

Research Article

Hybrid Electro Search with Ant Colony Optimization Algorithm for Task Scheduling in a Sensor Cloud Environment for Agriculture Irrigation Control System

Murali Subramanian ¹, Manikandan Narayanan ¹, B. Bhasker ¹, S. Gnanavel ²,
Md Habibur Rahman ³ and C. H. Pradeep Reddy ⁴

¹School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu, India

²Department of Computing Technologies, SRM Institute of Science and Technology, Chengalpattu, Tamil Nadu, India

³Department of Computer Science and Engineering, Faculty of Engineering and Technology, Islamic University, Kushtia 7003, Bangladesh

⁴School of Computer Science and Engineering, VIT-AP University, Amaravati, Andhra Pradesh, India

Correspondence should be addressed to Md Habibur Rahman; habib@iu.ac.bd

Received 11 June 2022; Accepted 5 August 2022; Published 4 October 2022

Academic Editor: Muhammad Ahmad

Copyright © 2022 Murali Subramanian et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Integrating cloud computing with wireless sensor networks creates a sensor cloud (WSN). Some real-time applications, such as agricultural irrigation control systems, use a sensor cloud. The sensor battery life in sensor clouds is constrained. The data center's computers consume a lot of energy to offer storage in the cloud. The emerging sensor cloud technology-enabled virtualization. Using a virtual environment has many advantages. However, different resource requirements and task execution cause substantial performance and parameter optimization issues in cloud computing. In this study, we proposed the hybrid electro search with ant colony optimization (HES-ACO) technique to enhance the behavior of task scheduling, for those considering parameters such as total execution time, cost of the execution, makespan time, the cloud data center energy consumption like throughput, response time, resource utilization task rejection ratio, and deadline constraint of the multicloud. Electro search and the ant colony optimization algorithm are combined in the proposed method. Compared to HESGA, HPSOGA, AC-PSO, and PSO-COGENT algorithms, the created HES-ACO algorithm was simulated at CloudSim and found to optimize all parameters.

1. Introduction

Food and agriculture are both important sources of income for many farmers worldwide. Irrigation is one of the most critical services supplied in agriculture. Most crops require irrigation in areas with low rainfall, as inadequate irrigation reduces crop quality and yield. Due to contemporary concerns such as water shortages, droughts, and resource scarcity, academics have tried to rationalize water usage in agriculture, one of the world's most water-intensive industries [1]. A large amount of water is required by the conventional irrigation method, resulting in water waste. The IIS is desperately needed to reduce water waste. In the

wheat field, IoT sensors capture exact ground and environmental data. The collected data is sent to a cloud-based server, which analyzes and advises farmers on irrigation. This recommendation system has an embedded feedback mechanism to make it robust and flexible [2]. Sensor-cloud technology integrates WSNs with cloud computing to reduce storage, processing, and scalability issues. Sensor cloud technology has recently been deployed to several real-world applications, including agriculture irrigation [3]. Sensor-cloud technology integrates WSNs with cloud computing to reduce storage, processing, and scalability issues. Sensor cloud technology has recently been deployed to several real-world applications, including agriculture irrigation [4]. A

sensor cloud is a collection of WSNs that provide sensing as a service to various applications. Efficiently managing task requests from many applications is crucial [5]. Combining WSNs and cloud capabilities will offer good services provided by sensor cloud infrastructure. In a sensor network, huge volume of data is transported to the gateway, which is present in the cloud to offer such services [6].

Figure 1 clearly shows the overall architecture of the sensor cloud environment. Sensor networks act as a link between virtual and physical worlds. These SNs are made up of micro-electro-mechanical nodes that can detect their environment and communicate. A sensor cloud is a group of WSNs with several sensors. It is a heterogeneous environment and allows customers to purchase and use cloud services.

Because of this, large-scale networks benefit greatly from the cost-effectiveness and affordability of cloud computing. Similar to cloud computing, SCs allow for the dynamic provisioning of resources in response to demand, enhancing operational flexibility. Performance and dependability in the sensor cloud are determined by several factors: scheduling, which includes resource, job, workflow, task, and deadline scheduling, among others. Task scheduling maximizes resource usage and ensures that activities are completed in the most efficient manner possible, resulting in a satisfactory final result for customers. In cloud computing, task scheduling refers to the process of allocating a task to available virtual machines in order to perform it as quickly as possible [7].

This research concentrates on energy-efficient task scheduling, especially communication between the cloud and users in a sensor cloud environment for agriculture irrigation control systems. Here, user requests are treated as a task for accessing the required on-demand information from the cloud. Clouds provide users with virtual resources. Users use virtual resources to perform tasks. Cloud computing's elasticity supports multiple virtual apps at once. Multiobjective computers exchange resources. User requests determine resource allocation. An efficient scheduler module checks and issues resource status. A good scheduler is essential for optimal cloud computing real-time performance. The task scheduling algorithm maps virtual machines with the tasks in the cloud. It uses available resources, reducing request latency and response time while increasing resource utilization and system throughput.

The processing needs and features of tasks in cloud computing differ. Scheduling tasks for cloud computing is an unsolved NP-complete issue. Hybrid, heuristic, and metaheuristic approaches are preferred, which poor the performance of systems and service providers impacted by task scheduling. Task scheduling is challenging due to the task's nature, the variety of the cloud's resources, utility, and deadline restrictions. Among the best task scheduling methods are FCFS, PSO, MIN-MIN, ACO, GA, and others. The greatest remedy for cloud computing task scheduling problems is metaheuristic algorithms. Although it is not declared to be optimal, using heuristic techniques offers the best optimal solution. By combining both methods, the metaheuristic technique and the heuristic approach create a hybrid scheduling algorithm solution.

The survey provides Electro Search (ES) and ACO techniques for scheduling the Cloudlet in the virtual machine with balanced cloudlet distribution. This study offers a hybrid task scheduling method based on the electro search (ES) and ACO techniques. The job scheduling issue was typically solved using ES and ACO-based algorithms. In order to offer the cone to the global solution to the best solution in the search space, the authors apply the ES algorithm for the global search strategy. The VM is the most important asset in a cloud environment, and the ACO technique helps maximize VM use. The goal of this research is to lessen the deadline limitations, makespan, execution cost, overall execution time, energy consumption, and throughput. The legislative cloud customers are served smoothly by the sensor cloud services provider (SCSP), who also makes the greatest money. Typically, SCSP receives a variety of tasks as requests. Additionally, these jobs need to be organized based on the requirements and any limitations. The authors presented the HMTS algorithm after they examined the case.

2. Related Works

Task scheduling in the sensor cloud environment, particularly in cloud computing, is a significant difficulty because of the varied nature of cloud resources. Numerous real-time approaches have been put out to address the task scheduling issue in the context of cloud computing. The scheduling issue has been solved using a variety of task scheduling strategies, each of which has its own advantages and disadvantages as a result of the many QoS parameters. Scheduling strategies are generally divided into heuristic, metaheuristic, and algorithmic categories. Heuristic strategies rely on prediction to arrive at the best answer with minor complexity and the shortest time. Rather than heuristic procedures, metaheuristic strategies identify the most efficient solution results. Below are some of the most recent task scheduling approaches.

A study explored energy-efficient sensor cloud environment approaches. An energy-efficient algorithm detects a change in the environment. Most of the research looked at did not address QoS, scalability, or network longevity with energy efficiency. Real-time applications in agriculture, healthcare, and intelligent homes need maximizing QoS and energy. This study will help academics build better techniques that consider QoS measurements and energy [8].

Ojha et al. suggested a dynamic duty scheduling technique for on-field sensor networks to reduce energy usage. The sensor cloud framework helps field WSNs reduce processing demand. The authors demonstrated a suitable time interval selection technique for data uploading to the cloud in duty time intervals. The plan proposes improving energy efficiency while cutting expenses. The proposed method outperformed traditional energy efficiency, network longevity, cost-effectiveness, and utility. Real-time sensor cloud applications require testing. Consider QoS guarantee as a parameter [9].

Sivakumar and Al-Anbary proposed a CMSP for IoT-based sensor cloud systems. The approach divides a dense

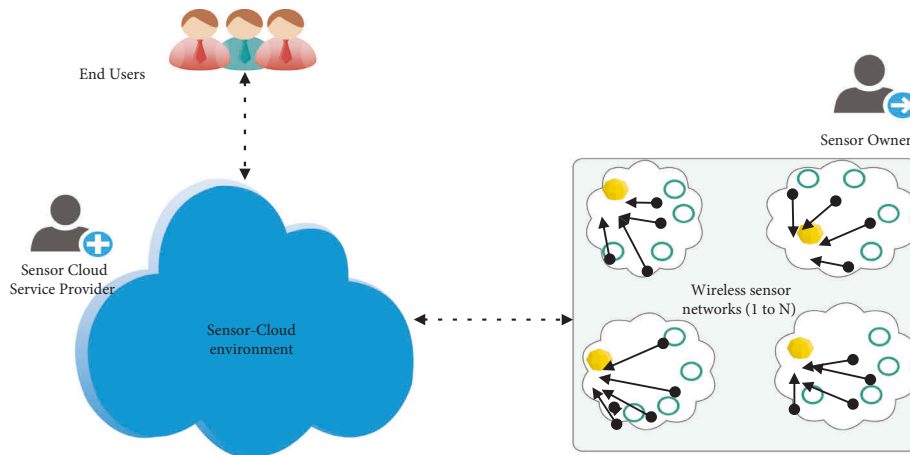


FIGURE 1: Sensor cloud architecture.

network into Voronoi structures. Each Voronoi division has its channel and data collector. They wanted to design a multichannel hybrid cluster protocol. It was made for static networks. ACMSP outperforms contemporary protocols such as MC-LMAC, MMSN, and TMCP in energy efficiency and throughput (Castalia tool). With IEEE 802.15.4, the proposed CMSP received intra- and intercluster data. The proposed CMSP must be tested in real time with real-time results for showing its effectiveness [10].

Chatterjee et al. optimized the selection algorithm for choosing suitable bridge nodes to reduce data transmission energy from sensor networks to sensor clouds. Research is focused on developing a multihop data transport system from PSNs to sensor clouds. One alternative was to save energy. Consider node heterogeneity, mobility, and other network factors to test real-time apps [11].

SC-iPaaS was proposed by Phan et al. SC-iPaaS has three layers: cloud, sensor, and edge. Between the layers, it uses push-pull communication. This study presents a Pareto optimal solution in the objective space. It uses evolutionary multiobjective optimization for its communication optimizer. The multiobjective analysis is critical in simulations. The results show that the proposed method obtains less bandwidth and energy consumption for more data yield. It is required to consider the real-time application and not QoS parameters [7].

Dinh et al. developed an effective sensor cloud interaction model to suit end-user QoS needs. We built a feedback control system to suit QoS requirements between physical WSN and sensor cloud. Optimize sensor energy utilization via feedback. This approach reduces latency compared to traditional protocols. A real-time application should be tested. Energy efficiency is a research issue—low signaling overhead [12].

A hybrid optimization model for effective VM task allocation was presented by Sreenivasulu and Paramasivam. The hybrid algorithm prioritizes workloads using cloud hierarchy and setting priorities using BAT/BAR models and VM traits. MOML preemption lowered VM burden. The simulation findings suggest that the suggested hybrid model outperformed existing

algorithms such as BAT and ACO. The hybrid technique has proven beneficial in utilizing bandwidth and memory. The hybrid algorithm prioritizes workloads using cloud hierarchy and setting priorities using BAT/BAR models, and VM traits reduced VM burden. By using a hybrid strategy, you can get more. The author ignores energy usage and must demonstrate the suggested algorithm's efficiency with real-time workflows [13].

To set up cloud-based task scheduling and to address the complex work scheduling issue, the author developed a hybrid AC-PSO method. The hybrid strategy that is suggested combines ACO and PSO (PSO). Using the proposed methods, jobs are successfully distributed to cloud-based virtual computers. The metrics Makespan time, cost, and resource utilization are superior to the proposed method. The author disregards time complexity, throughput, and energy efficiency [14].

Dubey and Sharma created a hybrid CR-PSO technique to improve PSO restrictions and handle task scheduling. The author created a mathematical model of work scheduling and specified its purpose and fitness function. Using both CRO and PSO, a hybrid CR-PSO strategy is created. Experiments show that the suggested algorithm is faster, cheaper, and shorter. Cost-effective CR-PSO outputs increase cloud system performance. The author scheduled dependent tasks and checked the proposed approach's energy usage, load balancing, task rejection ratio, and turnaround time [15].

Task scheduling played the key role in cloud and sensor cloud systems, according to Proshikshya Mukherjee. The author looked into the various challenges and difficulties associated with task scheduling issues in the sensor cloud. The ability to efficiently manage and schedule duties is the most critical component of this job. As a result of this investigation, the researchers will obtain a lot of knowledge regarding task scheduling. Designing an efficient hybrid task scheduling mechanism in this work is required. In the sensor cloud, multicriteria decision-making is needed [16].

In order to solve the task scheduling problem in cloud computing, a MOPSO algorithm (multiobjective particle

swarm optimization) proposed by Jena. The proposed solution was suited for the cloud environment since it used system resources to reduce energy consumption and time to completion. The designed MOPSO outperformed both the BRS and the RSA, according to simulation findings. This effort necessitated more bandwidth, load balancing, and cost, among other things. A more robust algorithm is required [17].

Swagatika et al. conducted a complete examination of conventional scheduling algorithms for efficient VM allocation in a cloud context. A modified Markov chain model is utilized to anticipate resource utilization. An upgraded PSO algorithm is employed for efficient resource allocation optimally in the cloud, with dynamic load balancing based on the VM allocation mechanism and required to focus on a more robust algorithm. Work needed to consider the many characteristics such as makespan time, energy consumption, and cost [18].

Nayak et al. proposed a novel approach for deadline-based work scheduling. Every lease has a current (CT) and gap time (GT). Previous methods did not include CT and GT as scheduling criteria. The proposed scheduling method is based on the lease acceptance rate. The proposed mechanism eliminates the necessity for a decision-maker such as AHP to resolve lease issues. In MATLAB R2015a, 10 different workloads are simulated and assessed. The proposed technique determined the average task rejection and acceptance ratios. The new mechanism outperforms the current backfilling process. Consider aspects such as VM switching costs, energy consumption, and makespan time for further work [19].

The author proposed using a hybrid cuckoo and particle swarm optimization technique to schedule jobs with multiple objectives. This task scheduling approach can achieve a close-to-optimal solution in a heterogeneous cloud environment because to the qualities of the recommended CPSO technique, which include being immediately convergent and simple to apply. When compared to existing algorithms such as ACO Min-Min, PBACO, and FCFS, the suggested CPSO approach obtains the lowest rate of deadline violations. Energy usage and other QoS service elements need to be taken into account [20].

Huang et al. developed a PSO algorithm with time-varying inertia weights for cloud work scheduling. This paper proposes a PSO-based scheduler with five update methods (i.e., simulated annealing, linear, chaotic, sigmoid decreasing, and logarithm decreasing). Logarithm PSO trumps alternative cloud work scheduling methods, experiments show. The suggested PSO-based scheduler outperforms the average GSA, ABC, and DA algorithms. Various criteria must be considered, such as load balance and energy usage. In addition, the proposed technique will have to be used in a variety of contexts and application workflows [21].

The proposed electro search algorithm adopts a three-phase scheme using the Bohr model and Rydberg formulae. The ES algorithm's new features enable it to find optimal global points without initializing tuning parameters. It was found that the optimal results outperformed selected algorithms such as GA and SA in terms of computation time

and success rate. The proposed algorithm outperformed the others. This method required testing for task scheduling and considering other performance parameters in cloud [22].

Bansal and Malik introduced a PSO-based multifaceted scheduling framework (MFOSF). This work presented a resource cost timeline model (RCTM) to define task resource needs. An updated PBPSO-based model was proposed to maximize performance scheduling and user cost. Third, better PBPSO was proposed to prevent PSO from going local. P_{best} and G_{best} changed the solution's quality based on performance and budget. Enhanced PBPSO is superior to similar approaches in cost, violation rate, and resource utilization, confirming its effectiveness. The proposed algorithm must be tested using specified QoS and energy usage statistics [23].

According to Khan and Santhosh, task scheduling can reduce time of waiting and increase service quality in cloud. The support vector machine loads first categorize it. PSGWO is used in the hybrid technique to find the best virtual machines and resource allocation. Traditional ant colony and PSGWO are compared to the proposed scheduling paradigm. In every parameter, the proposed hybrid optimization-based work scheduling outperforms previous approaches. This work cannot explore better QoS with VM allocation [24].

Kumar and Sharma proposed a resource allocation model employing PSO-COAGENT scheduling to maximize execution cost, makespan time, throughput, task rejection ratio, and energy consumption based on fitness function while taking deadline considerations into account. The proposed PSO-COAGENT method outperformed the already employed PSO, honeybee, and min-min strategies in terms of execution cost, execution time, and energy consumption. Consideration of various QoS factors, SLA, and testing for real-time applications such as agricultural is required for this task [25].

For instance, to enhance task scheduling behavior, a hybrid electro search with a genetic algorithm (HESGA) was proposed by Velliangiri et al. The advantages of genetic and electro search algorithms were integrated. Globally, electro search outperforms genetic algorithm. The proposed HESGA algorithm is compared to existing approaches. HESGA approach obtains better results than HEPESGA, GA, ES, and ACO. This work required an enhanced version that took energy consumption, QoS metrics, and real-time applications such as agriculture [26].

Gokuldhev and Singaravel proposed the LPMSA algorithm for cloud job scheduling. Moth search and floral pollination algorithms were combined (FPA). The proposed LPMSA picks the best cloud job scheduling solution. The suggested LPMSA is evaluated on machines with low and high heterogeneity. The suggested LPMSA saves time and energy over existing methods. The Wilcoxon test compares makespan minima and energy usage. This work's limitations are the necessity to test with real-time applications and add more parameters to the algorithm [27]. The local pollination-based gray wolf optimizer (LPGWO) method was used by Gokuldhev et al. to efficiently schedule jobs. GWO and FPA are both used in the hybrid algorithm. In the presence of GWO, data are distributed via local pollination to the

subsequent potential solution packet. These methods are used to solve early tasks. Work scheduling in low and high machine heterogeneity was enhanced.

Finally, comparing simulation results revealed the slowest convergence of makespan and energy consumption. This work required testing with real-time applications and adding more parameters to the method are needed [28]. Compare the various scheduling algorithms in this work. The HEFT algorithm ranks tasks and assigns them to processors. Give the heterogeneous processor tasks to reduce makespan time. The proposed algorithm outperforms load balancing and task makespan time with HEFT and CPOP. However, the algorithm can be improved by considering various deadlines, QoS parameters, and application tests [29]. This research proposed a SACO method with slave ants for cloud computing task scheduling. To avoid long routes with pheromones erroneously accumulated by leading ants, we use slave ants to diversify and reinforce—no pre-processing overhead for slave ants and beats existing ACO-based cloud work scheduling algorithms. SACO solves the NP-hard issue efficiently while maximizing cloud server utilization. Heterogeneous clusters cannot be considered because computing instances decide costs [30]. An ICM-PACO approach is proposed in this work for solving complicated large-scale optimization issues. The algorithms ACO and IACO perform better than ICM-PACO when dealing with gate assignment and travelling salesman issues. The gate assignment problem and typical TSP cases are both successfully solved by ICM-PACO. A total of 132 flights can be efficiently routed to 20 gates (83.5%). ICM-PACO outperforms ACO and IACO in terms of optimization and stability. The ICM-PACO algorithm requires more research because it takes longer to solve difficult optimization issues [31]. Q-ACOA is suggested for work scheduling and resource allocation in cloud computing based on current problems. For the first time, critical performance for task scheduling is established. Utilize Q-ACOA to efficiently execute processes, move data, and delight consumers. The work scheduling and resource allocation in cloud computing are optimized by Q-ACOA. Despite these successes, there are still problems with research. For some reason, there is no correlation between the tasks. To aid in resource allocation and scheduling in cloud computing, the research's limitations should concentrate on task correlation [32]. This study presents a FACO algorithm for cloud computing virtual machine load balancing. Ragmani et al. proposed a FACO cloud virtual machine scheduling algorithm. Due to scalability, ant colony optimization was used. CloudSim optimizes ACO settings. FACO uses evaporation to avoid nonoptimal early convergence. The proposed approach can cut response time by 80%, processing time by 90%, and total cost by 9%. We want to define pheromones beyond FACO. FACO has not been tested in a multicloud scenario [33].

When we think of implementing a new framework, we may have few merits and demerits. For example, smart bluetooth is one of the emerging wireless technologies used for data transfer between short distances. It is also cheaper than other technologies having advantage of being available on almost every smart phone [34]. The practice of current

traditional centralized security measures may lead us with limitations because of single point of failure, traceability, verifiability, as well as scalability [35]. When we chose multiclass model, development should be done with the consideration of the relative status of the factors taken [36].

Abualigah and Diabat offer a hybrid antlion optimization method with elite-based differential evolution to solve multi-objective task scheduling challenges in cloud computing. The MALO solution must maximize resource efficiency while minimizing makespan time [37]. Two experimental series on artificial and real trace data sets were run with CloudSim. MALO outperformed other well-known optimization methods.

This paper suggests a more efficient task scheduling method and an approach to optimal power minimization to help with dynamic resource allocation. Using a prediction mechanism and a dynamic resource table update approach can increase the effectiveness of resource allocation in terms of job completion and reaction time [38]. This architecture is successful in lowering total power consumption because it decreases data center power consumption. The proposed approach can be used to update the resource table. In order to achieve an effective resource deployment, improved job scheduling and a mechanism that uses less power are implemented. The simulation produces results that are 8% more accurate when compared to other approaches. To solve those problems, a hybrid machine learning (RATS-HM) technique is created. Finally, by simulating the suggested RATSHM technique with a new simulation setup and comparing the outcomes with those of other existing techniques, its utility is shown [39]. With regard to resource usage, energy consumption, response time, and other factors, the proposed method performs better than the existing one.

The task scheduling issue for tasks in the sensor cloud computing architecture has been addressed in the literature using a variety of metaheuristic- and heuristic-based algorithms, including PSO, GA, ES, ACO, and CRO. The majority of these algorithms are not effective enough for scheduling jobs in a multicloud setting, according to a review of pertinent studies. The algorithm used in related works lacks both global and local optimum solutions. The parameters necessary to improve task scheduling performance are not taken into account by many algorithms. Based on these conclusions, we developed the hybrid electro search-ant colony optimization technique (HES-ACO) to enhance task scheduling behavior by optimizing parameters such as makespan time, execution cost, total execution time, energy consumption, throughput, response time, resource utilization, and deadline constraints of the sensor cloud. Table 1 describes about the literature review of metaheuristic hybrid task scheduling algorithms with limitations

3. System Model and Problem Statement Formation of Problem Statement

Cloud computing has seized control of the computing market in the recent decade, offering users a wide range of services. The popularity of cloud computing is causing a significant increase in cloud users. As the number of users grows, the system encounters several challenges. Mapping

TABLE 1: literature review of metaheuristic hybrid task scheduling algorithms with limitations.

S. no.	Author and year	Methodology	Parameters	Limitations	Tool
1	Sreenivasulu and Paramasivam (2021) [13]	Hybrid optimization algorithm (BAT and BAR), hierarchy process model, and MOML preemption policy	Turnaround time, response time, memory utilization, bandwidth utilization, and resource utilization	(i) The author does not consider energy consumption and is required to prove the efficiency of the proposed algorithm with real-time workflows	CloudSim
2	Dubey and Sharma 2021) [14]	Hybrid AC-PSO algorithm and task scheduling	Makespan rate, cost, and resource utilization rate	(i) The author does not define the fitness function for energy (ii) The work does not consider the parameters of energy consumption, throughput, and schedule length (iii) Need to improve the time complexity	CloudSim
3	Dubey and Sharma (2021) [15]	Hybrid CR-PSO algorithm	Makespan rate, cost, execution time, and energy consumption	(i) Author required plan scheduling for dependent tasks and needed to verify the effectiveness of proposed work on parameters such as energy consumption, load balancing, task rejection ratio, and turnaround time	CloudSim
4	Prem Jacob and Pradeep (2019) [20]	CPSO	Deadline, makespan time, and cost	(i) Required to consider various other QoS service parameters and energy consumption	CloudSim
5	Khan and Santhosh (2021) [24]	PSGWO	Makespan time and execution time Waiting time, energy efficiency, and resource utilization	(i) Required to apply this technique for various applications	NetBeans
6	Kumar and Sharma (2018) [25]	PSO-COGENT	Execution time, execution cost, makespan time, energy consumption, throughput, and task rejection ratio	(i) Required to consider various SLA and QoS parameters for verifying the algorithm's effectiveness (ii) need to test for various workflows in the cloud	CloudSim
7	Velliangiri et al. (2021) [26]	HESGA	Makespan time, cost, and response time	(i) Required to apply this approach with other applications such as agriculture and so on (ii) Need to consider various parameters such as energy consumption, load balancing, QoS, and so on	CloudSim
8	Gokuldhev and Singaravel (2020) [27]	LPMSA	Makespan time and energy consumption	(i) The proposed technique is required to test with real-time application and needs to consider various parameters and is still required to enhance the algorithm	CloudSim with Java
9	Gokuldhev and Singaravel (2020) [28]	LPGWO	Makespan time and energy consumption	(i) This work needs scheduling in low and high machine heterogeneity was enhanced and consider various QoS metrics is required	CloudSim with Java
10	Dubey et al. (2018) [29]	HEFT	Makespan time and load balancing	(i) The proposed method is required to consider various QoS parameters and need test its effectiveness	CloudSim

TABLE 1: Continued.

S. no.	Author and year	Methodology	Parameters	Limitations	Tool
11	Ragmani et al. (2020) [33]	FACO (ACO and fuzzy logic)	Total processing time, response time, cost, and load balancing index	The proposed approach needs to be evaluated within a real and multicloud computing architecture and required considering various parameters such as energy consumption	CloudSim
12	Bhasker and Murali (2022)	The proposed method (HES-ACO)	Total execution time, execution cost, makespan time, energy consumption, throughput, task rejection ratio, resource utilization, and deadline constraint	The proposed approach extends this work further to consider security issues while users access the cloud's information	CloudSim

desired tasks to virtual machines and determining the best schedule sequence is a complex problem in the cloud. The most delicate virtual machine in the cloud must be used to process the user's task request. Under deadline limitations, this efficient strategy can reduce energy consumption, costs, resource utilization, execution time, makespan, throughput, and response time parameters. This study attempts to provide an efficient solution for processing applications depending on user demand and priority (time, energy, cost, and deadline) while concurrently improving the QoS level.

Consider the k number of tasks and the p number of computational resources that can be used to handle the task requests in the cloud data center's virtual machines. Based on their demands, service providers choose the best resources for end customers. The following are the definitions for the task set, resource set, and virtual machine set:

$$T = \{T_1, T_2, T_3, \dots, T_K\}, \quad (1)$$

$$R = \{R_1, R_2, R_3, R_4, \dots, R_m\}, \quad (2)$$

$$V_m = \{V_{m1}, V_{m2}, V_{m3}, V_{m4}, \dots, V_{mq}\}. \quad (3)$$

Every task T_a is defined as follows:

$$T_a = \{T_{id}, TL_i, D_i\}, \quad (4)$$

where the task identification number is denoted as T_{id} , MIPS length of the task is denoted as TL_i , and D_i is represented as deadline constraints associated with each task.

Similarly, each Vm_q is also characterized as follows:

$$Vm_q = \{Vm_{type}, Vm_{MIPS}, Vm_{speed}, Vm_{storage}, Vm_{id}\}, \quad (5)$$

where $MIPS$ signifies the virtual machine computational power. The cloud type is denoted by Vm_{type} to which the Vm belongs and is expressed in integer ranges. Virtual machine identification number is represented as Vm_{id} . Vm_{speed} is the virtual machine's processing speed, while $Vm_{storage}$ is each virtual machine's storage capacity in the cloud data center [14].

When an application is scheduled at resources (R_m), it has the option of getting the resource right away or waiting until the current application at R_m is defined in the following equation:

$$FTE = \sum_{m=0}^l R_m + TEx_{T_k Vm_q}. \quad (6)$$

FTE of task T_i at resource R_m should be less than the deadline of the task request ($\delta(Di)$).

$$FTE \leq \delta(Di). \quad (7)$$

3.1. Objective Function. The fundamental objective of the suggested approach is to enhance QoS metrics such as energy consumption, makespan time, computation cost, execution time, resource utilization, throughput, task rejection ratio, and response time. Users of the cloud also need services that are as inexpensive as possible. As a result, we create a fitness function with the deadline taken into account as a QoS parameter, whose objective is to minimize time, execution cost, and energy usage. The following functions are described by the authors.

3.1.1. Execution Time. A task's execution time is the length of time it takes the system to finish it.

$$TEx_{T_k Vm_q} = EEx_{T_k Vm_j} + TTime_{T_k Vm_j},$$

$$EEx_{T_k Vm_j} = \frac{TLen_{T_k}}{Vm_{MIPS}}, \quad (8)$$

$$TTime_{T_k Vm_j} = \frac{TLen_{T_k}}{Bw_{Vm_j}},$$

where $TEx_{T_k Vm_q}$ is the total time for processing the T_k task on the Vm_q virtual machines, and it is the total expected time $EEx_{T_k Vm_q}$ of the task T_k on the virtual machine Vm_q and the task transfer time $TTime_{T_k Vm_q}$.

3.1.2. Makespan Time. When all tasks have been processed or the entire amount of time has passed between the start and conclusion of the tasks, this is referred to as the "makespan time" of a task schedule.

$$Mspan_{time} = \max \left\{ \sum_{j=1}^q EEx_{T_k Vm_j} \right\}. \quad (9)$$

3.1.3. Execution Cost. The second objective functions goal is to reduce the overall execution cost represented as $TExe_{cost}$. While considering the deadline constraints, the Task represented as T_k on a per-hour basis for virtual machines Vm_j and cloud resources [25]. The execution cost is defined as follows:

$$TExe_{cost} = ExeC_{T_k Vm_j} * Cost_{Vm_j}, \quad (10)$$

where $ExeC_{T_k Vm_j}$ is the cost for executing the T_k task on the virtual machine Vm_j and $Cost_{Vm_j}$ is the resource cost of Vm_j in the cloud.

3.1.4. Energy Consumption (EC). Both dynamic and static energy were consumed by each physical machine. We only evaluate active energy consumption in this study since we believe that static energy consumption has a minor impact and can be ignored. The proposed algorithm's third scheduling goal is to reduce carbon emissions by maximizing resource use. The number of Vm instances available determines the quantity of dynamic energy consumed. Equation below shows how much energy a virtual machine uses [15].

$$Vm_j = \begin{cases} \varphi_i & \text{if } Vm_i \text{ is in active state} \\ \omega_j & \text{if } Vm_j \text{ is in idle state} \end{cases}. \quad (11)$$

The following equation computes the energy consumption of all the VMs (active and idle):

$$Ec_{Vm_j} = FTT_j * \varphi + (Mspan_{time} - FTT_j) * \omega, \quad (12)$$

where $Ec_{actualconsumption}$ is actual energy consumption is computed by using the following equation:

$$Min Ec_{actualconsumption} = (Ec_{max} - Ec_{min}) * Resu_t + Ec_{min}, \quad (13)$$

where Ec_{max} indicates resource usage R_m , while Ec_{min} denotes the ideal or minimum resource usage condition.

The total energy consumption of the data center is defined as follows:

$$Ec = \sum_{j=1}^q Ec_{Vm_j} + Min Ec_{actualconsumption}. \quad (14)$$

3.1.5. Throughput. The throughput (ϑ) is evaluated by using the following equation:

$$\text{Throughput } (\vartheta) = \frac{\text{Total number of successfully completed tasks}}{\text{total processing time}}. \quad (15)$$

3.1.6. Task Rejection Ratio (TR_i). If the task is not completed within the deadline constraint, it is computed using the following equation:

$$\text{Task Rejection Ratio } (TR_i) = \frac{\text{Number of tasks rejected}}{\text{Total number of tasks}} * 100. \quad (16)$$

3.1.7. Deadline Constraint. If the total time exceeds the deadline, it is defined as follows:

$$\delta(Di) = \frac{(\text{number of tasks violating the deadline constraint} * 100)}{\text{total number of tasks}}. \quad (17)$$

3.1.8. Fitness Function ft (R_m). All cloudlets should be handled prior to the deadline in order to meet our objective of reducing the amount of energy required by the specified operation or cloudlet (makespan, cost, time, throughput, and task rejection percentage). The fitness function of the problem of work scheduling with various objectives is defined by the following equation:

$$ft(R_m) = \alpha * Mspan_{time} + \beta * TExe_{cost} + \gamma * Ec. \quad (18)$$

Subject to

$$\sum_{j=1}^q \delta_{ij} = 1, \forall i \in T_i \forall j \in Vm_j. \quad (19)$$

Equation (17) shows that each application has only one resource assigned to it.

$$ft_{i,j} = ft_{i-1,j} \forall i \in T_i \forall j \in Vm_j. \quad (20)$$

Some assumptions and constraints are needed to consider for the tasks submitted in the cloud. α , β , and γ are the weight metrics of makespan $Mspan_{time}$, cost $TExe_{cost}$, and Energy consumption Ec , respectively [15].

4. Hybrid ES-ACO Task Scheduling Algorithm

The dynamic nature of task scheduling makes it difficult to identify the best resource. By taking into account a number of variables, we examine the issue of energy consumption and makespan, which must be reduced, and system performance, which must be optimized, because it directly affects the revenue and scalability of sensor cloud resource suppliers. This section covers the hybrid ES-ACO strategies for finding the ideal task scheduling solution in the sensor cloud environment as well as the sensor cloud model for the task scheduling algorithm. The recommended approach produced a hybrid ES-ACO task scheduling framework by fusing the advantages of ESO and ACO. The suggested hybrid ES-ACO architecture is depicted in Figure 2. The suggested architecture includes several WSNs, and it makes obvious how work scheduling was carried out in the cloud environment.

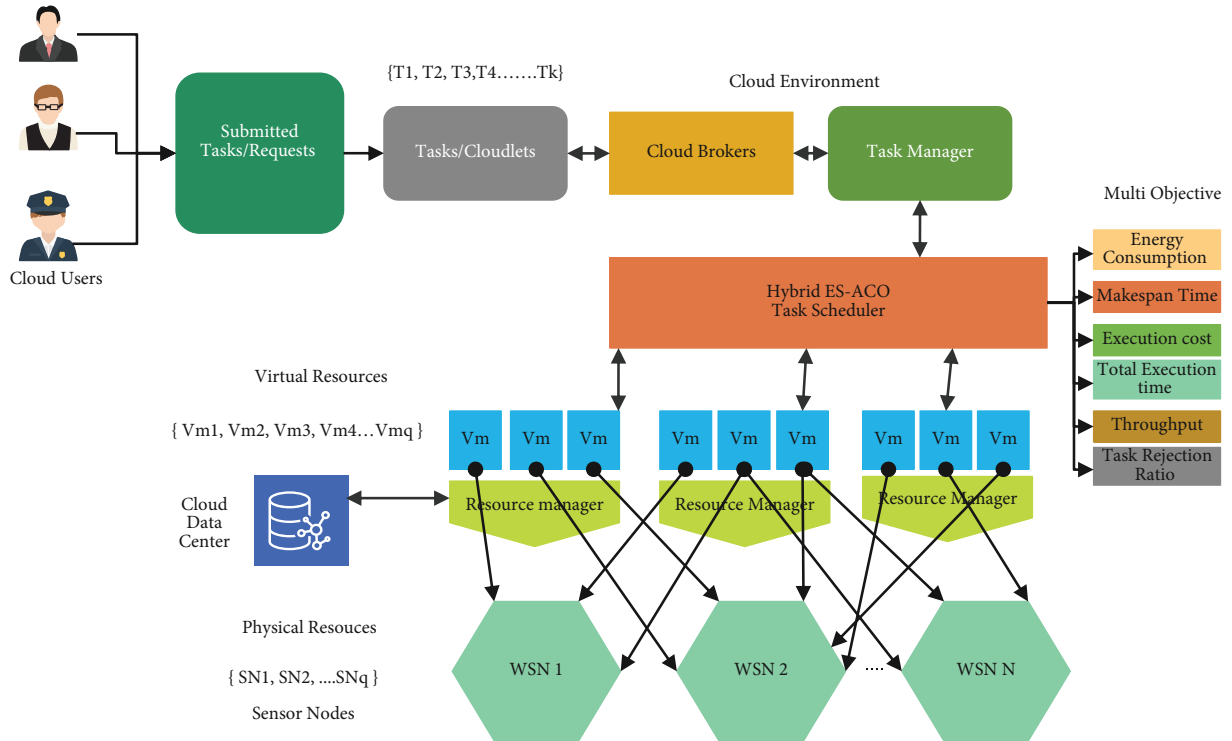


FIGURE 2: Proposed infrastructure model for task scheduling in sensor cloud environment.

4.1. Traditional Electro Search Optimization Algorithm (ESO). The electro search algorithm's domain of potential solutions is comparable to the molecular space in which various atoms are arranged. The electrons of every atom are arranged around its nucleus. To reach the highest energy level of molecular states, the orbits of the electrons that surround the nucleus of each atom gradually alter. It is equivalent to the maximum of the objective function [22].

4.1.1. Overview of the Standard Electro Search Procedure. The electro search algorithm can be divided into the following three phases:

- (1) Atom spreading
- (2) Orbital transition
- (3) Nucleus relocation

4.1.2. Atom Spreading Phase. The possible solutions are distributed at random in this step. Each potential candidate is an atom. They have a nucleus, which the electrons orbit around. The electrons are limited to precise orbits around the nucleus, and when they move between them, a certain amount of energy is either absorbed or released.

4.1.3. Orbital Transition Phase. The electrons around each nucleus expand their orbits during this phase in an effort to reach orbits with greater energies. The idea of quantized energy levels in a hydrogen atom served as inspiration for this orbital transition.

4.1.4. Nucleus Relocation Phase. The energy of a photon that is emitted during this phase, which is determined by the energy level difference between the two atoms, is used to determine the position of the new nucleus.

4.1.5. Traditional Ant Colony Optimization (ACO). Traditional ACO is a metaheuristic technique created by Italian researchers based on the foraging behavior of ant colonies [14]. When ants seek for food away from their nests and colonies, they leave a trail of pheromones in their wake. The likelihood of discovering the quickest route from the food source to the ant colonies was influenced by the density of pheromones. Once the food source is located, each ant travels in that direction using the quickest route and highest pheromone concentration. The shortest path is discovered using the ACO techniques in the subsequent steps.

- Step 1: Set the number of ant colonies and iterations
- Step 2: Set the beginning point at random
- Step 3: Each node chooses a direction based on pheromone concentrations
- Step 4: Then add the traverse path to the list
- Step 5: Update the pheromones after each iteration
- Step 6: Reiterate till the halting criteria are not reached

4.2. Hybrid ES-ACO Algorithm. Both evolutionary computational techniques and metaheuristic approaches are based on nucleus instincts and quickly produce an ideal

solution. ACO and ES are two terms that are used interchangeably.

The HES-PSO work scheduling method combines electro search with ant colony optimization. The hybrid ES-ACO strategy reduces makespan time, minimizes energy consumption, reduces computing cost, increases resource utilization, reduces execution time, reduces task rejection ratio, reduces throughput, and improves reaction time. Both approaches were used to quickly find an optimal solution with reduced time complexity.

To develop a metaheuristic method, the electro search (ES) algorithm with ant colony optimization (ACO) is employed. The three stages of the optimization of the electro-search method are spreading atom phase, orbital transition phase, and nucleus relocation phase.

This modification of the conventional electro search (ESO) is a first by incorporating the ACO to raise the standard of the ideal answer [26]. The recommended approach is broken down into the following four steps:

- (1) Initialization
- (2) Spreading of atoms
- (3) Transition of orbital
- (4) Relocation of nucleus

Tasks ($T_1, T_2, T_3, T_4, \dots, T_k$) and resources ($Vm_1, Vm_2, Vm_3, Vm_4, \dots, Vm_q$) are dispersed throughout the cloud datacenter in the first phase. The second phase designates the atoms as the tasks, that is, the cloud permits the release of the nucleus agent, which stores the locations and details of the virtual machines.

The state diagrams of atom spreading, orbital alteration, and nucleus relocation are shown in Figures 3–5. Similar to atoms, these agents are dispersed throughout the search space. This stage involves randomly dispersing the competitor configurations throughout the research area. Each user interacts with an iota, which is made up of an orbiting core and electrons. According to Velliangiri et al. [26], the electrons are confined to particular rings that encircle the core and can only move between them while emitting or maintaining particular levels of liveliness. The atoms scour the search space in quest of the best answer. The current and previous resource information will be stored in the third space, and each agent will follow atom around the resource pool, or virtual machine, to choose the best one. Following the best virtual machines, the atoms choose the optimal solution in the last step using the data stored in the agent. The jobs are consequently delegated to virtual machines.

This study introduces the ES-ACO algorithm, a multi-objective task scheduling system that combines the benefits of both traditional electro search and the ant colony optimization method. Schedule the work to a virtual machine in an inefficient manner, which reduces makespan time, minimizes energy consumption, decreases computation cost, increases resource use, decreases execution time, decreases task rejection ratio, increases throughput, and improves reaction time. When comparing HES-ACO methods to ES methods, PSO methods, GA methods, ACO methods, HESGA methods, and AC-PSO methods, the HES-ACO

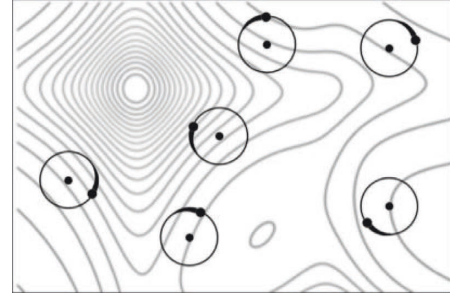


FIGURE 3: State diagram of atom spreading.

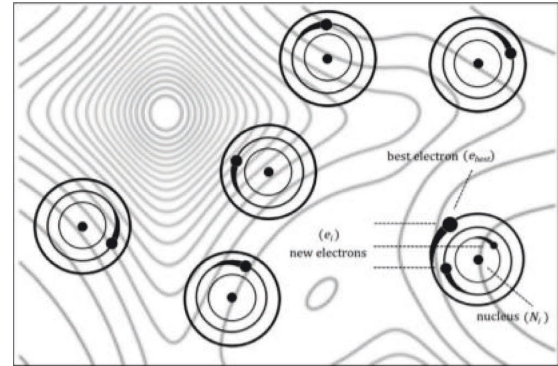


FIGURE 4: State diagram of orbital transition.

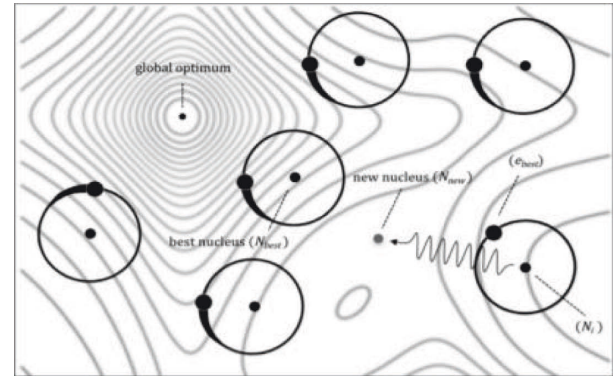


FIGURE 5: State diagram of nucleus relocation.

methods record a significantly higher space utilization. Algorithm 1 describes about hybrid ES-ACO task scheduling approach.

Figure 6 shows the flow diagram of proposed hybrid ES-ACO technique for task scheduling in sensor cloud environment.

4.2.1. Selection of Parameters

(1) *Number of Particles (h)*. The number of particles in the underlying population, such as the needed calculation

Input 1: Set of subtasks, that is, $T_1, T_2, T_3, T_4, \dots, T_k$. 2. Set of virtual machines, that is, $Vm_1, Vm_2, Vm_3, Vm_4, \dots, Vm_q$
Output: Mapping of the tasks to set the Vms (optimal schedule)
Step 1: Initialize the set of ant colonies
Step 2: Set the parameters of ACO
Step 3: Initialize the set of subtasks, that is, $T_1, T_2, T_3, T_4, \dots, T_k$
Step 4: Initialize the set of virtual machines, that is, $Vm_1, Vm_2, Vm_3, Vm_4, \dots, Vm_q$
Step 5: Compute pheromone value $\rho_{i,j}(0) = (c/TEx_{T_k Vm_j})$
Step 6: Submit the Vm list, which was created successfully in the data center, and set of tasks to the cloud broker
Step 7: For 1 to Q do()
Step 8: Generate nucleus $Q[i]$
Step 9: Initialize nucleus agent randomly
Step 10: End for
Step 12: $Rm = 0$
Step 13: Define the fitness function $ft(Rm)$
Step 14: $ft(Rm) = \alpha * Ms\text{pan}_{time} + \beta * TExe_{cost} + \gamma * Ec$
Step 15: Compute G_{best} and P_{best}
Step 16: While (Max iterations X) // $\forall i \in T_i \ \& \ \forall j \in Vm_j$
Step 17: Update the pheromone, that is, monitor the status of resources using the following equation
Step 18: $\bar{\partial}_{i,j}(t+1) = (1 - \theta) * \bar{\partial}_{i,j}(t) + \sum_{b=1}^{\sigma} \bar{\partial}_{i,j}^{\sigma}(t)$
Step 19: $\bar{\partial}_{i,j}^{\sigma}(t) = \begin{cases} (\epsilon/L_n)\sigma\text{th passes on the each nucleus} \\ 0, \text{ else the growth of aunt } 0 \end{cases}$
Step 20: $i = 0$
Step 21: Compute the fitness value for each nucleus of $Q[i]$
Step 22: $G_{best} = \text{best nucleus of } Q[i]$
Step 23: $i++$
Step 24: for $a = 1$ to Q
Step 25: $P_{best}[a] = Q[i]$
Step 26: End for
Step 27: End while
Step 28: Return the global best solution of atom

ALGORITHM 1: Hybrid ES-ACO task scheduling algorithm.

emphasis, work assessment, accomplishment rate, and so on, typically raises the thoroughness of the chase space inspection in an arbitrary hunt approach. Meanwhile, as we will see in the following section, increasing the number of molecules decreases the number of calculation cycles needed to fulfill the goal, hence boosting the achievement rate. Regardless of how significant the aforementioned criterion may be, the quantity of capacity assessments is viewed as the fundamental execution regulation for applications. Fewer molecules in a population result in a lower success rate and more cycles. A huge population, on the other hand, increases the achievement rate and requires an excessive number of capacity evaluations.

(2) *The span of Orbital (a)*. Electrons in each core might be able to move to larger rings. The orbital distance is the size of the greatest possible circle. Because of the electrons that orbit each nucleus, it is now feasible to shift the emphasis to any location. The migration removes (and consequently the orbital range of every core) change during the course of the cycles as the particle travels closer to the ideal place. As the orbital sweep is dynamically compressed, the transition region for electrons shrinks. This decrease permits the computation to merge toward the target by constraining

electron movement in the vicinity of the hypothetical global ideal point.

(3) *Number of Electrons (e)*. Electro Search's electrons around each core represent the randomness of scientific exploration into the cosmos. An increase in the merging rate is seen when arbitrary electrons in each focus are far off from one another. This method avoids hastily joining imperfect focal points together. Even when the underlying molecules are less than ideal, this method makes space exploration more efficient. It improves the inquiry space investigation, especially when the underlying molecules are far from ideal. Furthermore, the iterative procedure's modest compression of orbital sweep bounds the territory is taken into account orbital movement, preventing additional arbitrary electrons from being placed far from the core

(4) *Convergence Criteria*. Population mixing owing to emphasis is taken into account using stopping criteria in transformative calculations. When the total number of people living in the applicant setups reaches a "stale" number, the analysis is done. Different end criteria were utilized in the development writing, such as reaching the maximum number of cycles, arriving at a desirable

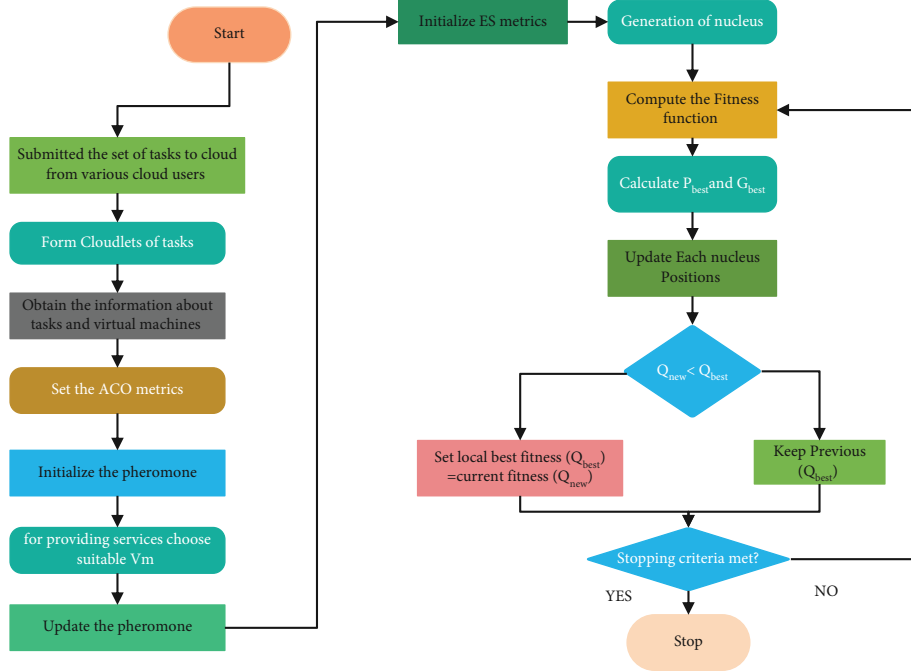


FIGURE 6: The flow chart of the proposed hybrid ES-ACO.

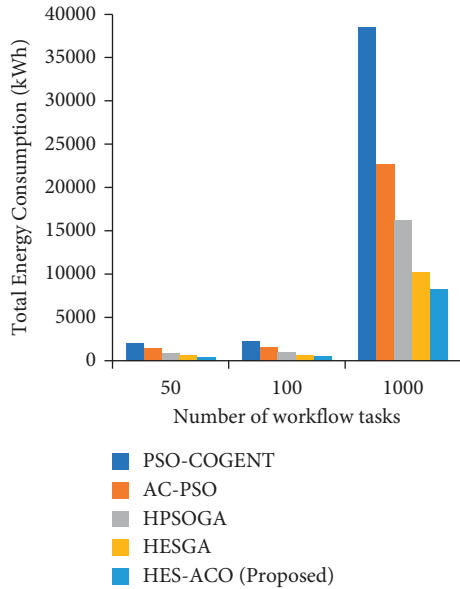


FIGURE 7: Energy consumption.

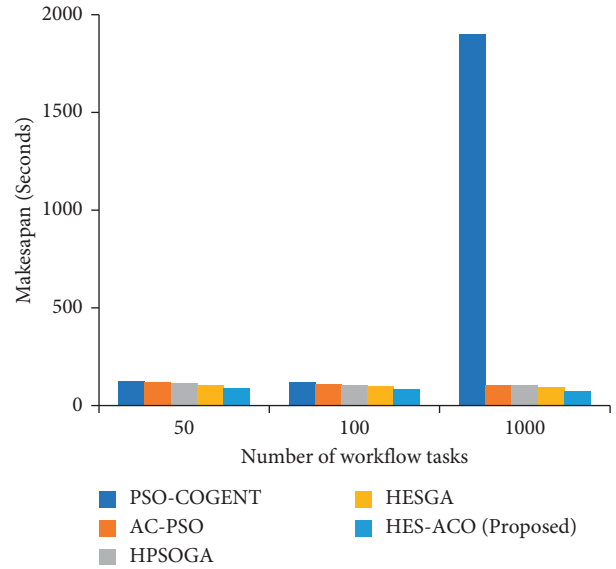


FIGURE 8: Makespan time.

arrangement, and so on. For the purpose of the electro search calculation, the highest number of cycles served as the ending rule [26].

5. Experimental Result and Discussion

The developed hybrid electro search with ant colony optimization (HES-ACO) task scheduling algorithm was compared to the HESGA, HPSOGA, AC-PSO, PSO-COAGENT, and the proposed algorithm in a comparative analysis (HES-ACO). Energy consumption, execution cost, total execution

time, throughput, makespan time, task rejection ratio, response time, and resource usage are all used to evaluate the proposed model HES-ACO performance.

5.1. Simulation and Results Analysis. Task scheduling experiments are described here so that their results can be analyzed. To ensure that the suggested scheduling paradigm works well in a cloud setting, we use the CloudSim simulator to replicate the environment. When it comes to simulating the infrastructure as a service (IaaS) cloud, the CloudSim simulator is a useful framework. In order to carry out the

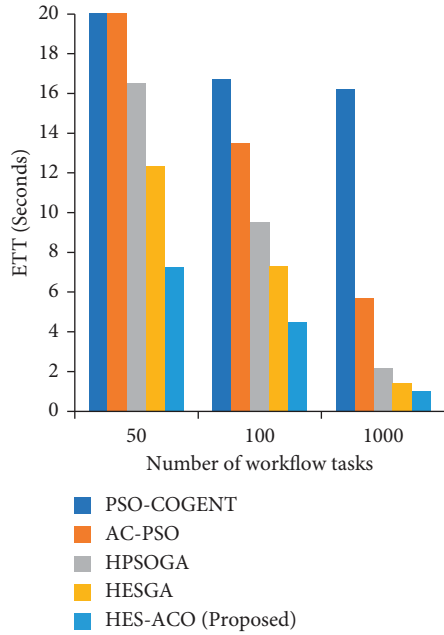


FIGURE 9: Execution time.

scientific workflow sustainably, new algorithms are implemented (including task scheduling, VM deployment, energy model, etc.). The efficacy of the HES-ACO algorithm is measured in an empirical fashion.

5.1.1. Simulation Parameters

- (i) Two distinct hosts are used, the HP ProLiant ML110 G5 and the HP ProLiant ML110 G4 [37], which use 135 W/s and 117 W/s of power, respectively.
- (ii) 2.3 W Energy consumption rate is consider to transfer 1 GB data.
- (iii) Four VMs with different CPU (in MIPS) and RAM (in MB) capacities are installed. On an average VM start-up time is 96.9 s. Around 2,500 MIPS with 870 MB RAM, 52,000 MIPS with 1740 MB, 1,000 MIPS with 1740 MB RAM, and 500 MIPS with 613 MB RAM run the scientific procedure. Based on the workflow requirements, the VMs are deployed/undeployed dynamically.
- (iv) Amazon Web Services offers 20 MBPS as average VM bandwidth.

5.2. Performance Metric and Simulation Parameters

Energy consumption: Total energy consumed by the servers to execute scientific workflows is computed using equation (13).

Makespan or total execution time: It is the total time to execute the workflow from entry tasks to the exit task. Equation (11) is used to calculate the makespan.

Execution time (ETT): Average execution time per task is calculated by equations (8)–(10).

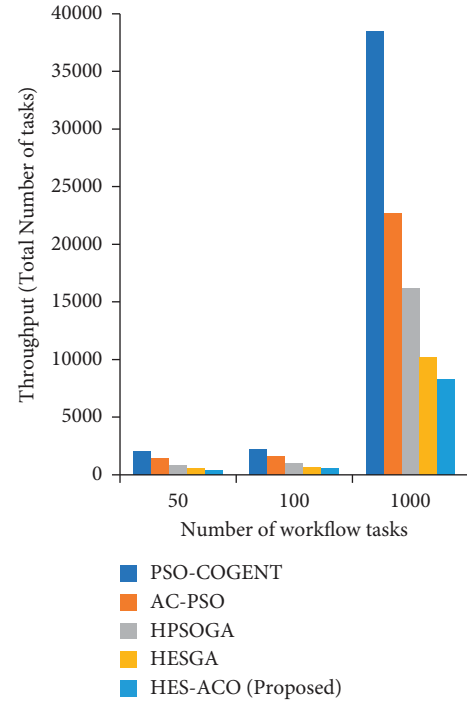


FIGURE 10: Throughput.

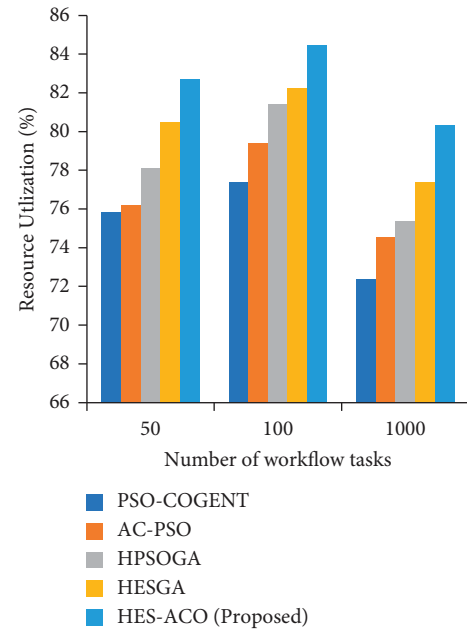


FIGURE 11: Resource utilization.

Throughput: It is evaluated by division of number tasks successfully executed and total number of tasks. It is calculated by using equation (17).

Execution cost: It is calculated by using equation (12).

pAverage RU: It is the ratio of allocated computing resources (such as CPU in MIPS) to execute the scientific workflow tasks and total computing resources of the server.

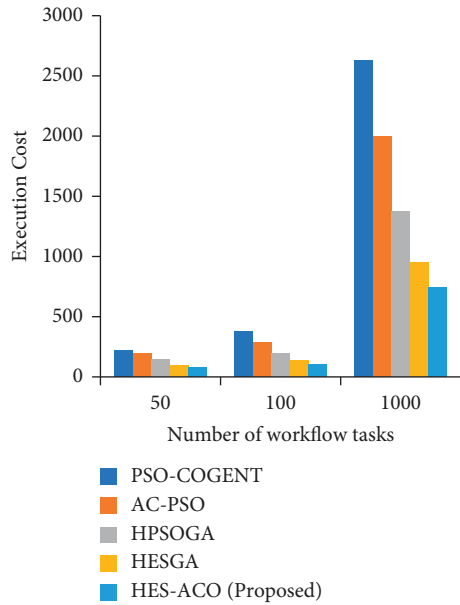


FIGURE 12: Execution cost for tasks.

Figure 7 clearly shows that proposed mechanism HES-ACO method outperformed existing methods such as AC-PSO, HPSOGA, and HESGA for energy consumption versus number of tasks workflow.

Figure 8 clearly show that proposed mechanism HES-ACO method outperformed existing methods such as AC-PSO, HPSOGA, and HESGA for makespan time versus number of tasks workflow.

Figure 9 clearly shows that proposed mechanism HES-ACO method outperformed existing methods such as AC-PSO, HPSOGA, and HESGA for execution time versus number of tasks workflow.

Figure 10 clearly shows that proposed mechanism HES-ACO method outperformed existing methods such as AC-PSO, HPSOGA, and HESGA for throughput versus number of tasks workflow.

Figure 11 clearly shows that proposed mechanism HES-ACO method outperformed existing methods such as AC-PSO, HPSOGA, and HESGA for resource utilization versus number of tasks workflow.

Figure 12 clearly shows that proposed mechanism HES-ACO method outperformed than existing methods such as AC-PSO, HPSOGA, and HESGA for execution cost versus number of tasks workflow.

6. Conclusion

There are not many task scheduling techniques for sensor clouds; hence, in this work, we mainly focus on user and cloud interaction. This study took into account a number of factors at once, including execution time, execution cost, throughput, energy use, makespan time, resource utilization, and deadline constraint parameters. In this essay, we have spoken about how customers' high processing needs are causing a daily increase in the number of cloud servers. Nevertheless, these servers use a lot of electricity. Both

sensor and cloud settings have significant issues with energy consumption. As a result, energy-efficient job scheduling is crucial for reducing energy use and improving the other variables. A hybrid electro search with ant colony optimization (HES-ACO) strategy is suggested in this research. Tasks are inefficiently scheduled using the proposed HES-ACO approach at virtual machine resources (V_m). It utilizes a fitness function to optimize the parameters (execution time, execution cost, throughput, reaction time, and energy consumption) while taking the task deadline into account as a quality-of-service parameter. The proposed method efficiently increases resource usage while minimizing energy consumption, cost, make-span time, execution time, throughput, and task rejection ratio when compared to HESGA, HPSOGA, AC-PSO, and PSO-COAGENT algorithms. The created algorithm can be applied in the future to various SLA, QoS, and security criteria that are also taken into consideration for further research.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] L. Hamami and B. Nassereddine, "Application of wireless sensor networks in the field of irrigation: a review," *Computers and Electronics in Agriculture*, vol. 179, p. 105782, 2020.
- [2] A. Bhoi, R. P. Nayak, S. K. Bhoi et al., "IoT-IIRS: internet of Things based intelligent-irrigation recommendation system using machine learning approach for efficient water usage," *PeerJ Computer Science*, vol. 7, pp. 5788–e615, 2021.
- [3] M. Zhang and S. Xiong, "Sensor-cloud based precision sprinkler irrigation management system," in *CWSN 2019*, C. S. Guo, Ed., Springer Nature Singapore Pte Ltd, 2019.
- [4] T. Ojha, S. Misra, and N. S. Raghuvanshi, "Sensing-cloud: leveraging the benefits for agricultural applications," *Computers and Electronics in Agriculture*, vol. 135, pp. 96–107, 2017.
- [5] R. Dalvi and S. K. Madria, "Energy-efficient scheduling of fine-granularity tasks in a sensor cloud," *Lecture Notes in Computer Science*, vol. 9050, pp. 498–513, 2015.
- [6] M. T. Rahman, T. F. Shuva, and R. T. Khan, "Efficient sensor-cloud communication using data classification and compression," *International Journal of Information Technology and Computer Science*, vol. 9, no. 6, pp. 9–17, 2017.
- [7] D. H. Phan, J. Suzuki, S. Omura, and K. Oba, "Toward sensor-cloud integration as a service: optimizing three-tier communication in cloud-integrated sensor networks," *Proceedings of the th International Conference on Body Area Networks*, vol. 1, pp. 355–362, 2013.
- [8] K. Das, S. Das, R. K. Darji, and A. Mishra, *Review Article Survey of Energy-Efficient Techniques for the Cloud-Integrated Sensor Network*, Springer, America, 2018.
- [9] T. Ojha, S. Bera, S. Misra, and N. S. Raghuvanshi, "Dynamic duty scheduling for green sensor-cloud applications," in *Proceedings of the International Conference on Cloud*

- Computing Technology and Science*, pp. 841–846, CloudCom, 2015-February.
- [10] S. Sivakumar and A. Al-anbuky, “Dense clustered multi-channel wireless sensor cloud,” *Journal of Sensor and Actuator Networks*, vol. 4, no. 3, pp. 208–225, 2015.
 - [11] S. Chatterjee, S. Sarkar, and S. Misra, *Energy-Efficient Data Transmission in Sensor-Cloud*, pp. 1–6, IEEE, China, 2015.
 - [12] T. Dinh and Y. Kim, “An efficient sensor-cloud interactive model for on-demand latency requirement guarantee,” *IEEE International Conference on Communications*, vol. 6, pp. 1–6, 2017.
 - [13] G. Sreenivasulu and I. Paramasivam, “Hybrid optimization algorithm for task scheduling and virtual machine allocation in cloud computing,” *Evolutionary Intelligence*, vol. 14, no. 2, pp. 1015–1022, 2021.
 - [14] K. Dubey and S. C. Sharma, “A hybrid multi-faceted task scheduling algorithm for the cloud computing environment,” *International Journal of Systems Assurance Engineering and Management*, vol. 32, p. 56845, 2021.
 - [15] K. Dubey and S. C. Sharma, “A novel multi-objective CR-PSO task scheduling algorithm with deadline constraint in cloud computing,” *Sustainable Computing: Informatics and Systems*, vol. 32, p. 100605, 2021.
 - [16] T. S. Proshikshya Mukherjee, *Issues Of Some Task Scheduling Strategies On Sensor Cloud Environment (Smart Inte)*, Springer Nature, Singapore, 2020.
 - [17] R. K. Jena, “Multi-objective task scheduling in cloud environment using nested PSO framework,” *Procedia Computer Science*, vol. 57, pp. 1219–1227, 2015.
 - [18] S. Swagatika, A. K. Rath, and P. K. Pattnaik, “Markov chain model and PSO technique for dynamic heuristic resource scheduling for system-level optimization of cloud resources,” *ARN Journal of Engineering and Applied Sciences*, vol. 13, no. 3, pp. 1042–1052, 2018.
 - [19] S. C. Nayak, S. Parida, C. Tripathy, and P. K. Pattnaik, “An enhanced deadline constraint-based task scheduling mechanism for cloud environment,” *Journal of King Saud University - Computer and Information Sciences*, vol. 832, 2018.
 - [20] T. Prem Jacob and K. Pradeep, “A multi-objective optimal task scheduling in cloud environment using cuckoo particle swarm optimization,” *Wireless Personal Communications*, vol. 109, no. 1, pp. 315–331, 2019.
 - [21] X. Huang, C. Li, H. Chen, and D. An, “Task scheduling in cloud computing using particle swarm optimization with time-varying inertia weight strategies,” *Cluster Computing*, vol. 23, no. 2, pp. 1137–1147, 2020.
 - [22] A. Tabari and A. Ahmad, “A new optimization method: electro-Search algorithm,” *Computers & Chemical Engineering*, vol. 103, pp. 1–11, 2017.
 - [23] M. Bansal and S. K. Malik, “A multi-faceted optimization scheduling framework based on the particle swarm optimization algorithm in cloud computing,” *Sustainable Computing: Informatics and Systems*, vol. 28, p. 100429, 2020.
 - [24] M. S. A. Khan and R. Santhosh, “Task scheduling in cloud computing using hybrid optimization algorithm,” *Soft Computing*, vol. 5, p. 457, 2021.
 - [25] M. Kumar and S. C. Sharma, “PSO-COGENT: cost and energy efficient scheduling in cloud environment with deadline constraint,” *Sustainable Computing: Informatics and Systems*, vol. 19, no. January, pp. 147–164, 2018.
 - [26] S. Velliangiri, P. Karthikeyan, V. M. Arul Xavier, and D. Baswaraj, “Hybrid electro search with genetic algorithm for task scheduling in cloud computing,” *Ain Shams Engineering Journal*, vol. 12, no. 1, pp. 631–639, 2021.
 - [27] M. Gokuldhev and G. Singaravel, “Local pollination-based moth search algorithm for task-scheduling heterogeneous cloud environment,” *The Computer Journal*, vol. 65, no. 2, pp. 382–395, 2020.
 - [28] M. Gokuldhev, G. Singaravel, and N. R. Ram Mohan, “Multi-objective local pollination-based Gray Wolf optimizer for task scheduling heterogeneous cloud environment,” *Journal of Circuits, Systems, and Computers*, vol. 29, no. 07, pp. 2050100–2050124, 2020.
 - [29] K. Dubey, M. Kumar, and S. C. Sharma, “Modified HEFT algorithm for task scheduling in cloud environment,” *Procedia Computer Science*, vol. 125, pp. 725–732, 2018.
 - [30] Y. J. Moon, H. C. Yu, J. M. Gil, and J. B. Lim, “A slave ants-based ant colony optimization algorithm for task scheduling in cloud computing environments,” *Human-Centric Computing and Information Sciences*, vol. 7, no. 1, p. 28, 2017.
 - [31] W. Deng, J. Xu, and H. Zhao, “An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem,” *IEEE Access*, vol. 7, pp. 20281–20292, 2019.
 - [32] Y. Su, Z. Bai, and D. Xie, “The optimizing resource allocation and task scheduling based on cloud computing and Ant Colony Optimization Algorithm,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, p. 0123456789, 2021.
 - [33] A. Ragmani, A. Elomri, N. Abghour, K. Moussaid, and M. Rida, “FACO: a hybrid fuzzy ant colony optimization algorithm for virtual machine scheduling in high-performance cloud computing,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 10, pp. 3975–3987, 2020.
 - [34] P. K. Reddy Maddikunta, S. Hakak, M. Alazab et al., “Unmanned aerial vehicles in smart agriculture: applications, requirements, and challenges,” *IEEE Sensors Journal*, vol. 21, no. 16, pp. 17608–17619, 2021.
 - [35] R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, T. R. Gadekallu, and G. Srivastava, “Sp2f: a secured privacy-preserving framework for smart agricultural unmanned aerial vehicles,” *Computer Networks*, vol. 187, p. 107819, 2021.
 - [36] N. Deepa, M. Z. Khan, B. Prabadevi, D. R. Vincent Pm, P. K. R. Maddikunta, and T. R. Gadekallu, “Multiclass model for agriculture development using multivariate statistical method,” *IEEE Access*, vol. 8, pp. 183749–183758, 2020.
 - [37] L. Abualigah and A. Diabat, “A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments,” *Cluster Computing*, vol. 24, no. 1, pp. 205–223, 2021.
 - [38] J. Praveenchandar and A. Tamilarasi, “Retracted article: dynamic resource allocation with optimized task scheduling and improved power management in cloud computing,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 3, pp. 4147–4159, 2021.
 - [39] P. K. Bal, S. K. Mohapatra, T. K. Das, K. Srinivasan, and Y. C. Hu, “A joint resource allocation, security with efficient task scheduling in cloud computing using hybrid machine learning techniques,” *Sensors*, vol. 22, no. 3, p. 1242, 2022.