 (2018)

237. Natesan, G., Chokkalingam, A.: Task scheduling in heterogeneous cloud environment using mean grey wolf optimization algorithm. ICT Express **5**(2), 110–114 (2019)

238. Natesha, B.V., Sharma, N.K., Domanal, S. and Guddeti, R.M.R., 2018, September. GWOTS: grey wolf optimization based task scheduling at the green cloud data center. In *2018 14th International Conference on Semantics, Knowledge and Grids (SKG)* (pp. 181–187). IEEE.

239. Mohammadzadeh, A., Masdari, M., Gharehchopogh, F.S., Jafarian, A.: Improved chaotic binary grey wolf optimization algorithm for workflow scheduling in green cloud computing. Evol. Intel. **14**, 1997–2025 (2021)

240. Arora, N., Banyal, R.K.: A particle grey wolf hybrid algorithm for workflow scheduling in cloud computing. Wireless Pers. Commun. **122**(4), 3313–3345 (2022)

241. Balasubramanian, K., Ramya, K., Devi, K.G.: Improved swarm optimization of deep features for glaucoma classification using SEGSO and VGGNet. Biomed. Signal Process. Control **77**, 103845 (2022)

242. Zhou, J., Dong, S.: Hybrid glowworm swarm optimization for task scheduling in the cloud environment. Eng. Optim. **50**(6), 949–964 (2018)

243. Alboaneen, D.A., Tianfield, H. and Zhang, Y., 2017, March. Glowworm swarm optimisation based task scheduling for cloud computing. In *Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing* (pp. 1–7).

244. Abdullahi, M., Ngadi, M.A.: Symbiotic organism search optimization based task scheduling in cloud computing environment. Futur. Gener. Comput. Syst. **56**, 640–650 (2016)

245 Sa'ad, S., Muhammed, A., Abdullahi, M., Abdullah, A., Hakim Ayob, F.: An enhanced discrete symbiotic organism search algorithm for optimal task scheduling in the cloud. Algorithms **14**(7), 200 (2021)

246. Abdullahi, M., Ngadi, M.A., Dishing, S.I., Ahmad, B.I.E.: 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)

247 Abdullahi, M., Ngadi, M.A., Dishing, S.I., Abdulhamid, S.I.M.: An adaptive symbiotic organisms search for constrained task scheduling in cloud computing. J. Ambient Intell. Humanized Comput. (2022). https://doi.org/10.1007/s12652-021-03632-9

248. Sharma, M. and Verma, A., 2017, February. Energy-aware discrete symbiotic organism search optimization algorithm for task scheduling in a cloud environment. In *2017 4th International Conference on Signal Processing and Integrated Networks (SPIN)* (pp. 513–518). IEEE.

249. Zubair, A.A., Razak, S.A., Ngadi, M.A., Al-Dhaqm, A., Yafooz, W.M., Emara, A.H.M., Saad, A., Al-Aqrabi, H.: A cloud computing-based modified symbiotic organisms search algorithm (AI) for optimal task scheduling. Sensors **22**(4), 1674 (2022)

250. Siddique, N., Adeli, H.: Physics-based search and optimization: Inspirations from nature. Expert. Syst. **33**(6), 607–623 (2016)

251. Hashim, F.A., Houssein, E.H., Mabrouk, M.S., Al-Atabany, W., Mirjalili, S.: Henry gas solubility optimization: a novel physics-based algorithm. Futur. Gener. Comput. Syst. **101**, 646–667 (2019)

252 Aarts, E., Korst, J., Michiels, W.: Simulated annealing. Search Methodol.: Introd. Tutor. Optim. Decis. Support Techn. (2005). https://doi.org/10.1007/0-387-28356-0_7

253. Rashedi, E., Rashedi, E., Nezamabadi-Pour, H.: A comprehensive survey on gravitational search algorithm. Swarm Evol. Comput. **41**, 141–158 (2018)

254. Erol, O.K., Eksin, I.: A new optimization method: big bang–big crunch. Adv. Eng. Softw. **37**(2), 106–111 (2006)

255. Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.: GSA: a gravitational search algorithm. Inf. Sci. **179**(13), 2232–2248 (2009)

256. Kaveh, A., Talatahari, S.: A novel heuristic optimization method: charged system search. Acta Mech. **213**(3–4), 267–289 (2010)

257. Formato, R.A.: Central force optimization: a new deterministic gradient-like optimization metaheuristic. Opsearch **46**(1), 25–51 (2009)

258. Alatas, B.: ACROA: artificial chemical reaction optimization algorithm for global optimization. Expert Syst. Appl. **38**(10), 13170–13180 (2011)
259. Hatamlou, A.: Black hole: a new heuristic optimization approach for data clustering. Inf. Sci. **222**, 175–184 (2013)
260. Kaveh, A., Khayatazad, M.: A new meta-heuristic method: ray optimization. Comput. Struct. **112**, 283–294 (2012)
261. Abd Elaziz, M., Attiya, I.: An improved Henry gas solubility optimization algorithm for task scheduling in cloud computing. Artif. Intell. Rev. **54**, 3599–3637 (2021)
262. Wen, X., Huang, M. and Shi, J., 2012, October. 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* (pp. 219–222). IEEE.
263. Mathiyalagan, P., Sivanandam, S.N., Saranya, K.S.: Hybridization of modified ant colony optimization and intelligent water drops algorithm for job scheduling in computational grid. ICTACT J. Soft Comput. **4**(1), 651–655 (2013)
264. Cho, K.M., Tsai, P.W., Tsai, C.W., Yang, C.S.: A hybrid metaheuristic algorithm for VM scheduling with load balancing in cloud computing. Neural Comput. Appl. **26**, 1297–1309 (2015)
265. Madivi, R. and Kamath, S.S., 2014, July. An hybrid bio-inspired task scheduling algorithm in cloud environment. In *Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT)* (pp. 1–7). IEEE.
266. Singhal, U., Jain, S.: A new fuzzy logic and GSO based load balancing mechanism for public cloud. Int. J. Grid Distrib. Comput. **7**(5), 97–110 (2014)
267. Mandal, T. and Acharyya, S., 2015, December. Optimal task scheduling in cloud computing environment: meta heuristic approaches. In *2015 2nd International Conference on Electrical Information and Communication Technologies (EICT)* (pp. 24–28). IEEE.
268. Ramezani, F., Lu, J. and Hussain, F., 2013. Task scheduling optimization in cloud computing applying multi-objective particle swarm optimization. In *Service-Oriented Computing: 11th International Conference, ICSOC 2013, Berlin, Germany, December 2-5, 2013, Proceedings 11* (pp. 237-251). Springer Berlin Heidelberg.
269. Ramezani, F., Lu, J., Taheri, J., Hussain, F.K.: Evolutionary algorithm-based multi-objective task scheduling optimization model in cloud environments. World Wide Web **18**, 1737–1757 (2015)
270. 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)
271. He, H., Xu, G., Pang, S., Zhao, Z.: AMTS: Adaptive multi-objective task scheduling strategy in cloud computing. China Commun. **13**(4), 162–171 (2016)
272. Raju, R., Babukarthik, R.G., Chandramohan, D., Dhavachelvan, P. and Vengattaraman, T., 2013, February. Minimizing the makespan using Hybrid algorithm for cloud computing. In *2013 3rd IEEE International Advance Computing Conference (IACC)* (pp. 957–962). IEEE.
273. Khalili, A. and Babamir, S.M., 2015, May. Makespan improvement of PSO-based dynamic scheduling in cloud environment. In *2015 23rd Iranian Conference on Electrical Engineering* (pp. 613–618). IEEE.
274. Gabi, D., Ismail, A.S. and Dankolo, N.M., 2019, June. Minimized makespan based improved cat swarm optimization for efficient task scheduling in cloud datacenter. In *Proceedings of the 2019 3rd High Performance Computing and Cluster Technologies Conference* (pp. 16–20).
275. Frincu, M.E. and Craciun, C., 2011, December. Multi-objective meta-heuristics for scheduling applications with high availability requirements and cost constraints in multi-cloud environments. In *2011 Fourth IEEE International Conference on Utility and Cloud Computing* (pp. 267–274). IEEE.
276. Cui, H., Li, Y., Liu, X., Ansari, N., Liu, Y.: Cloud service reliability modelling and optimal task scheduling. IET Commun. **11**(2), 161–167 (2017)
277. Tao, F., Feng, Y., Zhang, L., Liao, T.W.: CLPS-GA: a case library and pareto solution-based hybrid genetic algorithm for energy-aware cloud service scheduling. Appl. Soft Comput. **19**, 264–279 (2014)
278. Goyal, A. and Chahal, N.S., 2015, November. Bio inspired approach for load balancing to reduce energy consumption in cloud data center. In *2015 Communication, Control and Intelligent Systems (CCIS)* (pp. 406–410). IEEE.
279. Meshkati, J., Safi-Esfahani, F.: Energy-aware resource utilization based on particle swarm optimization and artificial bee colony algorithms in cloud computing. J. Supercomput. **75**(5), 2455–2496 (2019)
280. Meena, J., Kumar, M., Vardhan, M.: Cost effective genetic algorithm for workflow scheduling in cloud under deadline constraint. IEEE Access **4**, 5065–5082 (2016)
281. Nasr, A.A., El-Bahnasawy, N.A., Attiya, G., El-Sayed, A.: Cost-effective algorithm for workflow scheduling in cloud computing under deadline constraint. Arab. J. Sci. Eng. **44**, 3765–3780 (2019)
282. Wu, Z., Liu, X., Ni, Z., Yuan, D., Yang, Y.: A market-oriented hierarchical scheduling strategy in cloud workflow systems. J. Supercomput. **63**, 256–293 (2013)
283. Gabi, D., Zainal, A., Ismail, A.S. and Zakaria, Z., 2017, May. Scalability-Aware scheduling optimization algorithm for multi-objective cloud task scheduling problem. In *2017 6th ICT International Student Project Conference (ICT-ISPC)* (pp. 1–6). IEEE.
284. Yassa, S., Chelouah, R., Kadima, H., Granado, B.: Multi-objective approach for energy-aware workflow scheduling in cloud computing environments. Sci. World J. (2013). https://doi.org/10.1155/2013/3509345
285. Li, Z., Ge, J., Yang, H., Huang, L., Hu, H., Hu, H., Luo, B.: A security and cost aware scheduling algorithm for heterogeneous tasks of scientific workflow in clouds. Futur. Gener. Comput. Syst. **65**, 140–152 (2016)
286. Wen, Y., Liu, J., Dou, W., Xu, X., Cao, B., Chen, J.: Scheduling workflows with privacy protection constraints for big data applications on cloud. Futur. Gener. Comput. Syst. **108**, 1084–1091 (2020)
287. Sharma, M., Garg, R.: HIGA: Harmony-inspired genetic algorithm for rack-aware energy-efficient task scheduling in cloud data centers. Eng. Sci. Technol. Int. J. **23**(1), 211–224 (2020)
288. Thanka, M.R., Uma Maheswari, P., Edwin, E.B.: An improved efficient: artificial bee colony algorithm for security and QoS aware scheduling in cloud computing environment. Clust. Comput. **22**, 10905–10913 (2019)
289. Maurya, A.K. and Tripathi, A.K., 2018, March. Deadline-constrained algorithms for scheduling of bag-of-tasks and workflows in cloud computing environments. In *Proceedings of the 2nd International Conference on High Performance Compilation, Computing and Communications* (pp. 6–10).
290. Wu, Q., Yun, D., Lin, X., Gu, Y., Lin, W. and Liu, Y., 2013. On workflow scheduling for end-to-end performance optimization in distributed network environments. In *Job Scheduling Strategies for Parallel Processing: 16th International Workshop, JSSPP 2012, Shanghai, China, May 25, 2012. Revised Selected Papers 16* (pp. 76-95). Springer Berlin Heidelberg.
291. Jianfang, C., Junjie, C., Qingshan, Z.: An optimized scheduling algorithm on a cloud workflow using a discrete particle swarm. Cybern. Inf. Technol. **14**(1), 25–39 (2014)

292. Sakellariou, R., Zhao, H.: A low-cost rescheduling policy for efficient mapping of workflows on grid systems. Sci. Program. **12**(4), 253–262 (2004)

293. Liu, K., 2009. *Scheduling algorithms for instance-intensive cloud workflows*. Swinburne University of Technology, Faculty of Engineering and Industrial Sciences, Centre for Complex Software Systems and Services.

294 Wang, X., Wang, Y., Zhu, H.: Energy-efficient multi-job scheduling model for cloud computing and its genetic algorithm. Math. Probl. Eng. (2012). https://doi.org/10.1155/2012/589243

295. Negru, C., Pop, F., Cristea, V., Bessisy, N. and Li, J., 2013, September. Energy efficient cloud storage service: key issues and challenges. In *2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies* (pp. 763–766). IEEE.

296. Shu, W., Wang, W., Wang, Y.: A novel energy-efficient resource allocation algorithm based on immune clonal optimization for green cloud computing. EURASIP J. Wirel. Commun. Netw. **2014**(1), 1–9 (2014)

297. Sellami, K., Ahmed-Nacer, M., Tiako, P.F., Chelouah, R.: Immune genetic algorithm for scheduling service workflows with QoS constraints in cloud computing. S. Afr. J. Ind. Eng. **24**(3), 68–82 (2013)

298. Zhao, C., Zhang, S., Liu, Q., Xie, J. and Hu, J., 2009, September. Independent tasks scheduling based on genetic algorithm in cloud computing. In *2009 5th international conference on wireless communications, networking and mobile computing* (pp. 1–4). IEEE.

299. Almezeini, N., Hafez, A.: Task scheduling in cloud computing using lion optimization algorithm. Int. J. Adv. Comput. Sci. Appl. (2017). https://doi.org/10.14569/IJACSA.2017.081110

300. Li, K., Xu, G., Zhao, G., Dong, Y. and Wang, D., 2011, August. Cloud task scheduling based on load balancing ant colony optimization. In *2011 sixth annual ChinaGrid conference* (pp. 3–9). IEEE.

301. Hu, Y., Xing, L., Zhang, W., Xiao, W. and Tang, D., 2010. A knowledge-based ant colony optimization for a grid workflow scheduling problem. In *Advances in Swarm Intelligence: First International Conference, ICSI 2010, Beijing, China, June 12-15, 2010, Proceedings, Part I 1* (pp. 241-248). Springer Berlin Heidelberg.

302. Liu, W., Peng, S., Du, W., Wang, W., Zeng, G.S.: Security-aware intermediate data placement strategy in scientific cloud workflows. Knowl. Inf. Syst. **41**, 423–447 (2014)

303. Javanmardi, S., Shojafar, M., Amendola, D., Cordeschi, N., Liu, H. and Abraham, A., 2014. Hybrid job scheduling algorithm for cloud computing environment. In *Proceedings of the fifth international conference on innovations in bio-inspired computing and applications IBICA 2014* (pp. 43–52). Springer International Publishing

304. Rodriguez, M.A., Buyya, R.: Deadline based resource provisioningand scheduling algorithm for scientific workflows on clouds. IEEE Trans. Cloud Comput. **2**(2), 222–235 (2014)

305. Verma, A. and Kaushal, S., 2014, March. Bi-criteria priority based particle swarm optimization workflow scheduling algorithm for cloud. In *2014 Recent Advances in Engineering and Computational Sciences (RAECS)* (pp. 1–6). IEEE.

306. Milan, S.T., Rajabion, L., Darwesh, A., Hosseinzadeh, M., Navimipour, N.J.: Priority-based task scheduling method over cloudlet using a swarm intelligence algorithm. Clust. Comput. **23**, 663–671 (2020)

307. Wang, X., Cao, B., Hou, C., Xiong, L. and Fan, J., 2015, October. Scheduling budget constrained cloud workflows with particle swarm optimization. In *2015 IEEE Conference on Collaboration and Internet Computing (CIC)* (pp. 219–226). IEEE.

308. Guo, P. and Xue, Z., 2017, October. Cost-effective fault-tolerant scheduling algorithm for real-time tasks in cloud systems. In *2017 IEEE 17th International Conference on Communication Technology (ICCT)* (pp. 1942–1946). IEEE.

309. Islam, M.R. and Habiba, M., 2012, December. Dynamic scheduling approach for data-intensive cloud environment. In *2012 International Conference on Cloud Computing Technologies, Applications and Management (ICCCTAM)* (pp. 179–185). IEEE.

310. Kumar, N. and Patel, P., 2016, March. Resource management using feed forward ANN-PSO in cloud computing environment. In *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies* (pp. 1–6).

311. Hu, H. and Wang, H., 2016, October. A prediction-based aco algorithm to dynamic tasks scheduling in cloud environment. In *2016 2nd IEEE International Conference on Computer and Communications (ICCC)* (pp. 2727–2732). IEEE.

312. Rahman, M., Hassan, R., Ranjan, R., Buyya, R.: Adaptive workflow scheduling for dynamic grid and cloud computing environment. Concurr. Comput.: Pract. Exp. **25**(13), 1816–1842 (2013)

313. Alla, H.B., Alla, S.B. and Ezzati, A., 2016, May. A novel architecture for task scheduling based on dynamic queues and particle swarm optimization in cloud computing. In *2016 2nd International Conference on Cloud Computing Technologies and Applications (CloudTech)* (pp. 108–114). IEEE.

314. Askarizade Haghighi, M., Maeen, M., Haghparast, M.: An energy-efficient dynamic resource management approach based on clustering and meta-heuristic algorithms in cloud computing IaaS platforms: energy efficient dynamic cloud resource management. Wireless Pers. Commun. **104**, 1367–1391 (2019)

315. Negi, S., Panwar, N., Vaisla, K.S. and Rauthan, M.M.S., 2020. Artificial neural network based load balancing in cloud environment. In *Advances in Data and Information Sciences: Proceedings of ICDIS 2019* (pp. 203–215). Springer Singapore.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Farida Siddiqi Prity completed her Bachelor of Science in Information & Communication Engineering in 2020 from Noakhali Science and Technology University, Bangladesh. She is currently studying Master of Science in Information & Communication Engineering at the same university. Her research interests are digital image processing, cloud computing, Artificial Intelligence, deep learning, transfer learning, and biomedical image processing.

Hasan Gazi has worked as an assistant programmer at the Supreme Court of Bangladesh. He completed his B.Sc. in Computer Science and Engineering from Patuakhali Science and Technology University, Bangladesh and M.Sc. degree in Computer Science from Jagannath University, Bangladesh. His research interests are Machine Learning, cloud computing, and image processing.

Aslam Uddin is currently working as a M.Phil. student under the institute of information technology at University of Dhaka, Bangladesh. Moreover, he is also an Assistant Professor of Information and

Communication Engineering at Noakhali Science and Technology University (NSTU), Noakhali, Bangladesh. He completed his B.Sc. in Computer Science and Engineering from Patuakhali Science and Technology University, Bangladesh and M.Sc. degree in Computer Science from Jahangirnagar University, Bangladesh. His research interest includes cloud computing, machine learning, software engineering.