



# Exploring swarm intelligence optimization techniques for task scheduling in cloud computing: algorithms, performance analysis, and future prospects

Farida Siddiqi Prity<sup>1</sup> · K. M. Aslam Uddin<sup>1</sup> · Nishu Nath<sup>1,2</sup>

Received: 17 July 2023 / Accepted: 13 October 2023 / Published online: 7 November 2023  
© The Author(s), under exclusive licence to Springer Nature Switzerland AG 2023

## Abstract

The advent of the cloud computing paradigm has enabled innumerable organizations to seamlessly migrate, compute, and host their applications within the cloud environment, affording them facile access to a broad spectrum of services with minimal exertion. A proficient and adaptable task scheduler is essential to manage simultaneous user requests for diverse cloud services using various heterogeneous and varied resources. Inadequate scheduling may result in issues related to either under-utilization or over-utilization of resources, potentially causing a waste of cloud resources or a decline in service performance. Swarm intelligence meta-heuristics optimization technique has evinced conspicuous efficacy in tackling the intricacies of scheduling difficulties. Thus, the present manuscript seeks to undertake an exhaustive review of swarm intelligence optimization techniques deployed in the task-scheduling domain within cloud computing. This paper examines various swarm-based algorithms, investigates their application to task scheduling in cloud environments, and provides a comparative analysis of the discussed algorithms based on various performance metrics. This study also compares different simulation tools for these algorithms, highlighting challenges and proposing potential future research directions in this field. This review paper aims to shed light on the state-of-the-art swarm-based algorithms for task scheduling in cloud computing, showing their potential to improve resource allocation, enhance system performance, and efficiently utilize cloud resources.

**Keywords** Cloud computing · Task scheduling · Meta-heuristics · Swarm intelligence

## 1 Introduction

The expanding landscape of the global corporate realm has stimulated the implementation of advanced technologies to address its complexities [1]. Among such innovations, cloud computing emerges as a promising solution, enabling expedited software deployment processes while saving time and effort [2]. Cloud computing allows the accessing and storing of information via the internet rather than relying on locally stored systems [3, 4]. It harnesses remote servers connected to the internet to store, manage, and facilitate online access to

data. Organizations must establish sophisticated cloud management systems that guarantee appropriate, dependable, and adaptable hardware and software to utilize cloud resources and optimize costs [5, 6]. With its inherent reliability and accessibility, cloud infrastructure minimizes the likelihood of infrastructure failures, enabling widespread and consistent server availability. By leveraging shared infrastructures, cloud computing enables cost-effective utilization of diverse applications by many users [7].

Task scheduling in cloud computing encompasses a complex subject matter that addresses multiple challenges related to effectiveness, resource utilization, and workload management [8]. Moreover, it necessitates considering diverse factors such as task dependencies, QoS specifications, assurances, and confidentiality concerns [9]. Most task-scheduling problems fall into the category of NP-complete or NP-hard, presenting significant computational complexities [1]. In contemporary computer systems, two primary scheduling

---

✉ K. M. Aslam Uddin  
aslam@nstu.edu.bd

<sup>1</sup> Department of Information and Communication Engineering, Noakhali Science and Technology University, Noakhali 3814, Bangladesh

<sup>2</sup> Gianforte School of Computing, Montana State University, Bozeman, MT, USA

techniques are employed: exhaustive algorithms and Deterministic Algorithms (DAs) [10]. DAs exhibit superior efficiency over traditional (exhaustive) approaches and heuristic methods for addressing scheduling problems [1]. However, DAs possess two key challenges: first, they are tailored to handle specific data distributions, and second, only a subset of DAs is equipped to tackle intricate scheduling issues [11]. To overcome these challenges, swarm intelligence optimization algorithms have gained significant attention in recent years for task scheduling of cloud computing. Inspired by the collective behavior of social insect colonies, swarm-based algorithms leverage the principles of self-organization and decentralized decision-making to solve complex optimization problems. These algorithms offer promising solutions for task scheduling in cloud computing by effectively balancing the workload and resources in a distributed and adaptive manner.

### 1.1 Research gap

This paper has drawn inspiration from previous peer surveys conducted in the literature, which have highlighted the significance of task scheduling in cloud computing. The goal of task scheduling in cloud computing is to optimize the utilization of virtual machines while minimizing data center operating costs, resulting in improved quality of service metrics and overall performance. By employing an effective task-scheduling technique, many user requests can be efficiently processed and assigned to suitable VMs, thereby meeting the needs of both cloud users and service providers more effectively. In the comprehensive review of existing literature on swarm intelligence scheduling approaches, we found that some studies [3, 5, 6, 8, 9] do not cover all the essential aspects, such as QoS-based comparative analysis, state-of-the-art advancements, comparisons of simulation tools, research gap identification, and future directions for task scheduling. In the paper [1], 12 swarm intelligence algorithms are evaluated to underscore the significance of swarm intelligence in cloud computing task scheduling. However, there is a pressing need to explore a more extensive array of algorithms to enhance our understanding of their effectiveness and applicability in addressing the evolving challenges of cloud-based task scheduling. This gap in the literature highlights the need for a thorough assessment of task scheduling using swarm intelligence optimization techniques to keep up with the continuously expanding research in this field. Therefore, this review aims to provide valuable insights and contribute to the advancement of task scheduling in cloud computing by addressing the comprehensive evaluation of swarm intelligence techniques and filling the existing research gaps.

### 1.2 Our contribution

The primary objective of this review paper is to provide a comprehensive analysis of swarm-based algorithms for task scheduling in cloud computing. This study explores various categorizations of task-scheduling techniques in cloud computing, detailing their advantages and disadvantages to underscore the significance of swarm intelligence. This study meticulously examines seventeen swarm-based algorithms, including Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Bat Algorithm (BA), Artificial Bee Colony (ABC), Whale Optimization Algorithm (WOA), Cat Swarm Optimization (CSO), Firefly Algorithm (FA), Grey Wolf Optimization (GSO), Crow Search, Cuckoo Search, Glowworm Swarm Optimization (GSO), Wild Horse Optimization (WHO), Symbiotic Organism Search (SOS), Chaotic Social Spider Optimization (CSSO), Monkey Search Optimization (MSO), Sea Lion Optimization (SLO), and Virus Optimization Algorithm (VOA). It investigates the practical application of these algorithms in the task-scheduling domain within cloud environments, offering a detailed comparative analysis based on a spectrum of performance metrics, including reliability, makespan, cost-efficiency, security, load balancing, rescheduling capability, energy efficiency, and resource utilization. The review also encompasses a comparative assessment of frequently employed simulation tools in cloud computing. Moreover, this study takes a comprehensive approach by examining the challenges and proposing potential avenues for future research. In particular, it highlights the state-of-the-art swarm-based algorithms deployed for task scheduling in cloud computing, emphasizing their pivotal role in optimizing resource allocation, enhancing system performance, and facilitating the efficient utilization of cloud resources. By illuminating these advancements and research gaps, the paper reinforces the significance of swarm-based algorithms in propelling the evolution of cloud computing while directing attention toward promising directions for further exploration and innovation.

### 1.3 Structure of paper

The paper's structure is outlined as follows: In Sect. 2, a detailed description of the categories of cloud task scheduling is provided, along with an analysis of the strengths and weaknesses of each technique. Section 3 delves into a comprehensive discussion of seventeen swarm intelligence algorithms and their applications in cloud computing task scheduling. Section 4 evaluates each swarm intelligence algorithm based on various performance metrics. Section 5 presents a comparative analysis of the simulation tools commonly employed in swarm intelligence algorithms. Sections 6 and 7 are dedicated to addressing the challenges and

exploring prospects associated with swarm intelligence algorithms. Finally, Sect. 8 offers a conclusion that meticulously discusses and underscores the significance of swarm intelligence algorithms in cloud computing task scheduling.

## 2 Categorization of cloud task-scheduling schemes

The categorization of cloud task-scheduling schemes entails a classification into three distinct groups, as shown in Fig. 1: traditional scheduling, heuristics scheduling, and meta-heuristics scheduling [1]. This section contrasts traditional and heuristic strategies, conducting a comprehensive evaluation and assessment of their effectiveness within cloud computing systems, particularly in the context of swarm intelligence optimization algorithms. By dissecting and comparing these approaches, this review aims to shed light on the strengths and limitations of each, ultimately contributing to a deeper understanding of their relevance and applicability in the dynamic landscape of cloud-based task scheduling.

### 2.1 Traditional scheduling

Traditional scheduling refers to the conventional methods and algorithms used for task scheduling in computing systems, including cloud computing. These approaches are typically rule-based and rely on predefined policies and heuristics to allocate tasks to available resources [12]. Traditional scheduling methods often prioritize task completion time, resource utilization, and load balancing. In cloud computing, traditional scheduling may involve allocating virtual machines or containers to specific tasks or applications based on fixed rules and priorities. While these methods can be straightforward to implement, they may not always be efficient or adaptive to changing workloads and resource availability, which has led to the exploration of more advanced scheduling techniques, including heuristics and meta-heuristics, to address the complexities of modern cloud environments [13, 14].

### 2.2 Heuristics scheduling

Heuristic scheduling in cloud computing involves using rule-based algorithms that make approximate decisions to allocate tasks or jobs to available resources efficiently [15]. Heuristic algorithms are designed to find reasonably good solutions promptly, even for complex optimization problems. However, their performance can vary depending on the specific problem they are applied to. While heuristics excel in certain situations, they may struggle with challenging optimization problems. Despite their limitations, heuristics are valuable for providing quick and practical solutions. Several

heuristic techniques have been developed in cloud computing to address task-scheduling challenges, including workflow and independent tasks and applications. Some notable heuristic algorithms used in cloud environments include Min–Min [16], Max–Min [17], First Come First Serve (FCFS) [18], Shortest Job First (SJF) [19], Round Robin (RR) [20], Heterogeneous Earliest Finish Time (HEFT) [21], Minimum Completion Time (MCT) [22], and Sufferage [23].

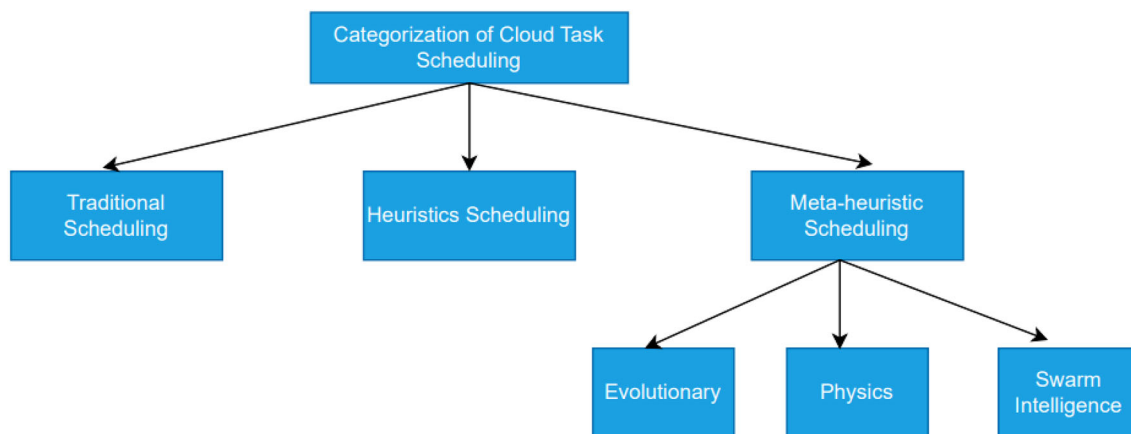
The Min–Min heuristic algorithm prioritizes the shortest task that can be completed in the least amount of time among all available tasks [24, 25]. This task is then assigned to a virtual machine that can complete it quickly. This process is repeated until all tasks are scheduled, potentially increasing the overall makespan as each task's completion time extends [26]. While Min–Min efficiently handles smaller tasks, it may lead to the starvation of larger jobs waiting for smaller ones to be completed [27, 28]. The Max–Min algorithm focuses on maximizing the minimum completion time of tasks, potentially leading to resource over-utilization and under-utilization [29–31]. Other heuristics like SJF, Round Robin, and Sufferage have strengths and weaknesses, ranging from load balancing issues to potential starvation problems in specific scenarios [32–34]. While heuristic algorithms provide practical solutions for many cloud scheduling problems, they may not always guarantee optimal results [1, 11, 35]. Researchers continue exploring and refining heuristic techniques to address their limitations and improve their performance in various cloud computing scenarios.

### 2.3 Meta-heuristics scheduling

Meta-heuristics scheduling in the context of task scheduling in cloud computing represents a higher-level approach to solving complex optimization problems by guiding and enhancing the search for near-optimal solutions [36]. Unlike traditional heuristics, meta-heuristics provide more flexibility and adaptability in exploring solution spaces and finding improved scheduling solutions [37, 38]. These algorithms are often inspired by natural phenomena or processes and are designed to overcome the limitations of traditional scheduling methods. Meta-heuristics scheduling is classified into three types:

- Evolutionary based;
- Physics based; and
- Swarm intelligence based.

In this sub-section, a detailed exploration of each technique is undertaken, highlighting their strengths and weaknesses. This comprehensive analysis emphasizes the significance of choosing swarm intelligence algorithms over other meta-heuristics scheduling techniques in cloud computing.



**Fig. 1** Categorization of task scheduling in cloud computing

### 2.3.1 Evolution based

Evolutionary-based algorithms, a subset of meta-heuristics, have gained popularity in cloud task scheduling due to their ability to handle complex optimization problems efficiently. These algorithms draw inspiration from natural selection and evolution to search for optimal solutions within vast solution spaces. Some examples of evolutionary-based algorithms used in cloud task scheduling include Genetic Algorithm (GA) [39], Memetic Algorithm (MA) [40], Evolution Strategy (ES) [41], Probability-Based Incremental Learning (PBIL) [42], Genetic Programming (GP) [43], and Differential Evolution (DE) [44].

Genetic Algorithms involve a population of potential solutions (chromosomes) subjected to genetic operators like mutation, crossover, and selection to generate improved offspring over generations [45]. In cloud task scheduling, GAs adaptively optimize tasks to virtual machines based on objectives like makespan or resource utilization. Advantages of GAs in cloud scheduling include their global search capabilities, adaptability to various objectives, robustness in dynamic environments, and suitability for parallel execution [46]. However, GA requires extensive computational resources parameter tuning and does not guarantee optimal solutions, particularly in large-scale scenarios [47, 48]. Differential evolution operates by generating mutant vectors from the population's solutions and combining them with existing individuals to improve the population's fitness. DE has been used in cloud task scheduling to optimize task-to-VM assignments and resource allocation [49]. However, DE may require fine-tuning of control parameters, and its performance can vary based on problem characteristics [50].

### 2.3.2 Physics based

Physics-based algorithms, a subset of meta-heuristics inspired by physical phenomena, have also found application in cloud task scheduling. These algorithms replicate principles from the physical world, such as the laws of motion and thermodynamics, to optimize task assignments and resource allocation in cloud computing environments [51]. While less prevalent than evolutionary-based algorithms, physics-based approaches provide unique perspectives on optimization. Some notable examples of physics-based algorithms in cloud task scheduling are Gravitational Search Algorithm (GSA) [52], Simulated Annealing (SA) [53], and Quantum-Behaved Particle Swarm Optimization (QPSO) [54].

The gravitational search algorithm models solutions as celestial bodies that exert gravitational forces on each other based on their fitness. This attraction–repulsion dynamic leads to the convergence of solutions toward optimal configurations. GSA's advantages include its ability to handle complex problems, adaptability to diverse objectives, and convergence to near-optimal solutions. However, GSA's performance may depend on parameter settings and the choice of gravitational laws [55]. In cloud task scheduling, SA explores the solution space by accepting probabilistic transitions to higher-energy states, allowing it to escape local optima. Nevertheless, SA requires careful temperature scheduling and may be computationally demanding for large-scale scheduling problems [56].

### 2.3.3 Swarm intelligence based

Swarm-based optimization algorithms have emerged as a powerful approach for solving complex optimization problems by drawing inspiration from the collective behavior of social insect colonies. These algorithms mimic the decentralized decision-making and self-organization principles

observed in nature, enabling them to navigate large search spaces and find optimal solutions effectively. The key idea behind swarm-based optimization lies in the interaction and cooperation among a population of agents, referred to as particles, bees, or ants, as they iteratively explore and exploit the search space. Through local communication and global information exchange, the swarm collectively converges toward promising regions of the search space, gradually refining the solutions. Examples of popular swarm-based optimization algorithms include Particle Swarm Optimization (PSO) [57], Ant Colony Optimization (ACO) [58], Bat Algorithm (BA) [59], Artificial Bee Colony (ABC) [60], Whale Optimization Algorithm (WOA) [61], Cat Swarm Optimization (CSO) [62], Firefly Algorithm (FA) [63], Grey Wolf Optimization (GSO) [64], Crow Search [65], Cuckoo Search [66], Glowworm Swarm Optimization (GSO) [67], Wild Horse Optimization (WHO) [68], Symbiotic Organism Search (SOS) [69], Chaotic Social Spider Optimization (CSSO) [70], Monkey Search Optimization (MSO) [71], Sea Lion Optimization (SLO) [72], and Virus Optimization Algorithm (VOA) [73]. These algorithms have impressive capabilities in various domains, including function optimization, data clustering, and task scheduling. Their ability to handle non-linear and dynamic problems and their parallel and distributed nature make swarm-based optimization algorithms a promising tool for tackling complex optimization challenges in diverse fields. This review paper mainly focuses on swarm intelligence algorithms in task scheduling of cloud computing.

### 3 Swarm intelligence algorithms in task scheduling of cloud computing

In this section, some swarm intelligence algorithms that are used in task scheduling of cloud computing are briefly summarized. Figure 2 represents the swarm intelligence algorithms focused in this paper.

#### 3.1 Particle swarm optimization

Particle swarm optimization was developed by Eberhart and Kennedy, drawing inspiration from the collective behavior of particles, such as flocking birds [74]. In PSO, particles adjust their paths based on their optimal position and the optimal position of the best particle in the population for each generation. The initial positions and speeds of the particles are randomly initialized before the population is created [75–79]. Researchers have applied PSO and other techniques to address job searching and employment balancing. In one study [76], a Variable Neighborhood Search (VNS) approach is employed to improve existing particle placements, while

an elite tabu search mechanism is used for local search to generate an initial set of solutions. Another study [75] utilizes hill-climbing strategy to enhance local search capabilities and mitigate premature convergence in PSO. The combination of Discrete PSO and the Min–Min approach [80] aims to reduce execution time for scheduling activities on computational grids. Furthermore, a combination of PSO and Gravitational Emulation Local Search (GELS) is proposed to enhance search space exploration [81]. Hybrid Particle Swarm Optimization (HPSO) is utilized for task scheduling to minimize turnaround time and improve resource efficiency [82]. Performance evaluation is typically based on makespan and resource consumption metrics.

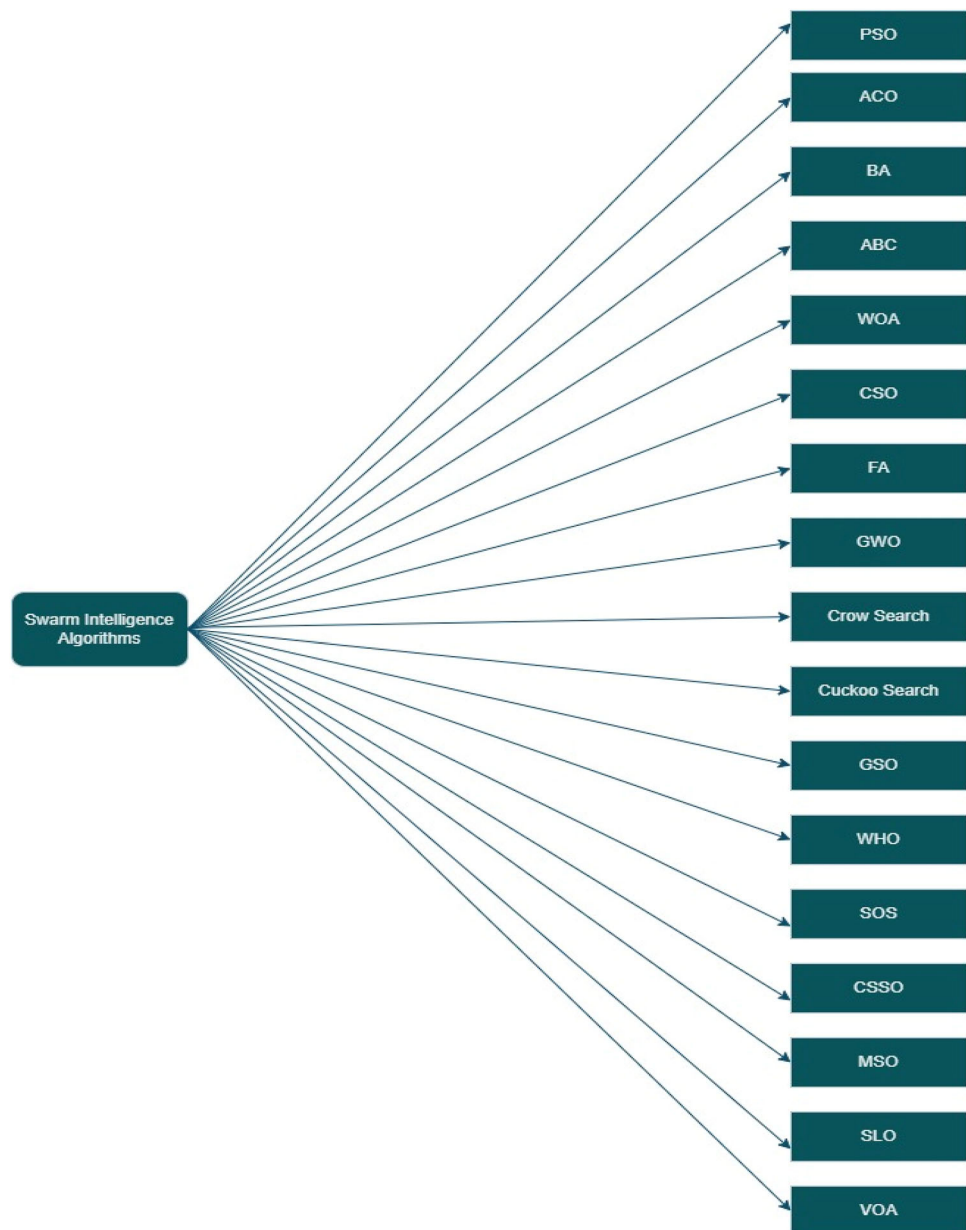
In a research paper [83], Tabu Search (TS) is combined with PSO to incorporate a local search mechanism. This integration enhances completion time and resource utilization. Another study [84] combines PSO with Cuckoo Search as a local search approach, reducing completion time and improving resource utilization. PSO has also been applied to task scheduling in grid environments [85, 86], where the particle population is initially randomly seeded to maximize completion time and resource utilization [86]. A PSO-based hyperheuristic for resource scheduling in grid contexts is proposed in [87], aiming to reduce time and cost while optimizing resource utilization. In [88], a load rebalancing algorithm utilizing PSO and the least position value technique is employed for task scheduling. The method's performance regarding makespan and average resource utilization in homogeneous and heterogeneous computing environments is evaluated. TBSLB-PSO (Task-based System Load Balancing method) is suggested in [89], where PSO reduces transfer and task execution times. Improved makespan and resource utilization are demonstrated in [90]. Furthermore, the energy consumption reduction of 67.5% is achieved by incorporating the particle swarm optimized tabu search mechanism (PSOTBM) [91].

#### 3.2 Ant colony optimization

The algorithm known as ant colony optimization was developed by Marco Dorigo, drawing inspiration from the foraging behavior observed in various ant species. In this algorithm, ants deposit pheromones on the ground to guide their fellow ants in following specific paths, and this behavior can be leveraged to solve optimization problems. Artificial ants act as agents navigating a solution space to find the best solutions. Pheromone values are used to explore the solution space, and the ants keep track of their positions and the quality of their solutions to identify an optimal solution [92]. In a research paper [93], the ACO technique is proposed for specifying task and resource selection criteria in clusters. Another study [94] introduces QoS limitations to achieve the desired quality in scheduling workflows. An improved version of the Ant Colony System (ACS)-based workflow



**Fig. 2** Swarm intelligence algorithms of the proposed study



scheduling algorithm is presented in [95], aiming to minimize costs while meeting deadlines. The ACO approach addresses grids' time-varying workflow scheduling problem to minimize overall costs within the given time constraint [96]. The ACO algorithm and the knowledge matrix concept are combined in [97] to track the historical desirability of placing tasks on the same physical machine. Energy-efficient scheduling algorithms based on ant colony behavior are proposed in [98], ensuring compliance with SLA throughput and response time restrictions. Updated pheromone schemes are presented in [99, 100], while [101] introduces a job scheduling approach for grids that adaptively adjusts the pheromone values to reduce execution time and improve

convergence rate. The population creation in [101] considers tasks' expected time and standard deviation, using the concept of biased starting ants. Additionally, [102] used an infinite number of ants for scheduling interdependent jobs in cloud-based environments, considering dependability, time, cost, and QoS constraints. Independent work scheduling for cloud computing systems based on the ACO approach is suggested in [103], optimizing makespan and comparing results with FCFS and RR scheduling algorithms. Furthermore, a modified version of ACO, known as the Multiple Pheromone Algorithm (MPA), exhibits advantages such as shorter makespan, lower cost, and increased dependability compared to conventional ACO and GA algorithms. Lastly, [104] proposes workflow scheduling for grid systems using

ACO, aiming to minimize the schedule length, and [105] considers both deadline and cost factors when scheduling workflows in hybrid clouds.

### 3.3 Bat algorithm

The bat algorithm is a swarm-based method that emulates the echolocation activity of bats [106]. Different aspects of bats' hunting techniques are associated with distinct sound pulses. Some bats rely on their sharp eyesight and keen sense of smell to quickly locate their targets. In resource scheduling, the BA algorithm has been utilized in cloud computing to optimize makespan more effectively than the genetic Algorithm [107]. Another study proposes a hybrid technique combining BA and harmony search for work scheduling in cloud computing [108]. The Gravitational Scheduling Algorithm (GSA) [109] is an extension of BA that incorporates time restrictions and a trust model, selecting resources for task mapping based on their trust values. The BA algorithm has been applied in a cloud environment to address workflow scheduling problems, effectively reducing execution costs compared to the best resource selection algorithms [110]. The paper [111] presents a combination of PSO and the Bat Algorithm for cloud profit maximization.

### 3.4 Artificial bee colony

The artificial bee colony algorithm, introduced by Dervis Karaboga in 2005, draws inspiration from the intelligent foraging behavior of honey bees and aims to address real-world problems [112]. In an ABC algorithm, bees in a colony collaborate to locate food sources, and this knowledge is utilized to guide decision-making during the search for optimal solutions. The colony consists of working bees, observers, and scout bees. Working bees continue searching for new food sources until they discover enough nectar to replace their current source. In an optimization problem, the food source represents a solution, and the amount of nectar corresponds to the quality of that solution. ABC has been successfully applied to various combinatorial problems, such as flow shop scheduling [113], on-shop scheduling [114], project scheduling [115], and traveling salesman problems [116]. Researchers have also employed bee colony optimization techniques for task scheduling in distributed grid systems, using bees' foraging behavior as a model for task mapping on available resources [117, 118]. Load balancing in non-preemptive independent task scheduling has been approached using ABC optimization [119, 120], resulting in improved resource utilization compared to the Min–Min algorithm by an average of 5.0383% when combined with particle swarm optimization [121]. ABC has been integrated with a memetic algorithm to reduce makespan and balance load [122]. The ABC algorithm has been applied in cloud

computing to schedule various tasks [123–125]. Its distinct characteristics, such as modularity and parallelism, make it suitable for scheduling dependent tasks in a cloud context [126, 127]. Moreover, energy-aware scheduling techniques have been suggested to manage resources and effectively enhance their utilization in the cloud [128].

### 3.5 Whale optimization algorithm

The whale optimization algorithm is an innovative method for addressing optimization problems inspired by the hunting behavior of humpback whales. This algorithm utilizes three operators that mimic the hunting strategies employed by whales: searching for prey, circling prey, and utilizing bubble nets [129]. Humpback whales are known to feed on krill or small fish in schools near the surface, employing a unique foraging technique involving the formation of bubbles along circular or '9'-shaped courses. By optimizing task allocation to resources, the WOA algorithm can reduce the makespan and improve the overall performance of cloud computing systems. Mangalampalli et al. introduced a Multi-objective Trust-Aware Scheduler with Whale Optimization (MOTSWO), which prioritizes jobs and virtual machines based on trust factors and schedules them to the most suitable virtual resources while minimizing time and energy consumption [130]. This approach significantly improves makespan, energy usage, overall running time, and trust factors such as availability, success rate, and turnaround efficiency. Another task-scheduling technique presented in [131] assigns tasks to appropriate virtual machines by determining task and virtual machine priorities. The WOA is used to model this technique, aiming to reduce data center energy usage and electricity costs. The W-Scheduler algorithm proposed in [132] builds upon a multi-objective model and the WOA. It calculates RAM and CPU cost functions to determine fitness values and optimizes jobs to virtual machines, minimizing costs, and makespan. Experimental results demonstrate that W-Scheduler outperforms existing techniques, achieving a minimum makespan of 7.0 and a minimum average cost of 5.8, effectively scheduling tasks to virtual machines.

An Improved WOA for Cloud Task Scheduling (IWC) is proposed to enhance the WOA's capacity for finding optimal solutions [133]. IWC demonstrates superior convergence speed and accuracy through simulation-based studies and comprehensive implementation compared to existing metaheuristic algorithms searching for optimal task-scheduling plans. Another variant of IWC is presented in [134]. The Vocalization of the Humpback Whale Optimization Algorithm (VWOA) is a new metaheuristic optimization technique proposed in [135]. VWOA is applied to optimize task scheduling in cloud computing environments, mimicking

the vocalization behavior of humpback whales. The proposed multi-objective model serves as the foundation for the VWOA scheduler, reducing time, cost, and energy consumption, while maximizing resource utilization. Experimental results on tested data show that the VWOA scheduler outperforms conventional WOA and Round Robin algorithms regarding makespan, cost, degree of imbalance, resource utilization, and energy consumption.

### 3.6 Cat swarm optimization

The cat swarm optimization algorithm, introduced by the author [136], is designed for continuous optimization problems. Inspired by the social behavior of cats, this heuristic algorithm leverages the seeking and tracking behavior modes exhibited by cats. Cats operate in seeking mode when they are cautious and move slowly, while they switch to tracing mode when they detect prey and begin pursuing it with more incredible speed. These behavioral patterns are employed in modeling optimization problems, with the cats' locations representing the solutions. The superiority of CSO over Particle Swarm Optimization (PSO) has been discussed by researchers [137]. Various modified versions of CSO have been proposed in [138, 139] to address discrete optimization problems in different domains. A binary variant of CSO, DBCSO, has been developed to solve the zero–one knapsack problem and the traveling salesperson problem [140]. In cloud computing, CSO has been utilized for workflow scheduling considering single and multiple objectives [141]. Optimization criteria such as makespan, computation expense, and CPU idle time are considered for mapping-dependent tasks. In population initialization, CSO and DBCSO are combined with the traditional genetic algorithm [142]. This hybrid approach, known as hybrid cat swarm optimization, aims to minimize makespan scheduling scientific applications using a cloud simulator, outperforming PSO, and binary PSO.

### 3.7 Firefly algorithm

The firefly algorithm is a nature-inspired optimization method that draws inspiration from the flashing behavior of fireflies [143]. This population-based algorithm emulates the flashing patterns of fireflies to determine the best solutions. In FA, randomly generated solutions are treated as fireflies, and their brightness is assigned based on their performance on the objective function. Fireflies move randomly if no brighter fireflies are nearby, but they are attracted to brighter fireflies if they exist. An efficient trust-aware task-scheduling algorithm using firefly optimization has been proposed in [144], demonstrating significant improvements over conventional approaches in minimizing makespan, increasing availability, success rate, and turnaround efficiency. Another study

[145] presents an acceptable enhancement in makespan and resource utilization using FA. The Crow Search algorithm is integrated with FA to enhance global search capability [146]. A hybrid approach combining firefly and genetic algorithms is proposed for task scheduling [147]. Furthermore, an intelligent meta-heuristic algorithm based on the combination of the Improved Cuckoo Search Algorithm (ICA) and FA is presented in [148], showcasing significant improvements in makespan, CPU time, load balancing, stability, and planning speed. A hybrid method combining Firefly and Simulated Annealing (SA) algorithms is proposed in [149]. In [150], the cat swarm optimization algorithm is combined with FA to create a hybrid multi-objective scheduling algorithm. In the context of Distributed Green Data Centers (DGDCs), Ammari et al. prioritize delay-bounded applications and efficiently schedule multiple heterogeneous applications while considering energy and cost optimization and ensuring compliance with delay-bound constraints [151]. They employ a modified Firefly Algorithm (mFA) to construct and optimize the operational cost minimization problem for DGDCs, resulting in successful optimization outcomes.

### 3.8 Gray wolf search optimization

The gray wolf optimization algorithm is a metaheuristic optimization algorithm inspired by the social interactions of gray wolves [64]. Gray wolves hunt in packs, with an alpha wolf leading the pack and beta, delta, and omega wolves following. The GWO algorithm models this hunting behavior to optimize various problems. In the cloud computing domain, the Performance Cost Gray Wolf Optimization (PCGWO) algorithm is proposed by Natesan et al. [152]. This algorithm optimizes resource and task allocation in cloud computing environments. The Modified Fractional Gray Wolf Optimizer for Multi-Objective Task Scheduling (MFGMTS) is presented in [153] as a multi-objective optimization technique. It employs the epsilon-constraint and penalty cost functions to compute objectives such as execution time, execution cost, communication time, communication cost, energy consumption, and resource usage. Gray Wolf Optimizer is also employed in [154]. Hybrid approaches have been proposed to enhance the performance of the GWO algorithm. The Genetic Gray Wolf Optimization Algorithm (GGWO) combines GWO with Genetic Algorithm [155]. It evaluates the algorithm's performance based on minimum computation time, migration cost, energy consumption, and maximum load utilization. In [156], a mean GWO algorithm is developed to improve task-scheduling system performance in heterogeneous cloud environments. A multi-objective GWO technique is introduced in [157] for task scheduling to optimize cloud resources while minimizing data center energy consumption and overall makespan. An enhanced version of



the GWO algorithm called IGWO is proposed by Mohammadzadeh et al. [158] to expedite convergence and avoid local optima. It incorporates the hill-climbing approach and chaos theory. The PSO–GWO algorithm, combining particle swarm optimization and gray wolf optimization, is proposed in [159]. Experimental results demonstrate that the PSO–GWO algorithm outperforms traditional particle swarm optimization and gray wolf optimization techniques, reducing average total execution cost and time.

### 3.9 Crow search algorithm

The crow search algorithm is a novel swarm intelligence optimization algorithm that imitates the intelligent behavior of crows in finding and hiding food [160]. Crows are known for their remarkable cognitive abilities, including remembering human faces and the location of hidden food. They exhibit cooperative behavior in food foraging, following each other to find better food sources. However, if a crow realizes it is being followed, it will relocate its food to prevent theft. This behavior has inspired the development of the crow search algorithm for solving optimization problems.

In the cloud computing task-scheduling field, the crow search algorithm has been applied to reduce makespan and select suitable virtual machines for jobs [161]. Experimental evaluations using CloudSim demonstrate the effectiveness of the crow search algorithm compared to the Min–Min and Ant Colony Optimization (ACO) algorithms. An Enhanced Crow Search Algorithm (ECSA) is proposed to improve the random selection of tasks [162]. Singh et al. suggest the Crow-Penguin Optimizer for Multi-objective Task Scheduling Strategy in Cloud Computing (CPO-MTS), which efficiently utilizes tasks and cloud resources to achieve the best completion results and exhibits a higher convergence rate toward global optima [163]. In a multi-objective task scheduling environment, Singh et al. propose the Crow Search-based Load Balancing Algorithm (CSLBA) [164]. This algorithm focuses on allocating the most suitable resources for tasks while considering factors such as Average Makespan Time (AMT), Average Waiting Time (AWT), and Average Data Center Processing Time (ADCPT).

Load balancing across virtual machines is a fundamental problem in cloud deployment, and a Crow Search-based load balancing method is introduced to address this issue [165]. This approach optimizes task resource mapping by considering average data center power consumption, cost, and load factors. The Gray Wolf Optimization and Crow Search Algorithm (GWO–CSA) is proposed to enhance the resource allocation model [166]. The crow search algorithm is combined with the Sparrow Search Algorithm (SSA) to minimize energy consumption in cloud computing environments [167]. Mangalampalli et al. propose a multi-objective task scheduling method using the crow search algorithm to

schedule tasks to appropriate virtual machines while considering the cost per energy unit in data centers [168].

#### 3.9.1 Cuckoo search algorithm

The cuckoo search algorithm is a recent optimization algorithm that takes inspiration from the brood parasitism behavior of certain cuckoo species [169]. These birds lay their eggs in the nests of other host birds, ensuring their eggs have a higher chance of survival. The cuckoo search algorithm simplifies this process into three idealized rules. Each nest represents a potential solution; a cuckoo can only lay one egg at a time. Cuckoos search for the best nests to lay their eggs, favoring nests that resemble the host bird's (close to optimal solutions). Lower-quality eggs (worse solutions) are either discarded or lead to the abandonment of the nest. Lévy flights are random walks with a heavy-tailed distribution and are used to select nests for new cuckoo eggs. The algorithm has been proposed in [170, 171] for optimizing task scheduling in cloud computing.

The cuckoo search algorithm has been combined with other optimization techniques to enhance its performance. The Cuckoo Search and Particle Swarm Optimization (CPSO) algorithms are merged in [172] to reduce makespan, cost, and deadline violation rates in task scheduling. The Cuckoo Crow Search method (CCSA) is introduced in [173] as a practical hybridized scheduling approach to find suitable virtual machines for task scheduling. In [174], a cuckoo search-based task scheduling method is proposed to efficiently allocate tasks among available virtual machines while maintaining low overall response times (QoS). For resource management in the Smart Grid, a load-balancing strategy based on the cuckoo search algorithm is suggested in [175], which involves identifying and turning off under-utilized virtual machines.

Furthermore, modifications and hybridizations of the cuckoo search algorithm have been proposed for various resource scheduling problems. The Standard Deviation-based Modified Cuckoo Optimization Algorithm (SDMCOA) [176] efficiently schedules tasks and manages resources. The Multi-objective Cuckoo Search Optimization (MOCSO) algorithm [177] addresses multi-objective resource scheduling problems in IaaS cloud computing. The Oppositional Cuckoo Search Algorithm (OCSA) [178], which combines cuckoo search with oppositional-based learning, offers a new hybrid algorithm. The Hybrid Gradient Descent Cuckoo Search (HGDCS) algorithm [179] combines the gradient descent approach with the cuckoo search algorithm for resource scheduling in IaaS cloud computing. The CHSA algorithm [180], combined with the cuckoo search and harmony search algorithms, optimizes the scheduling process. A group technology-based model and the cuckoo search algorithm are proposed for resource allocation [181].

A hybridized optimization algorithm that combines the Shuffled Frog Leaping Algorithm (SFLA) and Cuckoo Search Algorithm (CS) is suggested for resource allocation [182].

### 3.9.2 Glowworm swarm optimization

The glowworm swarm optimization method is a relatively recent swarm intelligence system that simulates the movement of glowworms in a swarm based on the distance between them and the presence of luciferin, a luminous substance [183]. It offers an effective approach to optimization problems. In [184], a Hybrid Glowworm Swarm Optimization (HGSO) algorithm is proposed to improve GSO for more efficient scheduling with affordable costs. This hybrid method incorporates evolutionary computation, quantum behavior strategies based on the neighborhood principle, offspring production, and random walk. The HGSO algorithm enhances convergence speed, facilitates escaping local optima, and reduces unnecessary computation and dependence on GSO initialization. Experimental results and statistical analysis demonstrate that the HGSO algorithm outperforms previous heuristic algorithms for most tasks. GSO is also applied to address task scheduling issues in cloud computing [185]. The GSO-based Task Scheduling (GSOTS) method reduces the overall cost of job execution while ensuring timely task completion. Simulation results show that the GSOTS method outperforms other scheduling algorithms, such as Shortest Task First (STF), Largest Task First (LTF), and Particle Swarm Optimization (PSO) in terms of reducing overall completion time and task execution costs.

### 3.9.3 Wild horse optimization algorithm

The wild horse optimizer is a recently developed metaheuristic algorithm that takes inspiration from the social dynamics observed among wild horses in their natural habitat. While WHO shows promising performance compared to specific algorithms, it faces challenges related to its exploitation capabilities and the potential for becoming trapped in local optima. The algorithmic model of WHO is shaped by the social interactions observed within wild horse populations. Stallions and the rest of the horse herd play distinct roles in these populations. WHO is designed to address various problems by leveraging collective behaviors such as group dynamics, grazing patterns, mating practices, dominance hierarchies, and leadership dynamics exhibited by the wild horse population. The paper [186] introduces an enhanced version of WHO called the Improved Wild Horse Optimizer (IWHO). The IWHO algorithm addresses the limitations of WHO by incorporating three improvements: the Random Running Strategy (RRS), Competition for Water Hole Mechanism (CWHM), and Dynamic Inertia Weight Strategy (DIWS). These enhancements aim to improve exploitation

capabilities and mitigate the issue of stagnation in local optima.

### 3.9.4 Symbiotic organism search

The symbiotic organism search optimization technique is a nature-inspired approach that can be applied to cloud computing task scheduling. This algorithm is based on organisms coexisting and utilizing each other's strengths to survive. In SOS, organisms represent potential solutions to the task scheduling problem, and their fitness is evaluated based on their ability to meet requirements while minimizing resource consumption and completion time.

The Discrete Symbiotic Organism Search (DSOS) technique is introduced in [69] specifically for optimal task scheduling on cloud resources. DSOS is well suited for large-scale scheduling problems as it exhibits faster convergence with increasing search size. However, DSOS can sometimes get trapped in local optima due to the significance of parameters such as makespan and response time. To address this, a faster convergent technique called enhanced Discrete Symbiotic Organism Search (eDSOS) is proposed in [187], particularly effective for more extensive or diverse search spaces. The chaotic symbiotic organisms search (CMSOS) technique was developed in [188] to tackle multi-objective large-scale task scheduling optimization problems in the IaaS cloud computing environment. CMSOS employs chaotic instead of random sequences to generate the initial population, enhancing the search process. In [189], the Adaptive Benefit Factors-based Symbiotic Organisms Search (ABF-SOS) method is presented, which balances local and global search techniques for faster convergence. This approach also incorporates an adaptive constrained handling strategy to efficiently adjust penalty function values, preventing premature convergence and impractical solutions. Energy-aware task scheduling in cloud environments is addressed by the Energy-aware Discrete Symbiotic Organism Search (E-DSOS) Optimization algorithm proposed in [190]. This algorithm focuses on optimizing energy consumption while performing task scheduling. A modified symbiotic organisms search algorithm called G\_SOS is suggested in [191] to reduce task execution time, cost, reaction time, and degree of imbalance. G\_SOS aims to accelerate convergence toward an ideal solution in the IaaS cloud environment.

### 3.9.5 Chaotic social spider optimization

The rapid growth of cloud computing has presented challenges in efficient task scheduling and resource utilization. Several metaheuristic algorithms have been proposed to address these challenges. One such algorithm is the chaotic social spider optimization algorithm, introduced in [70],

inspired by spiders' social behavior. This algorithm minimizes the makespan and balance load by incorporating swarm intelligence and chaotic inertia weight-based random selection. Simulations conducted in the study show that the proposed algorithm outperforms other swarm intelligence-based algorithms.

Another algorithm proposed in [192] is the Chaotic Particle Swarm Optimization (CPSO) algorithm, designed to improve the limitations of the standard particle swarm algorithm. The CPSO algorithm incorporates a chaotic sequence in the initialization process to enhance diversity and employs a diagnosis mechanism to identify premature convergence. Chaotic mutation is used to help particles escape from local optima. Simulation experiments demonstrate the feasibility and effectiveness of CPSO for task scheduling in cloud computing environments. For the multi-objective task scheduling problem in cloud computing, the Chaotic Symbiotic Organisms Search (CMSOS) algorithm is proposed in [188]. CMSOS utilizes a chaotic optimization strategy to generate an initial population and applies chaotic sequences and local search strategies to achieve global convergence and avoid local optima. The performance of CMSOS is evaluated using the CloudSim simulator, and the results demonstrate significant improvements in optimal trade-offs between execution time and financial cost.

### 3.9.6 Monkey search optimization algorithm

The monkey search optimization algorithm is a nature-inspired optimization algorithm that draws inspiration from the behavior of monkeys in search and foraging activities. The algorithm aims to solve optimization problems by mimicking the intelligent search patterns observed in monkeys. This paper [193] introduces a novel algorithm called Spider Monkey Optimization Inspired Load Balancing (SMO-LB) for load balancing. The algorithm is based on mimicking the foraging behavior of spider monkeys. Its primary goal is to balance the load among VMs, improving performance by reducing makespan (the total time to complete all tasks) and response time.

### 3.9.7 Sea lion optimization algorithm

Sea lions are known for their efficient foraging strategies, where they balance exploration and exploitation to find food sources in their natural habitats. The SLO algorithm mimics this behavior to search for the optimal solution in an optimization problem. In SLO, the potential solutions are represented as a population of sea lions. Each sea lion represents a candidate solution, and its fitness value reflects its performance on the objective function. The algorithm uses a combination of exploration and exploitation strategies to navigate the solution space. In this paper [194], the sea lion

optimization algorithm is proposed for task scheduling in cloud computing. The performance of SLO is compared to other algorithms, including vocalization of whale optimization algorithm, whale optimization algorithm, gray wolf optimization, and round robin. SLO outperforms the other algorithms in terms of makespan and imbalance degree.

### 3.9.8 Virus optimization algorithm

The virus optimization algorithm is a nature-inspired optimization algorithm that draws inspiration from the behavior of viruses in their quest for survival and replication. This algorithm mimics viruses' adaptive and evolutionary mechanisms to solve complex optimization problems. The VOA represents a population of candidate solutions as a set of viruses. Each virus contains a genetic code representing a potential solution to the problem. The algorithm iteratively evolves the population by applying genetic operators such as mutation, recombination, and selection to improve the quality of solutions. The key concept behind VOA is the idea of infection and spreading. Viruses can infect and influence other viruses in the population, leading to the exchange of genetic information and the exploration of new regions in the search space. This spreading mechanism promotes exploration and exploitation of solutions, enabling the algorithm to escape local optima and find better solutions. This paper [195] introduces a novel two-objective virus optimization algorithm designed for task mapping in cloud computing environments. The objectives considered in this algorithm are the makespan and the cost associated with assigning resources. Building upon the foundation of the genetic algorithm, the VOA proposes redefined parameters to enhance the sorting ability between virus infection strategies. This modification aims to improve the algorithm's efficiency in achieving high-quality solutions by effectively balancing the trade-off between makespan and cost.

## 4 Primary objectives of swarm-based algorithms

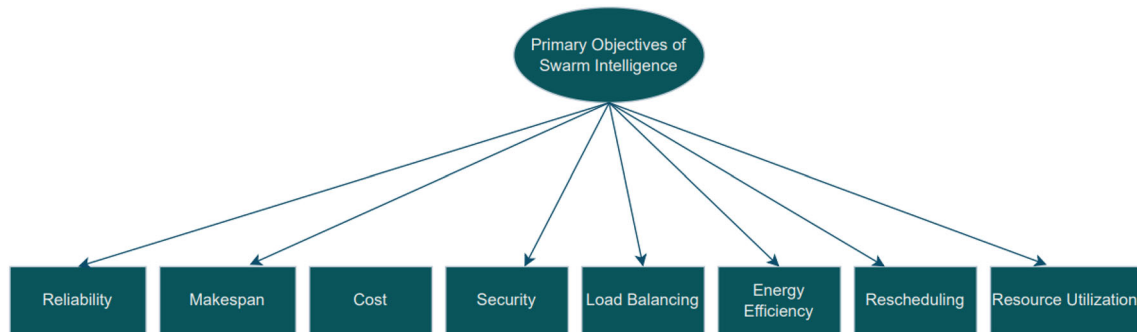
This study assesses comparative analysis among various swarm intelligence algorithms based on primary objectives. Table 1 represents the categorizations of swarm intelligence algorithms based on these criteria. Figure 3 depicts the primary objectives of swarm-based algorithms. Each criterion holds its importance in evaluating the effectiveness of the algorithms in task scheduling for cloud computing systems. These criteria are:

**Table 1** Primary objectives of swarm intelligence algorithms

	Reference	Reliability	Makespan	Cost	Security	Load balance	Energy efficient	Rescheduling	Resource utilization
PSO	[75]	–	Yes	–	Yes	–	–	–	–
	[77]	–	Yes	Yes	Yes	Yes	–	–	–
	[76]	–	Yes	Yes	–	–	–	–	–
	[81]	–	–	Yes	–	–	–	–	–
	[82]	–	Yes	Yes	–	–	–	–	–
	[91]	–	Yes	Yes	–	–	–	–	–
ACO	[94]	–	Yes	–	–	Yes	–	–	–
	[95]	–	Yes	–	–	–	–	–	–
	[96]	–	Yes	Yes	–	–	–	–	–
	[97]	–	Yes	Yes	–	–	–	–	–
	[98]	–	Yes	–	–	–	–	–	–
BA	[106]	–	Yes	Yes	–	–	–	–	–
	[108]	–	Yes	–	–	–	–	–	–
	[110]	–	Yes	Yes	–	–	–	–	–
ABC	[113]	–	Yes	–	–	–	–	–	–
	[114]	–	Yes	–	–	–	–	–	–
	[115]	–	–	–	–	Yes	Yes	–	Yes
	[116]	–	–	–	–	Yes	–	–	–
	[117]	–	–	–	–	Yes	–	–	Yes
WOA	[121]	–	Yes	Yes	–	–	–	–	–
	[122]	–	Yes	Yes	–	–	–	–	–
	[123]	–	Yes	Yes	–	–	Yes	–	Yes
	[124]	–	–	Yes	–	–	Yes	–	–
	[125]	–	Yes	Yes	–	–	–	–	–
CSO	[140]	–	–	Yes	–	Yes	–	–	–
	[141]	–	–	Yes	–	–	–	–	–
FA	[144]	–	Yes	Yes	–	–	–	–	–
	[145]	–	Yes	Yes	–	Yes	–	–	–
	[146]	–	Yes	–	–	–	–	–	–
GWO	[152]	–	Yes	Yes	–	Yes	Yes	–	–
	[153]	–	Yes	Yes	–	–	–	–	–
	[154]	–	Yes	Yes	–	–	–	–	–
	[155]	–	Yes	Yes	–	–	Yes	–	–
Crow search	[161]	–	Yes	–	–	–	Yes	–	–
	[162]	–	Yes	Yes	–	Yes	–	–	Yes
	[163]	–	–	Yes	–	–	Yes	–	–
	[164]	–	Yes	–	–	–	–	–	–
	[165]	–	–	–	–	–	–	–	Yes
Cuckoo search	[170]	–	–	–	–	Yes	–	–	–
	[171]	–	Yes	–	–	–	–	–	–
	[172]	–	Yes	Yes	–	Yes	–	–	–
	[173]	–	Yes	Yes	–	–	–	–	–
	[174]	–	Yes	–	–	–	–	–	–

**Table 1** (continued)

	Reference	Reliability	Makespan	Cost	Security	Load balance	Energy efficient	Rescheduling	Resource utilization
	[175]	–	Yes	Yes	–	–	–	–	–
	[176]	–	Yes	–	–	Yes	–	–	–
GSO	[185]	–	Yes	Yes	–	–	–	–	–
WHO	[186]	–	Yes	Yes	–	–	Yes	–	–
SOS	[69]	–	Yes	–	–	–	–	–	–
	[187]	–	Yes	–	–	–	–	–	–
	[188]	–	–	–	–	–	Yes	–	–
CSSO	[70]	–	Yes	–	–	Yes	–	–	–
	[192]	–	Yes	Yes	–	–	–	–	–
	[188]	–	Yes	Yes	–	–	–	–	–
MSO	[193]	–	Yes	–	–	Yes	–	–	–
SLO	[194]	–	Yes	Yes	–	Yes	–	–	–
VOA	[195]	–	Yes	Yes	–	Yes	–	–	–

**Fig. 3** Primary objectives of swarm intelligence algorithm in task scheduling of cloud computing

#### 4.1 Reliability

Reliability measures the ability of an algorithm to consistently provide accurate and dependable task scheduling results, minimizing errors, failures, and system downtime.

#### 4.2 Makespan

This metric measures the total time to complete all tasks in a scheduling algorithm. It reflects the efficiency and speed of task execution.

#### 4.3 Cost

The cost criterion considers the financial implications associated with task scheduling algorithms. It encompasses resource usage costs, operational expenses, and optimizing resource allocation to minimize expenses.

#### 4.4 Security

This criterion assesses the level of protection the algorithms provide against potential security threats, such as unauthorized access, data breaches, and vulnerabilities in the cloud environment.

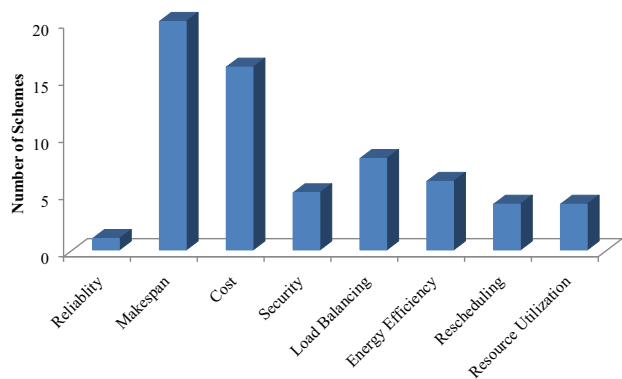
#### 4.5 Load balancing

Load balancing evaluates how effectively an algorithm distributes tasks among virtual machines, ensuring that workloads are evenly distributed and maximizes resource utilization. It aims to prevent overloading or under-utilization of specific resources.

#### 4.6 Energy efficiency

Energy efficiency focuses on minimizing the cloud computing system's power consumption and carbon footprint.





**Fig. 4** Frequency of primary objectives

Algorithms that optimize energy usage contribute to cost savings and environmental sustainability.

#### 4.7 Rescheduling

An algorithm can adapt and dynamically adjust task assignments based on changes in workload, resource availability, or system conditions. Efficient rescheduling minimizes disruptions and optimizes task execution.

#### 4.8 Resource Utilization

This criterion measures the efficiency of resource allocation in terms of CPU usage, memory utilization, network bandwidth, and storage capacity. Higher resource utilization signifies effective allocation and optimization of available resources.

The various scheduling objectives are evaluated by analyzing the algorithms, as depicted in Fig. 4. Among these objectives, makespan is the most commonly employed scheduling objective. Cost is identified as the second most significant factor. On the other hand, reliability is considered by only a limited number of studies, as indicated in Fig. 4.

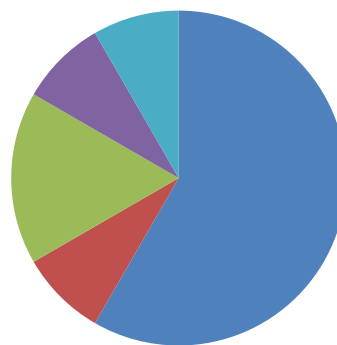
### 5 Simulation Tool

Conducting experiments using new strategies in a real-world setting is highly challenging due to the potential risks of negatively impacting end-user quality of services. To overcome this limitation, researchers have turned to various simulation tools to evaluate the effectiveness of novel scheduling algorithms in different cloud environments. Among these tools, the CloudSim simulation toolkit has gained significant popularity as a well-known platform for task scheduling. The toolkit offers pre-existing programmatic classes that can be extended to incorporate specific algorithms for evaluating a wide range of QoS parameters. These parameters

**Table 2** Simulation tools

	References	Swarm intelligence algorithm
GCC compiler	[196]	FA
CloudSim	[197]	PSO
	[198]	PSO
	[177]	CSO
	[199]	PSO
	[200]	PSO
Real environment	[198]	ACO and CSA
	[201]	PSO
Matlab	[202]	PSO
Java	[203]	ACO
	[204]	PSO

■ CloudSim ■ GCC compiler ■ Real environment ■ Matlab ■ Java



**Fig. 5** Usage measurement of simulation tools

include makespan, financial cost, computational cost, reliability, availability, scalability, energy consumption, security, throughput, and other constraints like deadline, priority, budget, and fault tolerance. In addition to CloudSim, other reputable simulation tools are available in the cloud, such as iCloud, GridSim, CloudAnalyst, NetworkCloudSim, WorkflowSim, and GreenCloud. These tools provide researchers with comprehensive options for testing and categorizing different techniques. A detailed breakdown of these techniques and their respective testing tools can be found in Table 2. The usage measurement of simulation tools for swarm intelligence algorithms is depicted in Fig. 5.

### 6 Challenges in swarm intelligence algorithms for task scheduling in cloud computing

Swarm intelligence algorithms have gained significant attention for task scheduling in cloud computing due to their

ability to handle complex optimization problems. However, several challenges are associated with applying swarm intelligence algorithms in this context. Here are some key challenges:

### 6.1 Scalability

As cloud computing environments scale up with many tasks and virtual machines, swarm intelligence algorithms need to handle the increased complexity and size of the problem efficiently.

### 6.2 Dynamic environment

Cloud computing environments are dynamic, with varying workloads, resource availability, and task requirements. Swarm intelligence algorithms should adapt to these changes quickly and provide effective task-scheduling solutions. Handling dynamic environments with evolving conditions is a challenge for swarm intelligence algorithms.

### 6.3 Multiple objectives

Task scheduling in cloud computing involves multiple conflicting objectives, such as minimizing makespan, cost, and energy consumption while maximizing resource utilization and load balancing. Swarm intelligence algorithms must address these objectives' trade-offs and find optimal or near-optimal solutions.

### 6.4 Resource heterogeneity

Cloud computing environments comprise diverse resources with varying capacities, capabilities, and costs. Swarm intelligence algorithms should consider resource heterogeneity while making task-to-resource assignments to optimize performance and efficiency.

### 6.5 Communication overhead

In swarm intelligence algorithms, communication among agents is crucial for information sharing and coordination. However, excessive communication can lead to high overhead and delays. Balancing the communication overhead and its impact on algorithm performance is a challenge.

### 6.6 Convergence speed

Swarm intelligence algorithms should converge to optimal or near-optimal solutions within a reasonable time frame. The Enhancement of convergence speeds while maintaining solution quality is a challenge, especially for large-scale task scheduling problems.

### 6.7 Security and privacy

Task scheduling algorithms in cloud computing must address security and privacy concerns. Ensuring secure communication, protecting sensitive data, and preventing unauthorized access or tampering with scheduling decisions are essential challenges.

### 6.8 Robustness and fault tolerance

Swarm intelligence algorithms should be robust against failures, uncertainties, and noisy environments. They should recover from failures and adapt to changes to maintain reliable task scheduling performance.

Addressing these challenges requires ongoing research and innovation in swarm intelligence algorithms. Overcoming these hurdles will enable more effective and efficient task scheduling in cloud computing, leading to improved resource utilization, reduced costs, enhanced system performance, and better service quality for cloud users.

## 7 Future prospects of swarm intelligence algorithms

As the field continues to evolve, several future directions and emerging trends are shaping the development and application of swarm intelligence algorithms in task scheduling. Some of these exciting directions are:

### 7.1 Hybridization and integration

One key future direction is combining and integrating swarm intelligence algorithms with other optimization techniques. Researchers are exploring combining swarm intelligence with genetic algorithms, ant colony optimization, particle swarm optimization, and other metaheuristic approaches to enhance the performance and robustness of task scheduling algorithms. Hybrid algorithms have the potential to leverage the strengths of different algorithms and overcome their limitations.

### 7.2 Multi-objective optimization

Task scheduling in cloud computing involves multiple conflicting objectives, such as minimizing makespan, optimizing resource utilization, reducing energy consumption, and ensuring load balancing. Future research will focus on developing swarm intelligence algorithms that can effectively handle these multi-objective optimization problems. Multi-objective swarm intelligence algorithms enable decision-makers to explore trade-offs among different objectives and find Pareto-optimal solutions.

### 7.3 Dynamic and real-time scheduling

Cloud computing environments constantly evolve, with varying workloads, resource availability, and user demands. Future swarm intelligence algorithms must adapt to dynamic scenarios and perform real-time scheduling. These algorithms should be capable of handling task arrivals, departures, failures, and resource fluctuations efficiently. Dynamic scheduling algorithms based on swarm intelligence will enable better resource allocation and load balancing in dynamic cloud environments.

### 7.4 Security and privacy

With the increasing importance of data security and privacy in cloud computing, future swarm intelligence algorithms will incorporate mechanisms to address these concerns. Researchers will explore techniques to ensure secure task scheduling, protect sensitive data during the scheduling process, and prevent unauthorized access to cloud resources. Cryptography techniques, secure communication protocols, and privacy-preserving mechanisms can enhance swarm intelligence algorithms to provide robust, secure task-scheduling solutions.

### 7.5 Edge and fog computing

The emergence of edge and fog computing paradigms presents new challenges and opportunities for task scheduling. Swarm intelligence algorithms must adapt to these distributed computing architectures and consider the unique characteristics and constraints of edge devices and fog nodes. Future research will focus on developing swarm intelligence-based scheduling algorithms that optimize resource allocation and task offloading in edge and fog computing environments.

### 7.6 Machine learning and artificial intelligence

The integration of machine learning and artificial intelligence techniques with swarm intelligence algorithms holds great potential for advancing task scheduling in cloud computing. Researchers are exploring reinforcement learning, deep learning, and neural networks to enhance the decision-making capabilities of swarm intelligence algorithms. These algorithms can adapt and optimize scheduling decisions over time by leveraging historical data and learning from past scheduling experiences.

The future of swarm intelligence algorithms in task scheduling for cloud computing is promising and offers numerous exciting research directions. These advancements will improve cloud computing system's performance, resource utilization, and scalability.

## 8 Conclusion

The rapid adoption of cloud computing infrastructure and distributed computing paradigms has enabled many scientific applications to migrate to the cloud. This transition offers numerous advantages, such as virtualization and shared resource pools, which allow for the management and execution of large-scale workflow applications without the need for physical computing infrastructure. However, to fully leverage the cloud's potential, optimizing resources accessible through cloud computing is essential. In recent years, swarm intelligence nature-inspired scheduling algorithms have emerged as a promising approach for optimizing task scheduling in the cloud. These algorithms have shown superior optimization results compared to traditional and heuristic approaches. While several studies have applied swarm intelligence optimization algorithms to solve scheduling problems, most have focused on a limited number of parameters, providing only an overview and state-of-the-art analysis.

To gain a deeper understanding of task scheduling techniques based on swarm intelligence optimization, it is crucial to simultaneously consider state-of-the-art, comparative analysis of performance metrics, research challenges, and future directions. A comprehensive and systematic assessment of cloud swarm intelligence optimization scheduling approaches is needed to keep pace with the continuous growth of nature-inspired algorithms. To the best of our knowledge, there is a need for a thorough and systematic evaluation that provides a comprehensive analysis of swarm intelligence optimization scheduling approaches in the cloud. Therefore, this review provides a comprehensive analysis of various swarm-based algorithms such as particle swarm optimization, ant colony optimization, bat algorithm, artificial bee colony, whale optimization algorithm, cat swarm optimization, firefly algorithm, gray wolf optimization, crow search, cuckoo search, glowworm swarm optimization, wild horse optimization, symbiotic organism search, chaotic social spider, monkey search optimization, sea lion optimization, and virus optimization algorithm. This study also focuses on highlighting their application in cloud task scheduling. The performance of these algorithms is also compared using various metrics such as reliability, makespan, cost, security, load balancing, rescheduling, energy efficiency, and resource utilization. The most active study areas are makespan and cost, which are concentrated by most studies. A few simulation tools used in cloud computing are briefly addressed and contrasted for implementing and testing new algorithms. ClouSim is the most used simulation tool among all the tools. It also identifies challenges in cloud computing and proposes future research directions. Overall, swarm-based algorithms have demonstrated their potential to enhance resource allocation, improve system performance, and maximize the utilization of cloud resources. As the field

continues to evolve, further exploration and innovation in swarm-based algorithms are expected to drive advancements in cloud computing.

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

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

**Data availability** The datasets generated during the current study are available from the corresponding author upon reasonable request.

## Declarations

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

## References

- Prity, F.S., Gazi, M.H., Uddin, K.M.: A review of task scheduling in cloud computing based on nature-inspired optimization algorithm. *Clust. Comput.* (2023). <https://doi.org/10.1007/s10586-023-04090-y>
- Mazumder, A.M.R., Uddin, K.A., Arbe, N., Jahan, L. and Whaiduzzaman, M., 2019. Dynamic task scheduling algorithms in cloud computing. In 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA) (pp. 1280–1286). IEEE
- Kumar, R., Bhagwan, J.: A comparative study of meta-heuristic-based task scheduling in cloud computing. In: Mohan, H.D. (ed.) Artificial intelligence and sustainable computing: Proceedings of ICSISCT 2020, pp. 129–141. Springer Singapore, Singapore (2022)
- Chowdhury, N., Aslam Uddin, M.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)
- Singh, H., Tyagi, S., Kumar, P.: Scheduling in cloud computing environment using metaheuristic techniques a survey. In: Mandal, J.K. (ed.) Emerging technology in modelling and graphics: Proceedings of IEM graph 2018, pp. 753–763. Springer Singapore, Singapore (2020)
- Kumar, D.D.: Review on task scheduling in ubiquitous clouds. *J. IoT Soc. Mobile Anal. Cloud* **1**(1), 72–80 (2019)
- Manikandan, N., Gopalakrishnan, N., Pradeep, K.: Bee optimization based random double adaptive whale optimization model for task scheduling in cloud computing environment. *Comput. Commun.* **187**, 35–44 (2022)
- Ghafari, R., Kabutarkhani, F.H., Mansouri, N.: Task scheduling algorithms for energy optimization in cloud environment: a comprehensive review. *Clust. Comput.* **25**(2), 1035–1093 (2022)
- Murad, S.A., Muzahid, A.J.M., Azmi, Z.R.M., Hoque, M.I., Kowsher, M.: A review on job scheduling technique in cloud computing and priority rule based intelligent framework. *J. King Saud Univ. Comput. Inform. Sci.* **34**(6), 2309–2331 (2022)
- Morton, T. and Pentico, D.W., 1993. Heuristic scheduling systems: with applications to production systems and project management (Vol. 3). John Wiley & Sons.
- 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)
- Saidi, K., Bardou, D.: Task scheduling and VM placement to resource allocation in Cloud computing: challenges and opportunities. *Clust. Comput.* (2023). <https://doi.org/10.1007/s10586-023-04098-4>
- Abdel-Basset, M., Mohamed, R., Abd Elkhaliq, W., Sharawi, M., Sallam, K.M.: Task scheduling approach in cloud computing environment using hybrid differential evolution. *Mathematics* **10**(21), 4049 (2022)
- Sharma, N., Garg, P.: Ant colony based optimization model for QoS-Based task scheduling in cloud computing environment. *Measure. Sens.* **24**, 100531 (2022)
- Cao, H.: The analysis of edge computing combined with cloud computing in strategy optimization of music educational resource scheduling. *Int. J. Syst. Assur. Eng. Manage.* **14**(1), 165–175 (2023)
- Hamid, L., Jadoon, A., Asghar, H.: Comparative analysis of task level heuristic scheduling algorithms in cloud computing. *J. Supercomput.* **78**(11), 12931–12949 (2022)
- Sissodia, R., Rauthan, M.S., Barthwal, V.: A multi-objective task scheduling approach using improved max-min algorithm in cloud computing. In: Buyya, R. (ed.) International Conference on Advanced Communications and Machine Intelligence, pp. 159–169. Springer Nature Singapore, Singapore (2022)
- Bacanin, N., Zivkovic, M., Bezdán, T., Venkatachalam, K., Abouhawwash, M.: Modified firefly algorithm for workflow scheduling in cloud-edge environment. *Neural Comput. Appl.* **34**(11), 9043–9068 (2022)
- Wu, W., Hayashi, T., Haruyasu, K., Tang, L.: Exact algorithms based on a constrained shortest path model for robust serial-batch and parallel-batch scheduling problems. *Eur. J. Oper. Res.* **307**(1), 82–102 (2023)
- Saomoto, H., Kikkawa, N., Moriguchi, S., Nakata, Y., Otsubo, M., Angelidakis, V., Cheng, Y.P., Chew, K., Chiaro, G., Duriez, J., Duverger, S.: Round robin test on angle of repose: DEM simulation results collected from 16 groups around the world. *Soils Found.* **63**(1), 101272 (2023)
- Sirisha, D.: Complexity versus quality: a trade-off for scheduling workflows in heterogeneous computing environments. *J. Supercomput.* **79**(1), 924–946 (2023)
- Kubiak, W.: A note on scheduling coupled tasks for minimum total completion time. *Ann. Oper. Res.* **320**(1), 541–544 (2023)
- Elshahed, E.M., Abdelmoneem, R.M., Shaaban, E., Elzahed, H.A., Al-Tabbakh, S.M.: Prioritized scheduling technique for healthcare tasks in cloud computing. *J. Supercomput.* **79**(5), 4895–4916 (2023)
- Chen, H., Wang, F., Helian, N., Akanmu, G., 2013 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
- George Amalarethinam, D.I., Kavitha, S.: Rescheduling enhanced Min-Min (REMM) algorithm for metatask scheduling in cloud computing. In: Hemanth, J. (ed.) International Conference on Intelligent Data Communication Technologies and Internet of Things, pp. 895–902. Springer International Publishing, Cham (2019)
- Mao, Y., Chen, X., Li, X.: Max–min task scheduling algorithm for load balance in cloud computing. In: Patnaik, S., Li, X. (eds.) In Proceedings International Conference on Computer Science and Information Technology, pp. 457–465. Springer India, New Delhi (2014)

27. Sandana Karuppan, A., Meena Kumari, S.A., Sruthi, S.: A priority-based max-min scheduling algorithm for cloud environment using fuzzy approach. In: Smys, S. (ed.) International Conference on Computer Networks and Communication Technologies ICCNCT 2018, pp. 819–828. Springer Singapore, Singapore (2019)
28. 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)
29. 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)
30. Nazar, T., Javaid, N., Waheed, M., Fatima, A., Bano, H., Ahmed, N.: Modified shortest job first for load balancing in cloud-fog computing. In: Barolli, L. (ed.) *Advances on Broadband and Wireless Computing, Communication and Applications: The 13th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA2018)*, pp. 63–76. Springer International Publishing, Cham (2019)
31. Alworafi, M.A., Dhari, A., Al-Hashmi, A.A. and Darem, A.B., 2016. An improved SJF scheduling algorithm in cloud computing environment. In 2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECOT). IEEE. pp. 208–212
32. 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)
33. 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>
34. 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)
35. 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)
36. Khalid, O.W., Isa, N.A.M., Sakim, H.A.M.: Emperor penguin optimizer: a comprehensive review based on state-of-the-art meta-heuristic algorithms. *Alex. Eng. J.* **63**, 487–526 (2023)
37. Faramarzi-Oghani, S., Dolati Neghabadi, P., Talbi, E.G., Tavakkoli-Moghaddam, R.: Meta-heuristics for sustainable supply chain management: a review. *Int. J. Prod. Res.* **61**(6), 1979–2009 (2023)
38. Afzal, A., Buradi, A., Jilte, R., Shaik, S., Kaladgi, A.R., Arıcı, M., Lee, C.T., Nižetić, S.: Optimizing the thermal performance of solar energy devices using meta-heuristic algorithms: a critical review. *Renew. Sustain. Energy Rev.* **173**, 112903 (2023)
39. Alhijawi, B., Awajan, A.: Genetic algorithms: theory, genetic operators, solutions, and applications. *Evol. Intell.* (2023). <https://doi.org/10.1007/s12065-023-00822-6>
40. Dong, J., Wang, H., Zhang, S.: Dynamic electric vehicle routing problem considering mid-route recharging and new demand arrival using an improved memetic algorithm. *Sustainable Energy Technol. Assess.* **58**, 103366 (2023)
41. Hussien, A.G., Heidari, A.A., Ye, X., Liang, G., Chen, H., Pan, Z.: Boosting whale optimization with evolution strategy and Gaussian random walks: An image segmentation method. *Engineering with Computers* **39**(3), 1935–1979 (2023)
42. Liew, S.H., Choo, Y.H., Low, Y.F., Nor Rashid, F.A.: Distraction descriptor for brainprint authentication modelling using probability-based Incremental Fuzzy-Rough Nearest Neighbour. *Brain informatics* **10**(1), 21 (2023)
43. Althoey, F., Akhter, M.N., Nagra, Z.S., Awan, H.H., Alanazi, F., Khan, M.A., Javed, M.F., Eldin, S.M., Özkılıç, Y.O.: Prediction models for marshall mix parameters using bio-inspired genetic programming and deep machine learning approaches: a comparative study. *Case Studies in Construction Materials* **18**, e01774 (2023)
44. Chakraborty, S., Saha, A.K., Ezugwu, A.E., Agushaka, J.O., Zitar, R.A., Abualigah, L.: Differential evolution and its applications in image processing problems: a comprehensive review. *Archives of Computational Methods in Engineering* **30**(2), 985–1040 (2023)
45. Sohail, A.: Genetic algorithms in the fields of artificial intelligence and data sciences. *Annal. Data Sci.* **10**(4), 1007–1018 (2023)
46. Materwala, H., Ismail, L., Hassanein, H.S.: QoS-SLA-aware adaptive genetic algorithm for multi-request offloading in integrated edge-cloud computing in Internet of vehicles. *Veh. Commun.* **43**, 100654 (2023)
47. Zhou, G., Tian, W., Buyya, R., Wu, K.: Growable Genetic Algorithm with Heuristic-based Local Search for multi-dimensional resources scheduling of cloud computing. *Appl. Soft Comput.* **136**, 110027 (2023)
48. Agarwal, G., Gupta, S., Ahuja, R., Rai, A.K.: Multiprocessor task scheduling using multi-objective hybrid genetic Algorithm in Fog-cloud computing. *Knowl.-Based Syst.* **272**, 110563 (2023)
49. Gupta, P., Rawat, P.S., kumar Saini, D., Vidyarthi, A. and Alharbi, M.: Neural network inspired differential evolution based task scheduling for cloud infrastructure. *Alex. Eng. J.* **73**, 217–230 (2023)
50. Karkinli, A.E.: Detection of object boundary from point cloud by using multi-population based differential evolution algorithm. *Neural Comput. Appl.* **35**(7), 5193–5206 (2023)
51. Nemoto, R.H., Ibarra, R., Staff, G., Akhiiardinov, A., Brett, D., Dalby, P., Casolo, S., Piebalgs, A.: Cloud-based virtual flow metering system powered by a hybrid physics-data approach for water production monitoring in an offshore gas field. *Digit Chem Eng* **9**, 100124 (2023)
52. Xiao, L., Fan, C., Ai, Z., Lin, J.: Locally informed gravitational search algorithm with hierarchical topological structure. *Eng. Appl. Artif. Intell.* **123**, 106236 (2023)
53. Wang, Q., Yu, D., Zhou, J., Jin, C.: Data storage optimization model based on improved simulated annealing algorithm. *Sustainability* **15**(9), 7388 (2023)
54. Elsedimy, E.I., AboHashish, S.M., Algarni, F.: New cardiovascular disease prediction approach using support vector machine and quantum-behaved particle swarm optimization. *Multimed. Tools Appl.* (2023). <https://doi.org/10.1007/s11042-023-16194-z>
55. Praveen, S.P., Ghasempoor, H., Shahabi, N., Izanloo, F.: A hybrid gravitational emulation local search-based algorithm for task scheduling in cloud computing. *Math. Probl. Eng.* **2023**, 1–9 (2023). <https://doi.org/10.1155/2023/6516482>
56. Zhang, Y., Han, C., Liu, S.: A digital calibration technique for N-channel time-interleaved ADC based on simulated annealing algorithm. *Microelectron. J.* **133**, 105701 (2023)
57. Jiang, Y., Hu, T., Huang, C., Wu, X.: An improved particle swarm optimization algorithm. *Appl. Math. Comput.* **193**(1), 231–239 (2007)
58. Dorigo, M., Stützle, T.: *Ant colony optimization: overview and recent advances*, pp. 311–351. Springer International Publishing, Cham (2019)
59. Yang, X.S., He, X.: Bat algorithm: literature review and applications. *Int. J. Bio-Inspir. Comput.* **5**(3), 141–149 (2013)
60. Karaboga, D., Gorkemli, B., Ozturk, C., Karaboga, N.: A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artif. Intell. Rev.* **42**, 21–57 (2014)
61. Gharehchopogh, F.S., Gholizadeh, H.: A comprehensive survey: Whale optimization algorithm and its applications. *Swarm Evol. Comput.* **48**, 1–24 (2019)



62. 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/10.1155/2020/4854895>
63. Yang, X.S., He, X.: Firefly algorithm: recent advances and applications. *Int. J. Swarm intel.* **1**(1), 36–50 (2013)
64. 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)
65. 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)
66. Yang, X.S., Deb, S.: Cuckoo search: recent advances and applications. *Neural Comput. Appl.* **24**, 169–174 (2014)
67. Wu, B., Qian, C., Ni, W., Fan, S.: The improvement of glowworm swarm optimization for continuous optimization problems. *Expert Syst. Appl.* **39**(7), 6335–6342 (2012)
68. Naruei, I., Keynia, F.: Wild horse optimizer: A new meta-heuristic algorithm for solving engineering optimization problems. *Engineering with Computers* **38**(Suppl 4), 3025–3056 (2022)
69. Abdullahi, M., Ngadi, M.A.: Symbiotic organism search optimization based task scheduling in cloud computing environment. *Futur. Gener. Comput. Syst.* **56**, 640–650 (2016)
70. Arul Xavier, V.M., Annadurai, S.: Chaotic social spider algorithm for load balance aware task scheduling in cloud computing. *Clust. Comput.* **22**(Suppl 1), 287–297 (2019)
71. Duque, F.G., de Oliveira, L.W., de Oliveira, E.J., Marcato, A.L., Silva, I.C., Jr.: Allocation of capacitor banks in distribution systems through a modified monkey search optimization technique. *Int. J. Electr. Power Energy Syst.* **73**, 420–432 (2015)
72. Nguyen, B.M., Tran, T., Nguyen, T., Nguyen, G.: An improved sea lion optimization for workload elasticity prediction with neural networks. *Int. J. Comput. Intell. Syst.* **15**(1), 90 (2022)
73. Liang, Y.C., Cuevas Juarez, J.R.: A self-adaptive virus optimization algorithm for continuous optimization problems. *Soft. Comput.* **24**, 13147–13166 (2020)
74. 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).
75. 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.
76. 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.
77. Xue, S.J., Wu, W.: Scheduling workflow in cloud computing based on hybrid particle swarm algorithm. *Indones. J. Electr. Eng. Comput. Sci.* **10**(7), 1560–1566 (2012)
78. 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)
79. Beegom, A.A., Rajasree, M.S.: A particle swarm optimization based pareto optimal task scheduling in cloud computing. In: Tan, Y. (ed.) *In Advances in Swarm Intelligence: 5th International Conference, ICSI 2014, Hefei, China, October 17-20, 2014, Proceedings, Part II*, pp. 79–86. Springer International Publishing, Cham (2014)
80. 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)
81. Pooranian, Z., Shojafar, M., Abawajy, J.H., Abraham, A.: An efficient meta-heuristic algorithm for grid computing. *J. Comb. Optim.* **30**, 413–434 (2015)
82. KRISHNASAMY, K., 2013. Task scheduling algorithm based on Hybrid Particle Swarm Optimization in cloud computing environment. *J. Theoret. Appl. Inform. Technol.* **55**(1).
83. 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
84. Al-maamari, A., Omara, F.A.: Task scheduling using hybrid algorithm in cloud computing environments. *J Comput Eng (IOSR-JCE)* **17**(3), 96–106 (2015)
85. 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)
86. 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)
87. Aron, R., Chana, I., Abraham, A.: A hyper-heuristic approach for resource provisioning-based scheduling in grid environment. *J. Supercomput.* **71**, 1427–1450 (2015)
88. Sidhu, M.S., Thulasiraman, P. and Thulasiraman, 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.
89. 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)
90. Milani, F.S., Navin, A.H.: Multi-objective task scheduling in the cloud computing based on the patrice swarm optimization. *Int. J. Inform. Technol. Comput. Sci.* **7**(5), 61–66 (2015)
91. Wang, Z., Shuang, K., Yang, L., Yang, F.: Energy-aware and revenue-enhancing combinatorial scheduling in virtualized of Cloud Datacenter. *J. Conver. Inf. Technol.* **7**(1), 62–70 (2012)
92. Dorigo, M., Blum, C.: Ant colony optimization theory: a survey. *Theoret. Comput. Sci.* **344**(2–3), 243–278 (2005)
93. Chiang, C.W., Lee, Y.C., Lee, C.N., Chou, T.Y.: Ant colony optimisation for task matching and scheduling. *IEE Proceed-Comput. Dig. Tech.* **153**(6), 373–380 (2006)
94. 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.
95. 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.
96. 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)
97. 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).
98. Chimakurthi, L., 2011. Power efficient resource allocation for clouds using ant colony framework. *arXiv preprint arXiv:1102.2608*.
99. Mathiyalagan, P., Suriya, S., Sivanandam, S.N.: Modified ant colony algorithm for grid scheduling. *Int. J. Comput. Sci. Eng.* **2**(02), 132–139 (2010)
100. 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.

101. 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.
102. Chen, W.N., Zhang, J.: An ant colony optimization approach to a grid workflow scheduling problem with various QoS requirements. *IEEE Trans. Syst. Man Cybern. C* **39**(1), 29–43 (2008)
103. 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.
104. 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)
105. 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)
106. Hasançebi, O.Ğ.U.Z.H.A.N., Teke, T.Ü.R.K.E.R., Pekcan, O.N.U.R.: A bat-inspired algorithm for structural optimization. *Comput. Struct.* **128**, 77–90 (2013)
107. Jacob, L.: Bat algorithm for resource scheduling in cloud computing. *Population* **5**(18), 23 (2014)
108. Kumar, V.S., Aramudhan, M.: Trust based resource selection in cloud computing using hybrid algorithm. *Int. J. Intell. Syst. Appl.* **7**(8), 59 (2015)
109. Kumar, V.S.: Hybrid optimized list scheduling and trust based resource selection in cloud computing. *J. Theor. Appl. Inform. Technol.* **69**(3), 434 (2014)
110. 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.
111. George, S.: Hybrid PSO-MOBA for profit maximization in cloud computing. *Int. J. Adv. Comput. Sci. Appl.* **6**(2), 159–163 (2015)
112. Karaboga, D., 2005. An idea based on honey bee swarm for numerical optimization . Technical report-tr06, Erciyes university, engineering faculty, computer engineering department. 200: 1–10
113. 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)
114. 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)
115. 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)
116. 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. IEEE. pp. 50–53
117. 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)
118. Mousavinasab, Z., Entezari-Maleki, R., Movaghar, A.: A bee colony task scheduling algorithm in computational grids. In: Snasel, V. (ed.) *Digital information processing and communications international conference*, pp. 200–210. Springer, Berlin Heidelberg, Berlin (2011)
119. de Mello, R.F., Senger, L.J. and Yang, L.T., 2006. A routing load balancing policy for grid computing environments. In 20th International Conference on Advanced Information Networking and Applications 1-6
120. 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)
121. 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)
122. 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)
123. 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
124. 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).
125. 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)
126. 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.
127. 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.
128. Kansal, N.J., Chana, I.: Artificial bee colony based energy-aware resource utilization technique for cloud computing. *Concurr. Comput. Pract. Exp.* **27**(5), 1207–1225 (2015)
129. 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. Evol. Biol.* **290**(1), 1–31 (2007)
130. 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. Inform. Sci.* **35**(2), 791–809 (2023)
131. 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)
132. Sreenu, K., Sreelatha, M.: W-Scheduler: whale optimization for task scheduling in cloud computing. *Clust. Comput.* **22**, 1087–1098 (2019)
133. 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)
134. 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)
135. 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)
136. 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
137. Chu, S.C., Tsai, P.W.: Computational intelligence based on the behavior of cats. *Int. J. Innov. Comput. Inform. Control* **3**(1), 163–173 (2007)
138. Tsai, P.W., Pan, J.S., Chen, S.M., Liao, B.Y., Hao, S.P.: Parallel cat swarm optimization. *Int. Conf. Mach. Learn. Cybern. IEEE* **6**, 3328–3333 (2008)

139. Pradhan, P.M., Panda, G.: Solving multiobjective problems using cat swarm optimization. *Expert Syst. Appl.* **39**(3), 2956–2964 (2012)
140. 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
141. Bilgaiyan, S., Sagnika, S., 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
142. 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).
143. Arora, S. and Singh, S., 2013. The firefly optimization algorithm: convergence analysis and parameter selection. *Int J Comput Appl*, 69(3).
144. 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)
145. 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
146. 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)
147. Rajagopalan, A., Modale, D.R, Senthilkumar, R., 2020 Optimal scheduling of tasks in cloud computing using hybrid firefly-genetic algorithm In: Suresh C Satapathy (eds) *Advances in Decision Sciences, Image Processing, Security and Computer Vision on Emerging Trends in Eng.* Springer International Publishing, Cham, pp. 678–687
148. Kashikolaie, 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)
149. 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>
150. 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. Produc. Eng. Manage.* **14**(3), 333 (2019)
151. 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)
152. 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)
153. 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)
154. Bacanin, N., Bezdán, 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.
155. 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)
156. Natesan, G., Chokkalingam, A.: Task scheduling in heterogeneous cloud environment using mean grey wolf optimization algorithm. *ICT Express* **5**(2), 110–114 (2019)
157. 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.
158. 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)
159. 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)
160. Zolghadr-Asli, B., Bozorg-Haddad, O., Chu, X.: Crow search algorithm (CSA) Advanced optimization by nature-inspired algorithms. In: Bozorg-Haddad, O. (ed.) *Crow search algorithm (CSA)*, pp. 143–149. Springer Singapore, Singapore (2018)
161. Prasanna Kumar, K.R., Kousalya, K.: Amelioration of task scheduling in cloud computing using crow search algorithm. *Neural Comput. Appl.* **32**, 5901–5907 (2020)
162. Kumar, K.P., Kousalya, K., Vishnupriya, S., Ponni, S., Logeswaran, K.: Enhanced crow search algorithm for task scheduling in cloud computing. *IOP Conf. Ser. Mater. Sci. Eng.* **1055**(1), 012102 (2021)
163. Singh, H., Tyagi, S., Kumar, P.: Crow–penguin optimizer for multi-objective task scheduling strategy in cloud computing. *Int. J. Commun. Syst.* **33**(14), e4467 (2020)
164. 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)
165. Singh, H., Tyagi, S., Kumar, P.: Cloud resource mapping through crow search inspired metaheuristic load balancing technique. *Comput. Electr. Eng.* **93**, 107221 (2021)
166. 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. *Multimed. Res.* **4**(3), 17 (2021)
167. Kak, S.M., Agarwal, P., Alam, M.A., Siddiqui, F.: A hybridized approach for minimizing energy in cloud computing. *Clust. Comput.* (2022). <https://doi.org/10.1007/s10586-022-03807-9>
168. Mangalampalli, S., Mangalampalli, V.K., Swain, S.K.: A task scheduling approach in cloud computing to minimize the power cost in datacenters using crow search. *Clust. Comput.* (2022). <https://doi.org/10.1007/s10586-022-03786-x>
169. Joshi, A.S., Kulkarni, O., Kakandikar, G.M., Nandedkar, V.M.: Cuckoo search optimization-a review. *Mater. Today: Proceed.* **4**(8), 7262–7269 (2017)
170. 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)
171. 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)
172. 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)
173. 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)
174. Agarwal, M., Srivastava, G.M.S.: A cuckoo search algorithm-based task scheduling in cloud computing. In: Bhatia, S.K. (ed.)

- Advances in Computer and Computational Sciences, pp. 293–299. Springer Singapore, Singapore (2018)
175. Nazir, S., Shafiq, S., Iqbal, Z., Zeeshan, M., Tariq, S., Javaid, N.: Cuckoo optimization algorithm based job scheduling using cloud and fog computing in smart grid. In: Xhafa, F. (ed.) *Advances in Intelligent Networking and Collaborative Systems: The 10th International Conference on Intelligent Networking and Collaborative*, pp. 34–46. Springer International Publishing, Cham (2019)
  176. 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.* (2017). <https://doi.org/10.12720/jait.8.4.210-218>
  177. 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)
  178. Krishnadoss, P., Jacob, P.: OCSA: task scheduling algorithm in cloud computing environment. *Int. J. Intell. Eng. Syst.* **11**(3), 271–279 (2018)
  179. 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)
  180. 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)
  181. 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)
  182. Durgadevi, P., Srinivasan, S.: Resource allocation in cloud computing using SFLA and cuckoo search hybridization. *Int. J. Parallel Prog.* **48**, 549–565 (2020)
  183. 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)
  184. Zhou, J., Dong, S.: Hybrid glowworm swarm optimization for task scheduling in the cloud environment. *Eng. Optim.* **50**(6), 949–964 (2018)
  185. Alboaneen, D.A., Tianfield, H. and Zhang, Y., 2017. 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).
  186. Zheng, R., Hussien, A.G., Jia, H.M., Abualigah, L., Wang, S., Wu, D.: An improved wild horse optimizer for solving optimization problems. *Mathematics* **10**(8), 1311 (2022)
  187. Sad, 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)
  188. 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)
  189. Abdullahi, M., Ngadi, M.A., Dishing, S.I. and Abdulhamid, S.I.M., 2022. An adaptive symbiotic organisms search for constrained task scheduling in cloud computing. *Journal of Ambient Intelligence and Humanized Computing*, pp.1–12.
  190. Sharma, M. and Verma, A., 2017. 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.
  191. 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)
  192. Yingqiu, L., Shuhua, L., Shoubo, G., 2016. Cloud task scheduling based on chaotic particle swarm optimization algorithm. In *2016 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)* (pp. 493–496). IEEE.
  193. Alshattnawi, S., Al-Marie, M.: Spider monkey optimization algorithm for load balancing in cloud computing environments. *Int. Arab J. Inf. Technol.* **18**(5), 730–738 (2021)
  194. Masadeh, R., Alsharman, N., Sharieh, A., Mahafzah, B.A., Abdulrahman, A.: Task scheduling on cloud computing based on sea lion optimization algorithm. *Int. J. Web Inform. Syst.* **17**(2), 99–116 (2021)
  195. Bey, K.B., Bouznad, S., Benhammedi, F., Nacer, H., 2019. Improved Virus Optimization algorithm for two-objective tasks Scheduling in Cloud Environment. In *FedCSIS (Communication Papers)* (pp. 109–117).
  196. Mandal, T. and Acharyya, S., 2015. 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
  197. Ramezani, F., Lu, J., Hussain, F.: Task scheduling optimization in cloud computing applying multi-objective particle swarm optimization. In: Maglio, P.P. (ed.) *Service-Oriented Computing*, pp. 237–251. Springer, Berlin Heidelberg, Cham (2013)
  198. 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)
  199. Khalili, A. and Babamir, S.M., 2015. Makespan improvement of PSO-based dynamic scheduling in cloud environment. In *2015 23rd Iranian Conference on Electrical Engineering* (pp. 613–618). IEEE.
  200. Gabi, D., Ismail, A.S. and Dankolo, N.M., 2019. 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).
  201. 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)
  202. 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)
  203. 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)
  204. Kumar, N. and Patel, P., 2016. 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).

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.