Journal of Advanced Research in Applied Sciences and Engineering Technology 33, Issue 3 (2024) 194-218

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

A Survey on the Applications of Cuckoo Search Algorithm in Resource Management and Scheduling in Cloud Computing

Aishatu Yahaya Umar[1,*], Abdulsalam Ya'u Gital[1], Haruna Chiroma[2], Kabiru Ibrahim Musa[3], Ali Muhammad Usman[4], Shafi'i M. Abdulhamid[5], Abubakar Usman Othman[6]

[1] Department of Mathematical Sciences, Abubakar Tafawa Balewa University, Bauchi, Nigeria
[2] College of Computer Science and Engineering, University of Hafr Al Batin, Saudi Arabia
[3] Department of Environmental Management Technology, Abubakar Tafawa Balewa University, Bauchi, Nigeria
[4] Computer Science Department, Federal College of Education (Technical) Gombe, Nigeria
[5] Information Technology Department, Community College of Qatar, Doha, Qatar
[6] National Agency for Science and Engineering Infrastructure, Idu-Abuja, Nigeria

ARTICLE INFO

Article history:
Received 8 July 2023
Received in revised form 20 October 2023
Accepted 2 November 2023
Available online 17 November 2023

Keywords:
Resource Management; task scheduling; resource allocation; Cuckoo Search Algorithm; cloud computing

ABSTRACT

Users can quickly and affordably access computing resources and services from any location. However, due to the growth in magnitude, complexity, and power intake about provider requirements, the resources are distributed on a payment basis. Resource allocation in data centres is a difficult task since consumer requirements for execution time, dependability, availability, bandwidth, and cost frequently conflict with the goals of cloud providers; this is a serious problem with the cloud that needs to be solved. A meta-heuristic approach called Cuckoo Search Algorithm (CSA) is employed to optimize a variety of cloud computing issues. A number of research studies have been carried out in a cloud computing environment employing CSA. This paper examines the use of CSA for resource management and task scheduling and identifies the application domain. Finally, it identifies challenges and the key areas that require further research.

1. Introduction

Cloud computing is a hire-based service model that offers on-request access to resources (hardware, software, and platforms) over the internet so that infrastructure and maintenance costs can be reduced significantly [1-3]. It is the latest solution to the challenge of the availability of resources for individuals, companies, and organizations. Clients use the resources as needed as conveniently as possible, and they pay using a pay- per-usage mechanism [4,5]. The deployment model explains how to access cloud resources, whereas the service model outlines the services the cloud rendered to the cloud users. The services rendered in cloud include Software as a Service (SaaS), Infrastructure as a Service (IaaS), and Platform as a Service [1,6,7]. The deployment models for delivering cloud services to consumers are classified into four; private, public hybrid, and

* *Corresponding author.*
E-mail address: aishatuyu@gmail.com

https://doi.org/10.37934/araset.33.3.194218

community cloud [8]. Cloud computing uses two types of scheduling: Task scheduling and resource scheduling, where tasks or jobs are distributed and resources are assigned for computation [9]. Cloud scheduling issues are grouped into application scheduling, virtualization, and infrastructure layer [10].

Effective resource management and scheduling are always needed for the optimum utilization of resources for any successful cloud computing system [11]. Resources, money, and time are very scarce in most applications. The less time a resource is kept inactive between tasks, the more utilization. Thus, task-to-resource assignment is critical. A resource is essential to carry out a process in the cloud computing system and needs to be managed [12]. Resource management is the method by which the resources, including servers, storage, database, and software) are assigned to the right tasks for effective utilization [13].

Several algorithms have been applied for resource management and scheduling, including deterministic, heuristics, and meta-heuristics algorithms. Meta-heuristic algorithms have achieved close optimal solutions quickly compared to conventional algorithms. Cuckoo Search Algorithm (CSA), Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) are examples of common meta-heuristic algorithms used in resource management in cloud environments. Recent findings show the usefulness of CSA over other meta-heuristic algorithms [14]. CSA has recorded tremendous success as it strikes an equilibrium between local and global searching and has fewer parameters, improving the convergence speed [1].

The main of contributions of this research are summarized as follows:

i. A comprehensive review of the use of CSA in cloud environments is presented between 2011 and 2022.
ii. The different CSA variants used to address various issues in the cloud environment are identified. The taxonomy includes basic cuckoo, modified cuckoo, binary cuckoo, and hybrid cuckoo.
iii. A taxonomy on implementing CSA in cloud computing from different domains is created, including resource scheduling, task scheduling, load balancing, and virtual machine placement.
iv. The challenges ahead are explored, and opportunities for future research are outlined.

The paper is organized as follows: in section 2, a synopsis of CSA and a brief introduction to resource management and scheduling were provided. Section 3 introduces research that has applied CSA for resource scheduling and management in the cloud. Section 4 talks about the challenges that require further research, and finally, section 5 presents future trends and a conclusion.

1.1 Background
1.1.1 Resource scheduling and management in cloud computing environment

Currently, providers can receive numerous jobs when many consumers are registered or many requests on demand. These tasks need to be organized and executed on time and all available resources are used efficiently. Thus, task-to-resource mapping is essential. Scheduling deals with the problem of tasks to resource allocation. It refers to organizing and controlling workloads in a manner that enables their execution [15]. Resource scheduling allocates the specific and proper cloudlet or task to the cloud users in cloud computing [12]. It comprises two roles: resource mapping and resource allocation. Resource allocation is to apportion proper resources to clients according to their demands on time so that resources can be utilized effectively [16,17]. Depending on the Service Level Agreement (SLA) between the provider and the client, the resource is allotted with restrictions. The constraints on the side of cloud users include time, cost, deadline, and budget. In comparison, the cloud provider's side includes maximizing the cost, profit, and resource utilization [12,13,15]. Cloud performance is directly affected by how resources are fully allocated and utilized [18]. Resource mapping is the method of assigning jobs to the appropriate resources following the Quality of Service (QoS) requirements that the user specifies concerning the performance indicators, which include cost, execution time, and profit [19]. when resources are limited and demands need to be met, virtualization technology comes up as it allows concurrent users and supports on-request delivery [20]. Virtualization is a technique by which physical infrastructure is abstracted to provide virtualized computer resources called a virtual machine (VM). In addition, virtualization supports the live migration of virtual machines.

Issues in scheduling aim to obtain a viable schedule in a limited or unlimited set that optimizes the proposed objective function [21]. When making scheduling decisions, the optimization criterion is utilized [12]. Optimization is a procedure for examining the ideal solution to a specific problem of interest. We all have finite resources and time and want to make the most of them. Optimization algorithms are advanced techniques for providing solutions to problems, although such optimality is not always achievable [22]. Any technique or plan for allocating resources that works best must produce results that adhere to the SLA's QoS requirements [23]. Resource management and scheduling are serious problems in cloud computing that have been studied extensively. Efficient resource management and scheduling would help reduce costs, decrease virtual machines, and increase the fitness function [24].

1.1.2 Basic concept of the cuckoo search algorithm

Yang and Deb [25] created the CSA algorithm in 2009. It was inspired by the parasitic behavior of a cuckoo bird by laying its eggs in the most suitable nest of another bird called a host bird [14,26]. The key rules governing the algorithm are:

i. A cuckoo bird lay an egg and places it into a nest of its choosing at random.
ii. Nests that have good-quality eggs move to a subsequent iteration.
iii. With a fixed host nest, the host bird can reveal an unfamiliar egg placed by a cuckoo with a probability Pa$\in$ (0, 1). In this situation, the host bird has two options: discard the cuckoo egg or abandon the nest and create a new one to replace it [27-30].

A nest of eggs indicates solutions that have been set; a new solution illustrates cuckoo egg, and a nest that contains numerous eggs suggests solutions that have been set in complicated cases. The worst solutions in the nest can be substituted with a better solution [25]. The value of solutions or wellness is assessed with the objective function of solving the issue. In maximization issues, it is relative to the esteem of the objective function [22]. Yang and Deb [25] further revealed that levy flights are better in the random walk style search than the simple random walk.

The algorithm uses a well-adjusted combination of an arbitrary local walk and the global exploratory random walk, under the control of an exchange parameter pa. Eq. (1) represents the random local walk

$$x_i^{t+1} = x_i^t + \propto s \otimes H\,(pa - \epsilon) \otimes (x_j^t - x_k^t) \quad (1)$$

where x_j^t and x_k^t are solutions chosen at random by random permutation, H(u) is a Heaviside function, $\in$ is a random number selected at random from a uniform distribution, and *s* is the step size. The product $\otimes$ refers to the exclusive OR operation or entry-wise multiplication. Moreover, the global random walk, a new solution $x_i^{(t+1)}$ for cuckoo *i*, is carried out using Levy flights that play critical roles in controlling the adjustment between intensification and diversification [31].

$$x_i^{t+1} = x_i^t + \alpha L(s, \lambda) \tag{2}$$

where the step size scaling factor associated with the scaling of the relevant problem is α > 0. The stochastic equation for a random walk is given above. A random walk is a Markov chain in which the only factor determining the next position is the current position. To ensure that the system is not pinned in a local optimum, however, a significant number of fresh solutions should be produced through far-field randomization and their positions should be sufficiently removed from the previous best solution [25,30].

Based on the three rules above, CSA involves 4 steps, starting, searching, selection, and assessment. The steps are as follows:

i. Start: Arbitrarily initialize the N nestsx_0= ($x_{1,}^0 x_2^0, \ldots x_N^0$) in their places, calculate their objective values F_o, and choose the best position.
ii. Searching: create new positions in the Lévy flight method x_t = ($x_{1,}^t x_2^t \ldots x_N^t$), calculate their objective value F_t, and selecting the best position once more. Lévy flight employs a random walk approach as in Eq. (3)

$$x_{t+1,i} = x_{t,i} + \alpha_0 \frac{\emptyset \times u}{|v|^{\frac{1}{\beta}}} (x_{t,i} - x_{t,best}) \tag{3}$$

where, $x_{t,i}$ is the ith solution from generation t; x_{best} is the current best option; α_0 is the step constant, typically equal to $\alpha_0 = 0.01$; the normal standard random variables *u* and *v*; the Lévy flight control factor is β, generally β=1.5; ϕ uses to express as shown in Eq. (4).

$$\emptyset = \left[\frac{\Gamma(1+\beta) \times sinsin\,(\pi \times \beta/2)}{\Gamma\left[\left[\frac{1+\beta}{2}\right] \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}\right]} \right]^{\frac{1}{\beta}} \tag{4}$$

iii. Selection: Eliminate the worst positions in detection probability p_a, and apply (3) to create an equal number of new spots. Determine their objective values F_t and again select the best position.

$$x_{t+1,i} = x_{t,i} + r(x_{t,j} - x_{t,k}) \tag{5}$$

where, *r* is the scaling factor, which is a standard random variable; $x_{t,j}$, and $x_{t,k}$ is any two solutions in generation *t*.

iv. Completion assessment: If the iterative/maximum generation is reached, then stop or return to (iii) [25]:

Algorithm 1: Pseudocode of the Cuckoo Search Algorithm
Start
1. Set the objective function $f x$,x = (x_1, ..., x_d)^T
2. Generate the initial population of n host nests x_i (i = 1, 2, ..., n)
3. Set t = 0
4. while (t <MaxGeneration) or (stop criterion)
5. Get a cuckoo randomly by Levy flights
6. Evaluate its quality/fitness fi
7. Choose a nest among n (say, j) randomly
8. if (fi >fj).
9. replace j with the new solution.
10. Stop
11. A fraction (pa) of the worse nests is abandoned and new ones are built
12. Keep the best solutions/ nests with quality solutions for the next generation
13. Rank the solutions and find the current best
14. end while
15. Postprocess results and visualization

Stop

1.1.3 Variants of cuckoo search

Since its introduction in 2009, the Cuckoo Search Algorithm (CSA) has undergone numerous modifications to increase its effectiveness and efficiency in solving optimization problems. While CSA can efficiently find appropriate solutions, issues may arise where the solutions cannot be found, or attempts to improve efficiency to obtain better solutions. Fister Jr *et al.,* [27] state that there are several CSA variations, including Modified CSA, Multi-objective CSA (which combines multiple objectives), and a method that has been combined with machine learning and other optimization algorithms. Walton *et al.,* [32] discussed the cuckoo variations based on changes made to the fundamental algorithms and cuckoo hybridized with other search optimization techniques. Modifications are made to parameter control and representation of solutions (eggs). Kamat and Karegowda [28] described three variations of CSA: Binary CS, Modified CS, and Improved CS. Chiroma *et al.,* [1] reviewed the latest improvements of the adjustments made to the basic cuckoo search and state that the adjustment is based on solution representation, how new solutions are created, the evaluation function employed, and the replacement operators. Representations of solutions are categorized into Continuous Cuckoo Search (CCA) and binary Cuckoo Search (BCS). The division can also be based on parameter control, hybridization, and multi-objective function. For example, Shehab *et al.,* [29] categorized CSA into binary CS, Discrete Cuckoo Search (DCS), Modified Cuckoo Search (MCS), other improved CS, and hybrid, population-based algorithms, and other heuristic algorithms. The study revealed that the CSA variant applied in cloud computing is categorized into Basic Cuckoo Search (BCS), MCS, Binary Cuckoo Search, and hybridized CSA.

1.1.4 Applications of CSA

CSA has been applied in various domains, including the health sector, industry, image processing, job scheduling, pattern recognition, wireless sensor network, and optimizing parameters of numerous classifiers, which include Neural networks, RBF, SVM parameters, etc [33,34]. Moreover, it has shown promising superiority and efficiency over other algorithms [1,28]. Ouaarab *et al.,* [21] applied a DCS to reduce schedule completion time. As a result, DCS shows lower completion time and balances the search effectively between intensification and diversification. Piechocki *et al.,* [35] applied the MCS to identify optimal solutions for integrated energy systems and self-powered farms. The algorithm obtains optimal or near-optimal value with negligible costs, pollutant emission, and energy demand. Boushaki *et al.,* [36] applied an improved cuckoo search for document clustering (ICSCA) using four text document datasets and Iris standard dataset. BCS was also applied for feature selection using two data sets to detect thefts in power distribution systems using Optimum-Path Forest (OPF) classifier. BCS is the fastest approach for feature selection suitable for industrial data sets [37]. CSA is also used in structural design optimization of automotive parts to support innovative design and reduce development costs and time [38].

Maurya *et al*., [39] applied modified cuckoo search to images to improve contrast and brightness. Sanajaoba and Fernandez [40] employed a cuckoo search to find the size best for hybrid energy systems photovoltaic, wind, and photovoltaic-wind battery systems. Because of this, the convergence of CS is quicker, resulting in less resource consumption and higher-quality solutions. CSA was introduced by Yildiz [41] to optimize machining cutting parameters to maximize profit in milling operations. To increase the whole system's reliability and save costs, CSA was applied to resolve a reliability optimization issue [42]. To choose hardware for each subsystem from a variety of options, it is formulated as binary integer nonlinear programming for a series system.

2. Related Work

The service provider deals with many cloud users in the cloud environment simultaneously. Therefore, scheduling the appropriate workloads and assigning the available resources to the cloudlet must be done on time. Scheduling algorithms are used to solve these kinds of problems. Numerous researchers have proposed different meta-heuristics algorithms to manage resource in cloud computing. This paper focuses on research that has used CSA in cloud environment. CSA algorithm applied in cloud is categorized into the BCSA, MCSA, BCSA, and hybridized cuckoo with other search optimization algorithms. Figure 1 shows the different variants of CSA applied in resource management/task scheduling.

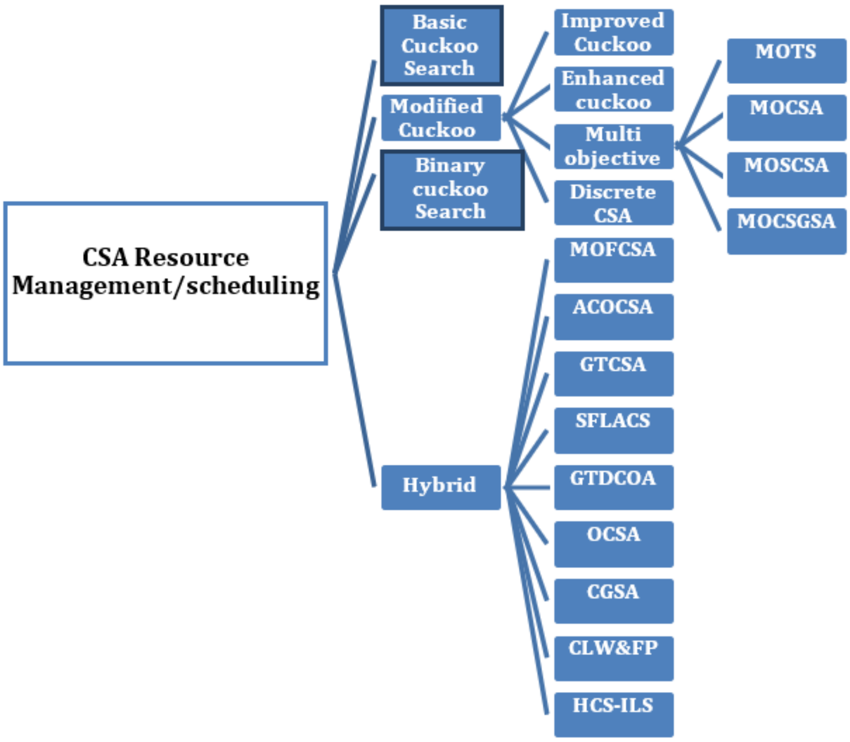

Fig. 1. Categories of CS Resource Allocation/Task scheduling Algorithm

2.1 Basic Cuckoo Search Algorithm

Salami *et al.,* [43] applied CSA in providing a solution to the Virtual Machine Placement Problem by developing new cost and perturbation functions to increase algorithm performance. The emergence of virtualization technology has resulted in an extensive application among users and organizations in cloud environment. In most cases, many physical assets are split up into smaller virtual entities. Numerous clients will have access to these little units. Two benchmark datasets are currently available for testing the suggested technique. This outperformed the best-fit decreasing, first-fit decreasing, multi-CSA, the reordered grouping genetic algorithm, and other CS techniques.

Kurokawa and Hayashibara [44] proposed a bio-inspired metaheuristic for Cloud-based Content delivery networks (CCDNs) that support multimedia services and improve availability, access efficiency, and load balancing with the fulfilment of Quality-of-Service requests. By serving or caching known content on servers close to users, CCDNs can provide much faster access to content. The method was also implemented and evaluated with the content hit rate, storage, and communication cost in the presence of errors. Results showed an advantageous improvement in the content hit rate without significantly altering storage and communication cost.

Aloboud and Kurdi [45] proposed a CSA to decrease the turnaround time and increase resource usage. Job priority is considered where the higher priority jobs (host bird) are given the required resources. Lower priority jobs (cuckoo jobs) must wait if pre-empted as they have no deadline. The proposed model was implemented using python and tested on the Haizea scheduler. The experimental result shows that the low-precedence cuckoo jobs performed better than non-fragment low-priority jobs with minimal waiting time and CPU utilization. Cuckoo search (CS)-based job scheduling was used by Agarwa and Srivastava [11] to optimally distribute workloads among the available virtual machines (VMs) while minimizing response times. Based on their size and processing capability. The algorithm allocates the jobs to the VM based on the processing power and length. In comparison with FIFO and greedy-based scheduling using the CloudSim, result shows that the CSA outperformed the greedy-based scheduling algorithm.

Silambarasan and Kumar [46] applied CSA to reduce execution time and optimized virtual machines to improve system efficiency. Mathlab was used for the implementation. The result shows that CSA produces better results than existing methods. Singh and Randhawa [47] proposed CSA to reduce the cost of execution and workflow execution time. A Directed Acyclic Graph (DAG) model of the workflows is used, with a single start node and a single end node. Resources are allocated based on the priority assigned to tasks, and deadlines as a constraint. An evaluation was conducted, and the results show cuckoo scheduler outperforms PSO and HEFT and reduces overall workflow completion time, thereby minimizing execution costs. Sarddar *et al.,* [48] suggested parallelism in CPU virtualization and scheduling using CSA to maximize throughput. CSA performance is evaluated and compared with traditional FIFO, SJF, and Round Robin algorithms. Experimental results show that CSA has less average waiting time than other algorithms.

Madni *et al.,* [12] proposed CSA to minimize response time, throughput, and Makespan for infrastructure as a service for resource planning. Virtual machines (VMs) are deemed in the research as resources. The result shows that CSA outperforms Ant Colony Optimisation (ACO) algorithm with higher proficiency, shorter response time, and better performance. However, the proposed system is limited to a few VM and users. Dong-han and Hai-tao [49] presented a cloud computing environment resource allocation ICS method with energy conservation awareness. CSA is applied to resource scheduling to enhance the usage rate and power efficiency of the cloud infrastructure. The method assumed the task as an egg and the host nest as the resource. Furthermore, a double threshold was introduced to fix the higher and minimum utilization thresholds. As a result, ICS algorithms show higher CPU utilization is energy efficient and environmentally friendly than GGA. However, the dependency between tasks is not considered. George *et al.,* [15] applied CSA to reduce the cost involved in task execution. The model assumes that the tasks are independent, and each task is assigned a unique resource to be executed; no pre-emption is done. The method shows superior performance with minimal execution time, though the proposed work only considers the cost of execution of the client job. Alexander and Joseph [50] proposed a load-aware allocation scheme using CSA to reduce the completion time and the cost of computation. Results indicated a drop in the time of execution and computational cost with a higher percentage of deadlines met than PSO. The proposed method was model to work with a single data center, and other fluctuations in bandwidth are not considered.

Sait *et al.,* [51] applied the CSA algorithm in solving the VM placement issues of data centers and used a MOCSO algorithm to concurrently optimize the data center's energy usage and resource wastage. CSO uses a VM fitness measure rather than group-based to optimize, to reduce the physical machines utilized in placement (server). Contrarily, in Multi-objective CSO, fuzzy logic is obtained by evaluating the fitness of a specific nest. The Perturb_ function, which takes a nest as input and produces x from the levy distribution, was employed by both methods. According to the simulation results, CSO and multi-objective CSO algorithms beat the enhanced FFD (IFFD) and least-loaded (ILL) VM placement strategies because they display a better balance between resource wastage and power usage. Yakhchi *et al.,* [52] used a CSA to reduce energy consumption and detect over-utilized hosts. CSA was used to detect over-loaded hosts, and the MMT policy was used to migrate the VM's to other hosts from the over-loaded or under-loaded hosts. The method is compared with MAD-MMT (Median Absolute Deviation- Minimum Migration Time), IQR-MMT (Interquartile Range- Minimum Migration Time), Bee-MMT (Bee colony algorithm- Minimum Migration Time), LR-MMT (local Regression-Minimum Migration Time). Violation in SLA is compared to the non-Power aware policy which it shows the lowest SLA Violation (in the PDM and SLAV metrics). Table 1 summarizes the BCSA observed in our study.

Table 1
Summary of Basic CSA methods

Reference	Cuckoo Search Variants	Comparison Algorithm	Findings	Associated problem
Vishal *et al.,* [53]	Cuckoo search with artificial neural network	Modified Best Fit Decreasing (MBFD)	Service level agreement violation is high	The level of violation of SLA is high
Sait *et al.,* [51]	Cuckoo search Multi-objective optimization	RGGA, GGA ILLand improved FFD (IFFD)	Less physical machines, power consumption, and resource wastage. efficient for a higher number of VM	Applied to only homogeneous data center, very expensive VM migration
Yakhchi *et al.,* [52]	COA-MMT	MAD-MMT, IQR-MMT	Reduced power consumption and fewer SLA violations	Response time is not considered which is a required factor to guarantee high quality of service
Navimipour and Milani [54]	CSA	None	Minimized execution time of tasks	Not implemented. Do not compare with any model/algorithm
George *et al.,* [15]	CSA	Default Scheduling	Superior performance with minimum makespan	The approach Modeled the execution cost of the client task only
Salami *et al.,* [43]	CSA	PSO	An improved response time	Poor selection of destination host for migration
Alexander and Joseph [50]	CSA	Particle Swarm Optimisation (PSO)	The proposed strategy shows reduced total execution time as well as computational time, the deadline met compared to PSO algorithm is also high	The method is designed for a single data center and fluctuations in network bandwidth are not considered
Sarddar *et al.,* [48]	CSA	FIFO, SJF, RR	Reduced starvation for processes, maximized resource utilization	Not implemented and simulated. Compared with only traditional algorithms
Madni *et al.,* [12]	CSA	Ant colony optimization (ACO)	higher proficiency and better performance with a shorter response time	The proposed scheme is limited to a small number of users, VM.
Singh and Randhawa [47]	CSA	Heterogeneous Earliest Finish Time (HEFT) & PSO	Reduced overall workflow time and minimal execution cost	Do not consider Energy consumption

Agarwal and Srivastava [11]	CSA	FIFO & Greedy algorithm	The cuckoo search proves to be more efficient with minimum execution time	The proposed approach focuses on execution time only, other QoS parameters that improve performance significantly are not considered
Silambarasan and Kumar [46]	CSA	Not specified	CSA provides a better result	Consider execution time only
Aloboud and Kurdi [45]	CSA	Non-fragmented low-priority jobs	Minimal time of execution and increased CPU usage	Consider execution time and CPU usage only
Kurokawa and Hayashibara [44]	Cloud-based Content delivery networks (CCDNs)	PSO	The improvement in the content of his ratio is minimal	The deployment or caching of famous content on servers degrades performance

2.2 Modified Cuckoo Search Algorithm

Barlaskar *et al.,* [9] proposed an enhanced cuckoo search (ECSA) to minimize power consumption and VM migration costs and decrease SLA breaches. The proposed method considers each cuckoo as a vector of natural numbers, the elements act as the physical machines to which the VMs are mapped. MMT & MU selection policies are used, ECSA showed less energy consumption and stable performance for SLA and VM migration compared to OFS, GA, and AC. In addition, evaluation results show that ECS performs well with less workload. However, with the ability to utilize levy flights, the algorithm can run just as efficiently in a large-scale, dynamic cloud environment. An improved cuckoo search was proposed by Silambarasan and Kumar [46] to optimize VMs and reduce task execution time. The proposed method was implemented using Matlab; results show an improvement in system performance. Table 2 presents the summary of related work on MCSA.

Table 2
Summary of related literature on modified cuckoo search algorithm

Reference	Cuckoo Search Variants	Compared Algorithm	Findings	Associated problem
Dong-han and Hai-tao [49]	ICS	GGA	Higher CPU utilization and energy efficiency	Dependencies between tasks are not considered
Barlaskar *et al.,* [9]	Enhanced Cuckoo Search (ECS)	Genetic Algorithm (GA), Optimized Firefly Search (OFS) algorithm, and Ant Colony (AC) algorithm	Consumes less energy and shows a decrease in SLA violation and migration of VM	The approach considered only two eggs in a single nest and was not implemented in a real heterogeneous environment
Silambarasan and Kumar [46]	Improved cuckoo search (ICS)	Not specified	Reduced execution time and better performance	Not compared with any other method

2.3 Multi-Objective Cuckoo Search

In their research, Madni *et al.,* [55] presents MOCSO model to deal with the resource allocation issues in the cloud to reduce cost, enhance performance, reduce completion time, and increase resource utilization. Multiple eggs are considered in each nest, where the nests are distributed and concentrated on nests in control with worst solutions and other nests with Pareto optimal solutions (best). MOCSO performance was compared with four scheduling algorithms, MOACO, MOGA, MOMM, and MOPSO. The computational results illustrated that the MOCSO algorithm outperformed the other algorithms. Furthermore, it effectively uses resources through cost minimization and time. Based on the resource allocation matrices for the IaaS cloud environment, it balanced the various objectives based on the projected make span and computational cost-to-completion matrices. A multi-objective-based hybrid CSA and GSA was proposed by Pradeep and Jacob [24] to schedule jobs based on parallel processing to minimize cost, storage, and power consumption. The proposed CGSA showed better results than GSA, CS, PSO, and GA-based scheduling algorithms.

Panda and Jana [56] proposed a multi-objective task scheduling algorithm to reduce makes pan and total costs for a heterogeneous multi-cloud environment. First, the application is in the form of a Directed Acyclic Graph. Min-Min method was used on the matrix after the procedure has scaled the values between 0 and 1 through the normalization procedure. The algorithm is then contrasted with PBTS and ETBTS, two other task-scheduling techniques. The result shows that MOTS outperforms both algorithms, balancing total cost and execution time.

To deal with the allocation of jobs on various systems, MOSCOA was proposed by Akbari and Rashidi [57] to minimize time of execution while parallelization is achieved at highest level. The scheduled program is expressed by DAG, with jobs representing a node, and every edge represents the task execution dependency. The performance of MOSCOA was tested using HEFT-T, HEFT-B, CPOP, and BGA algorithms on many random and real-world application graphs. The result shows the superiority of MOSCOA over the algorithms.

Saif *et al.,* [58] applied MOCSO for efficient resource management by minimizing energy consumption and optimizing CPU and Bandwidth usage considering QoS. The model was implemented on cloud sim and Eclipse IDE. Experimental results showed MOCSO performed well on energy consumption, CPU usage, SLA violation rate, and execution cost as performance metrics. Elnahary *et al.,* [59] applied an efficient cuckoo search (ECS) algorithm to minimize makespan and maximize speed up, efficiency and throughput. Small position Value (SLV), Largest Position Value (LPV), Round Nearest Function, Floor Nearest Function, and Ceil functions are used to convert continuous values to discrete values. Evaluation results on DAG of several cases show ECS outperformed Upward Rank, Downward Rank, Level Rank, BGA, GA_DE_HEFT, CPOP, MHEFT, and Hybrid chemical reaction (HCR) as shown in Table 3.

Table 3
Summary of related literature on Mocsa

Reference	Cuckoo Search Variants	Compared Algorithm	Findings	Limitation
Naik *et al.,* [7]	MOFCSA	MOGA, MOACO	Reduction of power usage and unnecessary VM migration.	SLA violation and communication overhead are not considered
Panda and Jana [56]	MOTS	Execution Time-Based Task Scheduling (ETBTS) and Profit Based Task Scheduling (PBTS)	Reduced Total Cost and makespan	SLA negotiation is not considered
Akbari and Rashidi [57]	Multi-objective scheduling cuckoo optimization algorithm (MOSCOA)	HEFT-T, HEFT-B, CPOP, and BGA	Reduced execution time, and communication cost with maximum parallelization	High repetition rate to produce the desired results
Madni *et al.,* [55]	MOCSA	MOACO, MOGA, MOMM, and MOPSO	Reduced estimated time and expected cost of completion	Service Level Agreement (SLA) Not considered

2.4 Hybrid Cuckoo Search Algorithm

Naik *et al.,* [7] considered a mixed fruit fly cuckoo search algorithm in virtual machine consolidation to minimize the power consumption of the data centre and resource wastage. The fruit fly algorithm is used to find the right position with its desperate vision and smell. It combines CS to update the recent position and solutions to prevent the fruit fly algorithm from falling into the local extremum. A local manager is employed within the current resource usage and availability of the host, detects host status either overload or underload, and a global manager acts as an overseer to all the local costs. The model demonstrates better results than MOGA and MOACO resulting in a significant decrease in VM migration, eliminating the selection of unsteady hosts, and improving both application performance and power consumption efficiency. Raju *et al.* (2012) applied a hybrid of ACO and CSA to improve performance and reduce the completion time of jobs. First, the best path is discovered using ant colony optimization. The local search is then conducted using the cuckoo search to get around ACO's longer search time. The model performs better than ACO, and its span and energy tend to reduce as the number of tasks rises.

Shahdi-Pashaki *et al.,* [5] presents a model based on Group Technology (GT) and CSA to assign VM to workflows to monitor transfer cost, penalty cost, and server cost. Group Technology is used to solve small-size problems; COA is used to handle large-scale problems for which there is no best solution. The proposed approach performs better in small and large problems when compared with the Round Robin Algorithm. The approaches focus only on cost and are compared with only traditional algorithms. Task scheduling hybrid cuckoo genetic algorithm was proposed by Aujla *et al.,* [60] to reduce time of execution and power usage and maximize resource usage. GA is used to schedule the jobs, and CSA uses the output from GA as input. The algorithm shows a better solution when the results are compared with FIFO, GA, and CS.

Nirmala and Veni [61] used the Nelder–Mead method, ACO, and CS with k-means++ scheduling algorithms to improve throughput and reduce execution time of cloud applications in a virtualized environment. ACO is efficient for finding the optimal path through the graph, while CS overcomes the drawback of considerably more time taken in ACO by performing the local search. Various file types, like text documents, PDFs, and images are used to implement the proposed hybrid technique. Results showed better performance compared to the PSO-based Kmeans++ algorithm.

Hybrid oppositional cuckoo search algorithm (OCSA) is presented by Pradeep and Jacob [24] to speed up implementation and lower costs. When compared to PSO, GA, and IDEA the performance of the model improves even when the number of tasks grows, also the cost is much lower compared to PSO and IDEA, but an extreme difference in cost compared to GA, thus proving its cost-efficiency. Its drawback is that it is not suitable for real-time applications. To minimize cost and the time of implementation. The technique is compared against PSO, GA, and IDEA, where the model's performance improves as the number of tasks grows. Compared to PSO and IDEA, the cost is significantly lower, but there is a significant cost difference compared to GA, demonstrating its effectiveness.

Pradeep and Jacob [62] employ a hybrid CS and harmony search (HS) algorithm (CHSA) to reduce costs, energy use, storage utilization, fines, and maximize credit. This technique explores a new exploration space via hybrid HS and uses CS to explore the population. HS optimizes the workload on virtual computers while enhancing the performance of CS algorithms. Three sub-divisions with varying amounts of physical and virtual machines serve as the foundation for performance analysis. It has been found that the CHSA outperforms the current hybridized algorithm (CGSA), individual CS, and HS algorithms.

According to Durgadevi and Srinivasan [63], they used a combined Shuffled Frog Leaping Algorithm (SFLA) and CSA to reduce the knapsack problem when allocating resources in a cloud computing environment. SFLA algorithm processes user requests ranging from request size initialization to request generation and SFLA fitness value estimation to request sorting, splitting, and scoring. CSA is capable of operating new runtime solutions, including initializing, generating, and evaluating their fitness functions, as well as making changes. The proposed method was assessed and compared with the HABCCS, the task of GTS algorithm, and the krill herd algorithm. According to the experimental findings, existing algorithms outperformed SFLA-CS in terms of execution times, throughput, reaction times, and allocation percentages.

Javaid *et al.,* [64] suggested using a CSA with levy walk distribution (CLW) and flower pollination (FP) to balance the load and cut reaction, processing, and cost times in an integrated cloud and fog environment. The proposed algorithm has been evaluated and compared to CSA and BAT algorithms based on CDC, ORT, and RDL service broker of cloud analyst. According to the experiment's findings, CSA and BAT had slower response times, longer processing times, and higher costs. However, the processing time for a certain task is not taken into account.

Loheswaran *et al.,* [65] To speed up task execution and better utilize available resources in cloud computing, researchers created the task scheduling hybrid cuckoo search and iterative learning search (HCS-ILS) method. CS is used for exploration, while ILS was used for exploitation with four perturbation methods: swap, insertion, inversion, and displacement. The algorithm is implemented on Cloudsim; results show that HCS-ILS outperformed both CS and CCS compared to the quality of solution, accuracy, consistency, and faster convergence.

A hybrid cuckoo and particle swarm algorithm were developed by Jacob and Pradeep [66] to minimize cost, make span, and violation dateline. The algorithm's performance is assessed and contrasted with PBACO, ACO, MIN-MIN, and FCFS after being implemented on a cloud sim. Due to its ability to minimize costs, delays, and deadline violations, CPSO outperforms competing algorithms, according to simulation data. Vishal *et al.,* [53] proposed a combined approach employing CS and an ANN to enhance the energy-saving mechanism in the cloud environment through the classification of VMs to the use of the central processor unit (CPU). Virtual machines are sorted using a customized version of the best fit-decreasing method. Sadawarti and Saini [67] proposed a CS- optimized FFBPNN to improve the validity and authenticity of data stored, RSA is used with AES and TDES. Mangalampalli *et al.,* [68] proposed a hybridized PSO-CS algorithm for workflow scheduling on virtual machines. The CS randomly generates the initial population and applies levy flight to perform initial iterations. The PSO is then employed to compute local and global best solutions to achieve optimal solutions. The algorithm aimed at optimizing both makespan and power consumption outperformed the constituent PSO and CS. It was implemented in CloudSim.

Qi *et al.,* [69] a hybrid Markov-Cuckoo Search (MarCuS) method for load balancing. First, the Markov process model determines the migrated nodes from over-load to under-load nodes. Then, a cuckoo search is applied to generate the superior nest location. The result shows that MarCUS outperformed ACO and PSO as it realizes global traffic balancing and cloud service reliability. Sadawarti and Saini [67] combine CS with SVM and ANN to minimize energy usage, VM migrations, and SLA violation. CS is applied to identify the best VM; ANN is used to train the task at hand, while SVM determines how n-dimensional space is categorized. Evaluation results show that the new model outperformed CS-SVM and EA-ABC-SVM. It was implemented on Matlab. Table 4 shows a summary of work done using HCSA.

Table 4
Summary of related literature on the hybridized CSA

Reference	CSA Variants	Compared Algorithm	Findings	Limitation
Naik *et al.,* [7]	MOFCSA	MOGA, MOACO	Decreased energy consumption and unnecessary VM relocation. Ensure stable host selection	SLA violation, communication overhead is not considered
Raju *et al.,* [70]	ACO &CSA	ACO	Improved performance, reduced energy, and makespan	The number of tasks is less significant.
Shahdi-Pashaki *et al.,* [5]	GT &COA	Round Robin (RR)	shows better performance than RR, on both minor and major problems	The proposed model focus on cost only, it is compared with the only traditional algorithm
Aujla *et al.,* [60]	GA &CSA	FIFO, CS, GA	minimizes the execution time and maximizes resource utilization and consumes less energy	Limits the total job to 120
Durgadevi and Srinivasan [63]	SFLA-CS	HABCCS, GTS, and krill herd algorithm	Less computed times are consumed	The method concentrates on time only. Other parameter values like SLA negotiation are not considered

Pradeep and Jacob [24]	Cuckoo and gravitational search algorithm (CGSA)	GSA, CS, PSO, and GA	Shows minimal cost, less energy consumption, and less waste of resources	Focus on cloud provider profit only
Tavana *et al.,* [71]	GT &DCOA	FF, RR, GA	DCOA shows qualitative and effective superiority over FF, RR & GA	Focus on cost only
Pradeep and Jacob [62]	CHSA	CGSA, CS, HS	Minimum cost, least amount of memory used, least amount of energy used, least amount of penalty, and most credit	Doesn't consider parallelization which can improve the search ability and convergence speed
Krishnadoss and Jacob [6]	OCSA	PSO, GA, and IDEA	Minimize makespan and cost	Doesn't support real-time operations
Javaid *et al.,* [64]	CSLW & FP	CS & BAT	Decrease processing time, response time, and cost	Individual processing time is not considered
Mangalampalli *et al.,* [68]	PSO_CS	PSO & CSO	Minimized makespan and energy consumption	Findings in view of cloud vendors only
Qi *et al.,* [69]	Markov-Cuckoo Search (Marcus)	ACO &PSO	Meets requirement for load balancing and cloud service reliability	Concentrates on load balancing
Sadawarti and Saini [67]	CS-SVM-ANN	CS-SVM&EA-ABC-SVM	Minimized energy usage, number of VM migrations, exhibits least SLA violations.	Additional characteristics such as security and complexity can enhanced outcomes

2.5 Discrete Cuckoo Search Algorithm

Tavana *et al.,* [71] applied Group Technology (GT) and a discrete cuckoo optimization algorithm for server, virtual machine, and task consolidation to reduce power consumption, find the kind and number of VM running on servers, control task migration, penalty cost. The DCOA model was utilized to tackle larger problems where finding an optimal solution using exact methods is very demanding. Comparing DCOA to FF, GA, and RR with test instances of varying sizes, the results indicated that DCOA performs better than FF, RR, and GA in terms of cost. Table 5 presents a concise overview of research outcomes that assesses how the Discrete Cuckoo Search Algorithm (DCSA) and its variations perform in comparison to alternative optimization algorithms across various situations.

Table 5
Summary of related literature on the DCSA

Reference	CSA Variants	Compared Algorithm	Findings
Bibiks *et al.,* [72]	Discrete Cuckoo Search (DCS)	GA, PSO, PRB, SA	There is the same level of superiority and competitiveness in DCS to compared algorithms
Ouaarab *et al.,* [21]	DCS	PSO	Reduced the completion time of the last process of the schedule

3. Domain of Application

CSA has been employed successfully to optimize solutions across numerous fields of cloud computing, such as task scheduling, resource scheduling, virtual machine placement and load balancing (Table 6). Table 6 reveals the scope of application of CSA's application in cloud computing while Figure 2 illustrates a taxonomy of the domains that CSA is used in within cloud computing environment.

Table 6
Domain of Application of CSA in Cloud Computing

Reference	Domain
Sait *et al.,* [51]	Virtual machine placement/server consolidation
Aujla *et al.,* [60]	Task scheduling
Barlaskar *et al.,* [9]	Virtual machine Placement
Naik *et al.,* [7]	Virtual machine placement
Raju *et al.,* [70]	Task scheduling
Yakhchi *et al.,* [52]	Load balancing
Abbasi and Mohri [73]	Task scheduling
Nirmala and Veni [61]	Task scheduling
Dong-han and Hai-tao [49]	Task scheduling
Pradeep and Jacob [24]	Task scheduling
Madni and Latiff [12]	Resource scheduling
Pradeep and Jacob [62]	Task scheduling
Krishnadoss and Jacob [6]	Task scheduling
Durgadevi and Srinivasan [63]	Resource allocation
Javaid *et al.,* [64]	Load Balancing

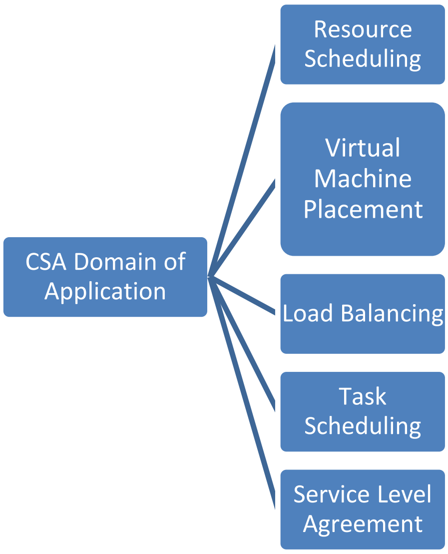

Fig. 2. Taxonomy of Domain of Application of CSA in Cloud Computing

4. Objective Functions

An objective function is a function that is desired to be minimized or maximized under certain constraints. It is what an optimization procedure uses to select better solutions over not-so-good solutions [74]. The parameters mostly used in cloud computing environment with CSA are listed in Table 7 and shown in Figure 3.

Table 7
Summary of Parameter Values used

Reference	Parameters
Yakhchi *et al.,* [52]	Power consumption, resource usage
Naik *et al.,* [7]	Communication cost, migration time, load balancing
Raju *et al.,* [70]	Energy consumption, makespan
Shahdi-Pashaki *et al.,* [5]	Cost of Transfer, penalty, Server and VM
Bibiks *et al.,* [72]	Makespan
Aujla *et al.,* [60]	Time, Resource utilization & energy consumption
Barlaskar *et al.,* [9]	Energy consumption VM migration cost, CPU utilization, and SLA breaches
Yakhchi *et al.,* [52]	Energy consumption, SLA violation
Durgadevi and Srinivasan [63]	Execution time, throughput, turnaround time, and allocation mechanism
Sarddar *et al.,* [48]	Resource wastage
Dong-han and Hai-tao [49]	Energy efficiency, resource utilization
Madni *et al.,* [55]	Response Time, Make Span, and throughput
Agarwal and Srivastava [11]	MakeSpan
George *et al.,* [15]	Computational Cost
Alexander and Joseph [50]	Makespan, Computational cost, CPU utilization & deadline constraint
Nirmala and Veni [61]	Execution time, throughput
Tavana *et al.,* [71]	Energy cost, penalty cost, Task migration cost
Singh and Randhawa [47]	Makespan, Execution cost
Krishnadoss and Jacob [6]	Makespan, cost
Pradeep and Jacob [24]	Cost, energy consumption, and memory usage
Pradeep and Jacob [62]	cost, energy, memory utilization, praise, and penalty

Krishnadoss and Jacob [6]	Makespan, Cost
Madni *et al.,* [55]	Makespan time, Cost &Utilization
Silambarasan and Kumar [46]	Execution time
Javaid *et al.,* [64]	Response time, processing time, and cost
Aloboud and Kurdi [45]	CPU Utilization, Turnaround time

5. Scheduling Parameters

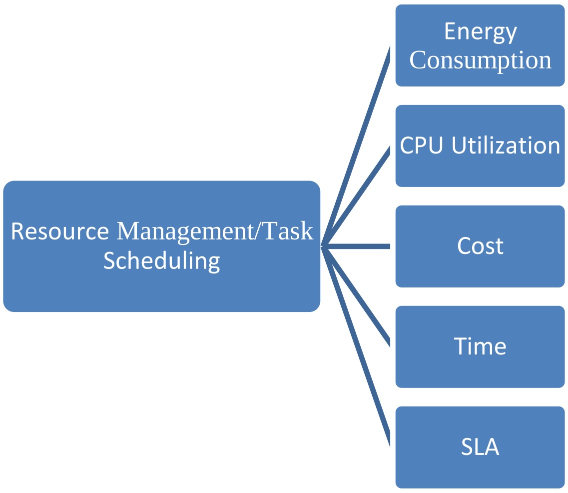

Fig. 3. Scheduling Parameters

6. Simulators/Platforms for Implementing CSA for Resource Provisioning in Cloud Computing Environment

It costs money to set up and operate an experiment in a genuine cloud environment. However, there are practical options that frequently offer acceptable substitutes for simulating cloud computing environments, such as modeling and simulation tool [75]. As cloud computing is scalable across n servers, analyzing the resource allocation in the environment requires a Cloud modeling and simulation tool that will build and manage the cloud as per requirement [76]. In addition, the tools provide the ability to evaluate the benchmarking application in a monitored environment. Simulators can be graphical user interface (GUI) simulators examples are Cloud Analyst, CloudReport, RealCloudSim, or language-based simulators (like Java, C++) examples: GridSim, CloudSim, WorkFlowSim, SimGrid, CloudSched [77]. The most common simulator used for resource management and provisioning in Cloud is CloudSim. It is an open- source simulation toolkit that allows for easy modeling, simulation, and testing of cloud computing models, services, and application delivery environments. It provides system and behavioral modeling of data centers, hosts, VMs, cloudlets, and resource allocation policies. It is more general and extensible without difficulty. It enables both single and interconnected cloud modeling and simulations [73,78]. Simulators that are used for implementation identified in our study are listed in Table 8.

Table 8
Summary of Simulators/Platform

Reference	Implementation Platforms	System Configuration
Naik *et al.,* [7]	CloudSim	PM CPU capacity 1000, 2000 or 3000 MIPS, RAM 0000MB, BW100000 bit/sec. VM CPU varied from 250,500,750 and 1000 RAM 128,512 and 1024, 2500, 5000 and 7500
Raju *et al.,* [70]	Xen Cloud Platform (XCP) Mathlab	Not specified
Ouaarab *et al.,* [21]	Java	2GHz Intel Core 2 Duo
Shahdi-Pashaki *et al.,* [5]	Lingo 9, MATLAB2010	2.50GHz CPU, Dual-core 2GB RAM
Bibiks *et al.,* [72]	Java	Windows 8.1 6bit Intel Core i72.4GHz CPU with 16 GB of RAM
Aujla *et al.,* [60]	C#	Window azure
Madni *et al.,* [13]	CloudSim	Linux with 512 RAM
Panda and Jana [56]	MATLAB R2012a	Windows 7 platform 4 GB RAM 2.20 GHz CPU
Sait *et al.,* [51]	MATLAB R2012b	3.5 GHz CPU 200GB RAM Windows 7
Yakhchi *et al.,* [52]	CloudSim	PM dual-core CPUs. HP ProLiant ML110 G4 &G5 servers VM single core.
Akbari and Rashidi [57]	C# language	2.2 GHz 6 GB RAM Core i7
Alexander and Joseph [50]	CloudSim JAVAwithCloudSim	Not specified Not specified
Dong-han and Hai-tao [49]	CloudSim	CPU 3100MIPS, 4 CORE, RAM 6G Storage 500G BW 10 CPU 2900 MIPS, 8CORE, RAM 10, Storage 500G, BW10
Tavana *et al.,* [71]	MATLAB.R2010	Win 7 2.50 GHz CPU 2 GB RAM,
Agarwal and Srivastava [11]	Cloud Sim	Not specified
Singh and Randhawa [47]	Workflows SIM	Not specified
George *et al.,* [15]	Not specified	Not specified
Nirmala and Veni [61]	CloudSim	3.40 GHz CPU 4GB RAM size
Barlaskar *et al.,* [9]	Cloud Sim	HP ProLiant ML110 G4/G5
Pradeep and Jacob [24]	JDK 1.6 & cloudSim 3.0	Windows 7, 2 GHz CPU 4 GB RAM
Pradeep and Jacob [62]	Java (JDK 1.6) with CloudSim	Windows 7 2 GHz
Madni *et al.,* [55]	Cloud sim	RAM 512MB,2048MB, Storage 1,000,000
Krishnadoss and Prem [6]	Java (JDK 1.6) & Cloudsim	WIN 7, 2 GHz dual-core, 4 GB RAM
Silambarasan and Kumar [46]	Mathlab	Not specified
Javaid *et al.,* [64]	Cloud analyst	Not specified
Aloboud and Kurdi [45]	Python and Haizea scheduler	Centos OS, 32GB HP ProLiant DL380p

Survey shows that different platforms were used for the implementation. The figure below shows the visual representation of the platforms used in different studies, and CloudSim is the most frequently used platform for implementing CSA in resource management/scheduling. Figure 4 shows the visual representation of simulators identified to be used in implementation.

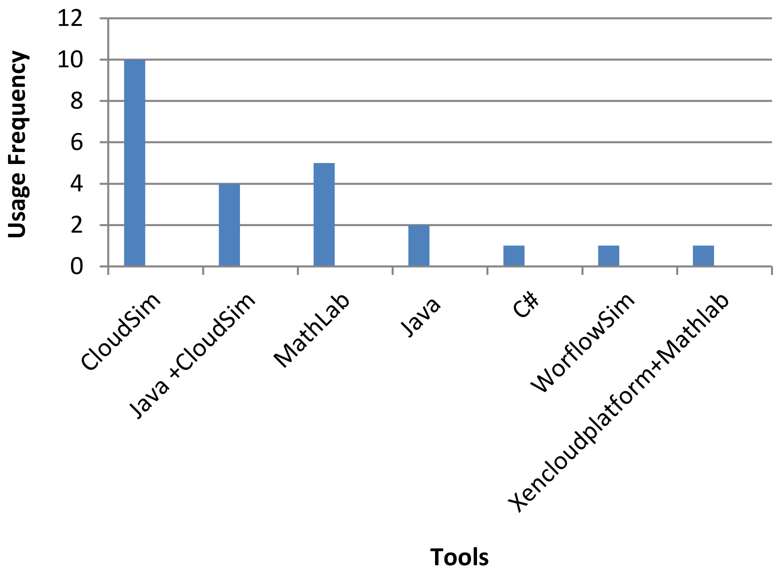

Fig. 4. Visual representation of the implementation platform

7. General Discussion of the Research Area

Scheduling is designed to balance the workloads and process cloud user requests quickly, so resources are strained and waiting time is reduced. Resource Allocation Algorithms play a vital role in this, as they determine which resources should be used for a given job. Traditional algorithms are no longer appropriate for cloud computing due its virtualized nature, so new algorithms must be improved and optimized to account for elasticity, scalability, increasing resources, and cost. Algorithms inspired by nature are very effective and efficient in solving complex problems as they are inspired by processes observed in nature. For example, Artificial Bee Colony mimics bee behavior, CSA is based on cuckoo birds' brood parasitism behavior, ACO uses ants' food-seeking behavior. Evolutionary theory was used to create GA [11]. Optimizing resource allocation in cloud computing has become a prominent topic recently; Researchers use different parameters to measure the performance of several algorithms in these environments such as execution cost, waiting time, Make Span, cost, power consumption, Utilization of CPU Time, throughput, memory usage, and migration time.

8. Challenges and Future Research Directions

This section outlined the problems associated with resource allotment/scheduling in the cloud. These problems include Task scheduling, QoS and SLA, key concerns for cloud service providers and users, and VM migration in multiple data centers.

i. Heterogeneity has been neglected at the resource level, even though some researchers consider it at the job level. Both need to be considered simultaneously.
ii. Numerous researchers have designed scheduling in one data center. New strategies for managing/scheduling a more significant number of virtual machines (VM) in multiple data centers should be investigated.
iii. Many researchers do not consider SLA negotiation between Cloud providers and Consumers. We can apply multi-agents-based parallel CSA to SLA negotiation and management in cloud environments with QoS based so resources can be situated and released based on consumer's needs and choice of a data center.

iv. Many researchers have worked on Multi-objective CSA scheduling workflows with different ways of computing objectives to find the best fitness function. There is still a need for improvement in real-time scheduling and the method of evaluation to improve results quality.
v. Improvement in VM allocation and migration method to minimize energy consumption and maximize resource usage.
vi. CSA-based load balancing in cloud computing: Load balancing is an important issue in cloud computing, and CSA can be used to optimize the allocation of resources to balance the workload across servers. This can improve the overall performance and efficiency of the cloud system.
vii. CSA-based energy-aware resource management and scheduling: Energy consumption is a critical issue in cloud computing, and CSA can be used to optimize resource allocation and scheduling to reduce energy consumption while maintaining performance.
viii. CSA-based fault-tolerant resource management and scheduling: CSA can be used to optimize resource allocation and scheduling in cloud computing systems to ensure fault tolerance and availability. This involves considering the failure of resources and optimizing resource allocation and scheduling to minimize downtime and data loss.

Overall, there are many potential research directions for the application of CSA in resource management and scheduling in cloud computing, and researchers can explore these directions to improve the performance, efficiency, and effectiveness of cloud computing systems.

9. Conclusions

This paper examined the potential of CSA in resource management and task scheduling in a cloud environment. Resource scheduling is a complicated task in the cloud, as limited resources must be shared between users. It is imperative that cloud service providers manage their resources efficiently to meet user needs. Research was conducted using CSAs within Cloud context, including basic CSA, modified CSA, or hybridized CSA with other algorithms to solve various problems, which includes, virtual machine placement, task scheduling, optimization to minimize parameters including cost, time, and energy, or maximized parameters such as throughput and resource utilization; However, there is limited research into CSA application for Service Level Agreement (SLA). Challenges were discussed and future research should be directed towards improving the use of CSA for resource management/task scheduling.

Acknowledgement
This work was supported and funded by the SIMAD University through its Center for Research and Development. Hence, the authors wish to express their profound gratitude to the entire SIMAD University management for finding their work worthy for funding.

References
[1] Chiroma, Haruna, Tutut Herawan, Iztok Fister Jr, Iztok Fister, Sameem Abdulkareem, Liyana Shuib, Mukhtar Fatihu Hamza, Younes Saadi, and Adamu Abubakar. "Bio-inspired computation: Recent development on the modifications of the cuckoo search algorithm." *Applied Soft Computing* 61 (2017): 149-173. https://doi.org/10.1016/j.asoc.2017.07.053
[2] Huth, Alexa, and James Cebula. "The basics of cloud computing." *United States Computer* (2011): 1-4.

[3] Vinothina, V. V., R. Sridaran, and Padmavathi Ganapathi. "A survey on resource allocation strategies in cloud computing." *International Journal of Advanced Computer Science and Applications* 3, no. 6 (2012). https://doi.org/10.14569/IJACSA.2012.030616
[4] Banu, A. Shahina, and W. R. Helen. "Scheduling deadline constrained task in hybrid IaaS cloud using cuckoo driven particle swarm optimization." *Indian Journal of Science and Technology* 8, no. 16 (2015): 1. https://doi.org/10.17485/ijst/2015/v8i16/61830
[5] Shahdi-Pashaki, S., Ehsan Teymourian, Vahid Kayvanfar, GH M. Komaki, and A. Sajadi. "Group technology-based model and cuckoo optimization algorithm for resource allocation in cloud computing." *IFAC-PapersOnLine* 48, no. 3 (2015): 1140-1145. https://doi.org/10.1016/j.ifacol.2015.06.237
[6] Krishnadoss, Pradeep, and Prem Jacob. "OCSA: Task Scheduling Algorithm in Cloud Computing Environment." *International Journal of Intelligent Engineering & Systems* 11, no. 3 (2018). https://doi.org/10.22266/ijies2018.0630.29
[7] Naik, B. Balaji, Dhananjay Singh, Arun B. Samaddar, and Sangsu Jung. "Developing a cloud computing data center virtual machine consolidation based on multi-objective hybrid fruit-fly cuckoo search algorithm." In *2018 IEEE 5G World Forum (5GWF)*, pp. 512-515. IEEE, 2018. https://doi.org/10.1109/5GWF.2018.8516947
[8] Sonkar, S. K., and M. U. Kharat. "A review on resource allocation and VM scheduling techniques and a model for efficient resource management in cloud computing environment." In *2016 International Conference on ICT in Business Industry & Government (ICTBIG)*, pp. 1-7. IEEE, 2016. https://doi.org/10.1109/ICTBIG.2016.7892646
[9] Barlaskar, Esha, Yumnam Jayanta Singh, and Biju Issac. "Enhanced cuckoo search algorithm for virtual machine placement in cloud data centres." *International Journal of Grid and Utility Computing* 9, no. 1 (2018): 1-17. https://doi.org/10.1504/IJGUC.2018.090221
[10] Zhan, Zhi-Hui, Xiao-Fang Liu, Yue-Jiao Gong, Jun Zhang, Henry Shu-Hung Chung, and Yun Li. "Cloud computing resource scheduling and a survey of its evolutionary approaches." *ACM Computing Surveys (CSUR)* 47, no. 4 (2015): 1-33. https://doi.org/10.1145/2788397
[11] Agarwal, Mohit, and Gur Mauj Saran Srivastava. "A cuckoo search algorithm-based task scheduling in cloud computing." In *Advances in Computer and Computational Sciences: Proceedings of ICCCCS 2016*, Volume 2, pp. 293-299. Springer Singapore, 2018. https://doi.org/10.1007/978-981-10-3773-3_29
[12] Madni, Syed Hamid Hussain, and Muhammad Shafie Abd Latiff. "Optimal Resource Scheduling for IaaS Cloud Computing using Cuckoo Search Algorithm." *Sains Humanika* 9, no. 1-3 (2017): 71-76. https://doi.org/10.11113/sh.v9n1-3.1145
[13] Madni, Syed Hamid Hussain, Muhammad Shafie Abd Latiff, Yahaya Coulibaly, and Shafi'i Muhammad Abdulhamid. "Resource scheduling for infrastructure as a service (IaaS) in cloud computing: Challenges and opportunities." *Journal of Network and Computer Applications* 68 (2016): 173-200. https://doi.org/10.1016/j.jnca.2016.04.016
[14] Fister, Iztok, Xin-She Yang, Dušan Fister, and Iztok Fister. "Cuckoo search: a brief literature review." *Cuckoo Search and Firefly Algorithm: Theory and Applications* (2014): 49-62. https://doi.org/10.1007/978-3-319-02141-6_3
[15] George, Neenu, K. Chandrasekaran, and A. Binu. "Optimization-aware scheduling in cloud computing." In *Proceedings of the International Conference on Informatics and Analytics*, pp. 1-5. 2016. https://doi.org/10.1145/2980258.2980293
[16] Rajasekar, B., and S. K. Manigandan. "An efficient resource allocation strategies in cloud computing." *International Journal of Innovative Research in Computer and Communication Engineering* 3, no. 2 (2015): 1239-1244.
[17] Sharma, Shivani, and Dhanshri Parihar. "A review on resource allocation in cloud computing." *International Journal of Advance Research, Ideas, and Innovations in Technology* 1, no. 3 (2014): 1-7.
[18] Asha, N., and G. Raghavendra Rao. "A review on various resource allocation strategies in cloud computing." *International Journal of Emerging Technology and Advanced Engineering (IJETAE)* 3, no. 7 (2013).
[19] Kaur, Harminder, Harsukhpreet Singh, and Anurag Sharma. "Geographic routing protocol: A review." *International Journal of Grid and Distributed Computing* 9, no. 2 (2016): 245-254. https://doi.org/10.14257/ijgdc.2016.9.2.21
[20] Akintoye, Samson B., and Antoine Bagula. "Optimization of virtual resources allocation in cloud computing environment." In *2017 IEEE AFRICON*, pp. 873-880. IEEE, 2017. https://doi.org/10.1109/AFRCON.2017.8095597
[21] Ouaarab, Aziz, Belaïd Ahiod, Xin-She Yang, and Mohammed Abbad. "Discrete cuckoo search algorithm for job shop scheduling problem." In *2014 IEEE International Symposium on Intelligent Control (ISIC)*, pp. 1872-1876. IEEE, 2014. https://doi.org/10.1109/ISIC.2014.6967636
[22] Yang, Xin-She. *Nature-inspired metaheuristic algorithms*. Luniver Press, 2010.
[23] Karwayun, Ms Ragini, H. S. Sharma, and K. P. Yadav. "Challenges in Dynamic Resource Allocation and Task Scheduling in Heterogeneous Clouds." *International Journal on Recent and Innovation Trends in Computing and Communication* 5, no. 7 (2017): 93-97.

[24] Pradeep, K., and T. Prem Jacob. "CGSA scheduler: A multi-objective-based hybrid approach for task scheduling in cloud environment." *Information Security Journal: A Global Perspective* 27, no. 2 (2018): 77-91. https://doi.org/10.1080/19393555.2017.1407848

[25] Yang, Xin-She, and Suash Deb. "Cuckoo search via Lévy flights." In *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, pp. 210-214. IEEE, 2009. https://doi.org/10.1109/NABIC.2009.5393690

[26] Singh, Anuj Kumar. "Evaluation of DSR Routing Protocol Embedded with a Code to Improve the Performance of Movement of Nodes in MANET Using NS-2." *International Journal of Advanced Research in Computer Science and Software Engineering* 4, no. 9 (2014): 138-143.

[27] Fister Jr, Iztok, Dušan Fister, and Iztok Fister. "A comprehensive review of cuckoo search: variants and hybrids." *International Journal of Mathematical Modelling and Numerical Optimisation* 4, no. 4 (2013): 387-409. https://doi.org/10.1504/IJMMNO.2013.059205

[28] Kamat, Sanket, and Asha Gowda Karegowda. "A brief survey on cuckoo search applications." *International Journal of Innovative Research in Computer and Communication Engineering* 2, no. 2 (2014): 7-14.

[29] Shehab, Mohammad, Ahamad Tajudin Khader, and Mohammed Azmi Al-Betar. "A survey on applications and variants of the cuckoo search algorithm." *Applied Soft Computing* 61 (2017): 1041-1059. https://doi.org/10.1016/j.asoc.2017.02.034

[30] Yang, Xin-She, and Suash Deb. "Cuckoo search: recent advances and applications." *Neural Computing and Applications* 24 (2014): 169-174. https://doi.org/10.1007/s00521-013-1367-1

[31] Ouaarab, Aziz, Belaïd Ahiod, and Xin-She Yang. "Discrete cuckoo search algorithm for the travelling salesman problem." *Neural Computing and Applications* 24 (2014): 1659-1669. https://doi.org/10.1007/s00521-013-1402-2

[32] Walton, Sean, Oubay Hassan, Kenneth Morgan, and M. Rowan Brown. "A review of the development and applications of the Cuckoo search algorithm." *Swarm Intelligence and Bio-Inspired Computation* (2013): 257-271. https://doi.org/10.1016/B978-0-12-405163-8.00011-9

[33] Valian, Ehsan, Shahram Mohanna, and Saeed Tavakoli. "Improved cuckoo search algorithm for feedforward neural network training." *International Journal of Artificial Intelligence & Applications* 2, no. 3 (2011): 36-43. https://doi.org/10.5121/ijaia.2011.2304

[34] Zhao, Jie, Xiujuan Lei, Zhenqiang Wu, and Ying Tan. "Clustering using improved cuckoo search algorithm." In *Advances in Swarm Intelligence: 5th International Conference, ICSI 2014*, Hefei, China, October 17-20, 2014, Proceedings, Part I 5, pp. 479-488. Springer International Publishing, 2014. https://doi.org/10.1007/978-3-319-11857-4_54

[35] Piechocki, Janusz, Dominik Ambroziak, Aleksander Palkowski, and Grzegorz Redlarski. "Use of modified cuckoo search algorithm in the design process of integrated power systems for modern and energy self-sufficient farms." *Applied Energy* 114 (2014): 901-908. https://doi.org/10.1016/j.apenergy.2013.07.057

[36] Boushaki, Saida Ishak, Nadjet Kamel, and Omar Bendjeghaba. "Improved cuckoo search algorithm for document clustering." In *Computer Science and Its Applications: 5th IFIP TC 5 International Conference, CIIA 2015*, Saida, Algeria, May 20-21, 2015, Proceedings 5, pp. 217-228. Springer International Publishing, 2015. https://doi.org/10.1007/978-3-319-19578-0_18

[37] Rodrigues, Douglas, Luis AM Pereira, T. N. S. Almeida, João Paulo Papa, A. N. Souza, Caio CO Ramos, and Xin-She Yang. "BCS: A binary cuckoo search algorithm for feature selection." In *2013 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 465-468. IEEE, 2013. https://doi.org/10.1109/ISCAS.2013.6571881

[38] Durgun, Ismail, and Ali R. Yildiz. "Structural design optimization of vehicle components using cuckoo search algorithm." *Materials Testing* 54, no. 3 (2012): 185-188. https://doi.org/10.3139/120.110317

[39] Maurya, Lalit, Prasant Kumar Mahapatra, and Garima Saini. "Modified cuckoo search-based image enhancement." In *Proceedings of the 4th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA)* 2015, pp. 625-634. Springer India, 2016. https://doi.org/10.1007/978-81-322-2695-6_53

[40] Sanajaoba, Sarangthem, and Eugene Fernandez. "Maiden application of Cuckoo Search algorithm for optimal sizing of a remote hybrid renewable energy System." *Renewable Energy* 96 (2016): 1-10. https://doi.org/10.1016/j.renene.2016.04.069

[41] Yildiz, Ali R. "Cuckoo search algorithm for the selection of optimal machining parameters in milling operations." *The International Journal of Advanced Manufacturing Technology* 64 (2013): 55-61. https://doi.org/10.1007/s00170-012-4013-7

[42] Sopa, Mana, and Niwat Angkawisittpan. "An application of cuckoo search algorithm for series system with cost and multiple choices constraints." *Procedia Computer Science* 86 (2016): 453-456. https://doi.org/10.1016/j.procs.2016.05.079

[43] Salami, Hamza Onoruoiza, Abubakar Bala, Sadiq M. Sait, and Idris Ismail. "An energy-efficient cuckoo search algorithm for virtual machine placement in cloud computing data centers." *The Journal of Supercomputing* 77 (2021): 13330-13357. https://doi.org/10.1007/s11227-021-03807-3

[44] Kurokawa, Takeru, and Naohiro Hayashibara. "Content placement using Cuckoo search in Cloud-based Content delivery networks." *Internet of Things* 16 (2021): 100430. https://doi.org/10.1016/j.iot.2021.100430
[45] Aloboud, Ebtesam, and Heba Kurdi. "Cuckoo-inspired job scheduling algorithm for cloud computing." *Procedia Computer Science* 151 (2019): 1078-1083. https://doi.org/10.1016/j.procs.2019.04.153
[46] Silambarasan, K., and P. Kumar. "An improved cuckoo search algorithm for system efficiency in cloud computing." In *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC) I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), 2018 2nd International Conference on*, pp. 733-736. IEEE, 2018. https://doi.org/10.1109/I-SMAC.2018.8653706
[47] Singh, Harshpreet, and Rajneesh Randhawa. "Cuckoo search based workflow scheduling on heterogeneous cloud resources." In *2017 7th International Conference on Cloud Computing, Data Science & Engineering-Confluence*, pp. 65-70. IEEE, 2017. https://doi.org/10.1109/CONFLUENCE.2017.7943125
[48] Sarddar, Debabrata, Sougata Chakraborty, and Priyajit Sen. "Parallelism in CPU virtualization and Scheduling using Cuckoo Search Algorithm." *International Journal of Grid and Distributed Computing* 10, no. 6 (2017): 1-10. https://doi.org/10.14257/ijgdc.2017.10.6.01
[49] Dong-han, Jiang, and Lin Hai-tao. "Research on a resource allocation algorithm based on energy conservation." In *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, pp. 569-572. IEEE, 2017. https://doi.org/10.1109/CompComm.2017.8322609
[50] Alexander, Aneena Ann, and Divya Lissia Joseph. "An efficient resource management for prioritized users in cloud environment using cuckoo search algorithm." *Procedia Technology* 25 (2016): 341-348. https://doi.org/10.1016/j.protcy.2016.08.116
[51] Sait, Sadiq M., Abubakar Bala, and Aiman H. El-Maleh. "Cuckoo search based resource optimization of datacenters." *Applied Intelligence* 44 (2016): 489-506. https://doi.org/10.1007/s10489-015-0710-x
[52] Yakhchi, Moona, Seyed Mohssen Ghafari, Shahpar Yakhchi, Mahdi Fazeli, and Ahmad Patooghi. "Proposing a load balancing method based on Cuckoo Optimization Algorithm for energy management in cloud computing infrastructures." In *2015 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*, pp. 1-5. IEEE, 2015. https://doi.org/10.1109/ICMSAO.2015.7152209
[53] Vishal, Vishal, Bikrampal Kaur, and Surender Jangra. "Cuckoo-Neural Approach for Secure Execution and Energy Management in Mobile Cloud Computing." *International Journal of Advanced Computer Science and Applications* 12, no. 1 (2021): 654-663. https://doi.org/10.14569/IJACSA.2021.0120175
[54] Navimipour, Nima Jafari, and Farnaz Sharifi Milani. "Task scheduling in the cloud computing based on the cuckoo search algorithm." *International Journal of Modeling and Optimization* 5, no. 1 (2015): 44. https://doi.org/10.7763/IJMO.2015.V5.434
[55] Madni, Syed Hamid Hussain, Muhammad Shafie Abd Latiff, Javed Ali, and Shafi'I. Muhammad Abdulhamid. "Multi-objective-oriented cuckoo search optimization-based resource scheduling algorithm for clouds." *Arabian Journal for Science and Engineering* 44 (2019): 3585-3602. https://doi.org/10.1007/s13369-018-3602-7
[56] Panda, Sanjaya K., and Prasanta K. Jana. "A multi-objective task scheduling algorithm for heterogeneous multi-cloud environment." In *2015 International Conference on Electronic Design, Computer Networks & Automated Verification (EDCAV)*, pp. 82-87. IEEE, 2015. https://doi.org/10.1109/EDCAV.2015.7060544
[57] Akbari, Mehdi, and Hassan Rashidi. "A multi-objectives scheduling algorithm based on cuckoo optimization for task allocation problem at compile time in heterogeneous systems." *Expert Systems with Applications* 60 (2016): 234-248. https://doi.org/10.1016/j.eswa.2016.05.014
[58] Saif, Mufeed Ahmed Naji, S. K. Niranjan, and Belal Abdullah Hezam Murshed. "Multi-Objective Cuckoo Search Optimization Algorithm for Optimal Resource Allocation in Cloud Environment." In *2022 3rd International Conference for Emerging Technology (INCET)*, pp. 1-7. IEEE, 2022. https://doi.org/10.1109/INCET54531.2022.9823985
[59] Elnahary, M. Kh, Ahmed Y. Hamed, and H. El-Sayed. "Task Scheduling Optimization in cloud computing by Cuckoo Search Algorithm." *Sohag Journal of Sciences* 7, no. 3 (2022): 29-37.
[60] Aujla, Sumandeep, Amandeep Ummat, C. G. C. Landran, and Punjab Mohali. "Task scheduling in cloud using hybrid cuckoo algorithm." *International Journal of Computer Networks and Applications (IJCNA)* 2, no. 3 (2015): 144-150.
[61] Nirmala, A. P., and S. Veni. "A Hybrid Scheduling Algorithm to improve the Performance using Ant Colony Optimization and Cuckoo Search Algorithm with K-means++ in a Virtualized Environment." *International Journal of Advanced Research in Computer Science* 7, no. 4 (2016).
[62] Pradeep, K., and T. Prem Jacob. "A hybrid approach for task scheduling using the cuckoo and harmony search in cloud computing environment." *Wireless Personal Communications* 101 (2018): 2287-2311. https://doi.org/10.1007/s11277-018-5816-0

[63] Durgadevi, P., and S. Srinivasan. "Resource allocation in cloud computing using SFLA and cuckoo search hybridization." *International Journal of Parallel Programming* 48 (2020): 549-565. https://doi.org/10.1007/s10766-018-0590-x
[64] Javaid, Nadeem, Ayesha Anjum Butt, Kamran Latif, and Amjad Rehman. "Cloud and fog based integrated environment for load balancing using cuckoo levy distribution and flower pollination for smart homes." In *2019 International Conference on Computer and Information Sciences (ICCIS)*, pp. 1-6. IEEE, 2019. https://doi.org/10.1109/ICCISci.2019.8716467
[65] Loheswaran, K., T. Daniya, and K. Karthick. "Hybrid cuckoo search algorithm with iterative local search for optimized task scheduling on cloud computing environment." *Journal of Computational and Theoretical Nanoscience* 16, no. 5-6 (2019): 2065-2071. https://doi.org/10.1166/jctn.2019.7851
[66] Jacob, T. Prem, and K. Pradeep. "A multi-objective optimal task scheduling in cloud environment using cuckoo particle swarm optimization." *Wireless Personal Communications* 109 (2019): 315-331. https://doi.org/10.1007/s11277-019-06566-w
[67] Sadawarti, Kanav, and Satish Saini. "Nature Inspired Virtual Machine Allocation and Migration under Cloud Environment." *Advances and Applications in Mathematical Sciences* 21, no. 8 (2022): 4733-4746.
[68] Mangalampalli, Sudheer, Kiran Sree Pokkuluri, Raviteja Kocherla, Anuj Rapaka, and Narayana Rao Kota. "An Efficient Workflow Scheduling Algorithm in Cloud Computing Using Cuckoo Search and PSO Algorithms." In *Innovations in Computer Science and Engineering: Proceedings of the Ninth ICICSE*, 2021, pp. 137-145. Singapore: Springer Singapore, 2022. https://doi.org/10.1007/978-981-16-8987-1_15
[69] Qi, Bin, Pan Zhang, Hong Wu, and Miao Yan. "Cloud Resource Scheduling Method based on Markov Process and the Cuckoo Search." In *Journal of Physics: Conference Series*, vol. 2320, no. 1, p. 012030. IOP Publishing, 2022. https://doi.org/10.1088/1742-6596/2320/1/012030
[70] Raju, R., R. G. Babukarthik, D. Chandramohan, P. Dhavachelvan, and T. Vengattaraman. "Minimizing the makespan using Hybrid algorithm for cloud computing." In *2013 3rd IEEE International Advance Computing Conference (IACC)*, pp. 957-962. IEEE, 2013. https://doi.org/10.1109/IAdCC.2013.6514356
[71] Tavana, Madjid, Saleh Shahdi-Pashaki, Ehsan Teymourian, Francisco J. Santos-Arteaga, and Mohammad Komaki. "A discrete cuckoo optimization algorithm for consolidation in cloud computing." *Computers & Industrial Engineering* 115 (2018): 495-511. https://doi.org/10.1016/j.cie.2017.12.001
[72] Bibiks, Kirils, Fun Hu, Jian-Ping Li, and Aleister Smith. "Discrete cuckoo search for resource constrained project scheduling problem." In *2015 IEEE 18th International Conference on Computational Science and Engineering*, pp. 240-245. IEEE, 2015. https://doi.org/10.1109/CSE.2015.39
[73] Abbasi, Mohammad Javad, and Mehrdad Mohri. "Scheduling tasks in the cloud computing environment with the effect of Cuckoo optimization algorithm." *SSRG International Journal of Computer Science and Engineering (SSRG-IJCSE)* 3, no. 8 (2016): 1-9. https://doi.org/10.14445/23488387/IJCSE-V3I8P101
[74] King, Alan J., and Stein W. Wallace. "Modeling the Objective Function." *Modeling with Stochastic Programming* (2012): 61-76. https://doi.org/10.1007/978-0-387-87817-1_3
[75] Bahwaireth, Khadijah, Lo'ai Tawalbeh, Elhadj Benkhelifa, Yaser Jararweh, and Mohammad A. Tawalbeh. "Experimental comparison of simulation tools for efficient cloud and mobile cloud computing applications." *EURASIP Journal on Information Security* 2016 (2016): 1-14. https://doi.org/10.1186/s13635-016-0039-y
[76] Ranbhise, S. M., and K. K. Joshi. "Simulation and analysis of cloud environment." *International Journal of Advanced Research in Computer Science & Technology (IJARCST 2014)* 2, no. 4 (2014): 206-209.
[77] Ahmed, Arif, and Abadhan Saumya Sabyasachi. "Cloud computing simulators: A detailed survey and future direction." In *2014 IEEE International Advance Computing Conference (IACC)*, pp. 866-872. IEEE, 2014. https://doi.org/10.1109/IAdCC.2014.6779436
[78] Calheiros, Rodrigo N., Rajiv Ranjan, Anton Beloglazov, César A. F. De Rose, and Rajkumar Buyya. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." *Software: Practice and Experience* 41, no. 1 (2011): 23-50. https://doi.org/10.1002/spe.995