

## RESEARCH ARTICLE

# Enhancement of Health Care Services Based on Cloud Computing in IOT Environment Using Hybrid Swarm Intelligence

KARIM M. HASSAN<sup>1,2</sup>, A. ABDO<sup>2,3</sup>, AND AHMED YAKOUB<sup>2</sup><sup>1</sup>Higher Institute of Computers and Information Technology, El-Shorouk Academy, Cairo 11837, Egypt<sup>2</sup>Department of Information Systems, Faculty of Computers and Artificial Intelligence, Helwan University, Cairo 11795, Egypt<sup>3</sup>Faculty of Computer Studies, Arab Open University, Cairo 11837, Egypt

Corresponding author: KARIM M. HASSAN (k.hassan@sha.edu.eg)

**ABSTRACT** Healthcare services (HCS) based on cloud computing and the Internet of Things are a great opportunity for the development of medical information technology. Task scheduling in cloud computing is one of the most critical problems facing health care services, as it affects the time required to fulfill user requests and the cost and quality of service delivery. The proposed HCS model structure consists of major components such as user devices, user requests, cloud broker, IoT endpoints, and HCS cloud. This paper proposes a new method to improve task scheduling in healthcare services based on cloud computing in the IoT environment (cloud-IoT). Specifically, A hybrid optimization algorithm HPSOSSA is proposed that combines the best existing swarm intelligence algorithms and integrates the advantages of particle swarm optimization (PSO) and the Salp Swarm Algorithm (SSA). The proposed model was implemented using the Cloudsim simulation package run on Eclipse with specific parameters. The proposed hybrid algorithm was compared to the most popular optimization algorithms that were previously used, such as Ant Colony Optimization (ACO), PSO, SSA, and hybrid PSO-GA. The experimental results showed that HPSOSSA in all cases outperforms the other existing algorithms in terms of makespan, waiting time, and resource utilization.

**INDEX TERMS** Cloud computing, HCS, swarm intelligence, Internet of Things, task scheduling, makespan.

## I. INTRODUCTION

Nowadays, with the spread of epidemics and health crises, healthcare services (HCS) based on cloud computing and IoT are regarded as one of the most important medical fields in the world, where the best use of HCS saves a lot of people's lives [1]. Cloud computing enables HCS users to access computing resources like applications, software, and hardware, etc., via the internet, which will be operated upon the user's requests. Therefore, cloud computing associated with the Internet of Things environment has become very useful for all users, especially in the healthcare services sector to transfer medical services over the Internet [2]. Due to the continued evolution of infrastructure and technology, the rapid adoption and ease of use in all sectors, and the proliferation of the

Internet of Things concept, this will increase the demands of users for cloud computing, doubling the amount of data and requests made by users. Task scheduling becomes a more complex problem. It is a tiresome task to deliver resources in accordance with the user's request while fulfilling quality of service (QoS) standards for the end-user [3]. A cloud comprises datacenters, hosts, virtual machines (VMs), resources, etc. End users are provided with distributed, scalable, and elastic computing resources via high-speed computer networks such as the Internet. Resources in a cloud environment can include everything from processing power (CPUs), memory, networking, and storage to development platforms and software apps [4]. Through virtualization technology, a cloud computing paradigm linked to an IoT environment allows for efficient use of existing physical resources. Hence, several clients can share a common physical infrastructure of hardware and software to store and retrieve multiple medical

The associate editor coordinating the review of this manuscript and approving it for publication was Shaohua Wan.

resources for healthcare services (HCS) users. Task scheduling problem (TSP) is one of the most important problems facing healthcare services, where users of healthcare services in a cloud computing environment face a time delay in medical requests. Several factors result in the time delay of medical requests, including waiting time, the turnaround time for medical requests, waste of CPU use, and the waste of resource utilization. TSP is a nondeterministic polynomial-time (NP) hard problem that is in charge of efficiently distributing application tasks to computation resources [5]. Generally, several strategies have been proposed to address the NP-hard problem, but no single technique provides a polynomial-time solution. However, meta-heuristic techniques have gained popularity in obtaining optimal solutions to this problem [6]. A meta-heuristic is a population-based technique. This technique is used in algorithms such as particle swarm optimization (PSO), ant colony optimization (ACO), genetic algorithm (GA), salp swarm algorithm (SSA), and several other algorithms that apply this technique [7]. Particle swarm optimization is one of the most popular Swarm Intelligence (SI) meta-heuristic techniques to solve the task scheduling problem.

The main aim of this study is to find a suitable solution for solving task scheduling problem to help the HCS users reduce their waiting time, computation costs, makespan, and turnaround time of medical requests on cloud computing and IoT environment as well as minimizing the waste of CPU utilization of VMs and maximizing the utilization of medical resources.

The main contributions of this paper include the following:

- Proposing a hybrid Particle Swarm Optimization salp swarm algorithm (HPSOSSA) that solve task scheduling of healthcare services on cloud computing and IoT environment.
- Presenting an intelligent model of cloud computing for HCS.
- Evaluating the proposed algorithm by comparing it with the most popular swarm intelligence optimization algorithms like PSO, ACO, SSA, and hybrid PSO-GA to prove the efficiency of the proposed hybrid algorithm.

The rest of this paper is organized as follows: Section 2 reviews the related work. Section 3 represents a brief review of PSO, SSA algorithms and then explains the proposed algorithm in detail. While section 4 introduces the proposed model. Section 5 shows the simulation setup and the experimental results. Finally, section 6 concludes the paper and sketches out future work.

## II. RELATED WORKS

Many studies that apply and use algorithms such as PSO, ACO, PPSO, BCO, and GA have been done to improve the scheduling process and performance of VMs in a cloud-IoT environment. This section reviews the recent studies in the literature related to that subject.

Jain and Sharma [8] proposed a QoS Aware Binary Salp Swarm Algorithm called “QBSSA” in cloud computing for efficient task scheduling inspired by salp’s natural behavior while navigating and searching for food in the sea. “QBSSA” is modeled and evaluated by comparing it with the other most widely used meta-heuristic algorithms, such as Ant Colony Optimization (ACO) and Grey Wolf Optimization (GWO). The simulation results revealed that QBSSA surpasses the competition regarding makespan, resource utilization, throughput, and average waiting time.

Another study [9] has presented a new hybrid method for scheduling tasks in cloud computing called hybrid particle swarm algorithm and a genetic algorithm based on phagocytosis PSO\_PGA. The main idea was to analyze the process of cloud task scheduling by dividing each group of particle swarm and changing the particle’s location in the subpopulation using the “phagocytosis mechanism” and “crossover mutation” of GA. Then, merging the subpopulations to maintain particle diversity in the population and decrease the chance of the algorithm finding the local optimal solution. The authors compared the proposed algorithm with several other existing algorithms. The results showed that the introduced technique greatly reduces the overall completion time of cloud tasks and increases convergence accuracy.

Kruekaew and Kimpan [10] introduced heuristic task scheduling using Artificial Bee Colony, which combines the artificial bee colony’s Swarm Intelligence algorithm with a heuristic scheduling technique (HABC). The main objective was to enhance virtual machine scheduling for cloud computing in both homogenous and heterogeneous conditions, and to reduce the makespan and balance the loads. The proposed method was tested against other swarm intelligence algorithms such as Ant Colony Optimization (ACO) using a standard heuristic algorithm, Particle Swarm Optimization (PSO) using a standard heuristic algorithm, and improved PSO (IPSO) using a standard heuristic methodology. The experimental results showed that virtual machine scheduling with an artificial bee colony algorithm and the largest job first (HABC-LJF) outperformed ACO, PSO, and IPSO.

A hybrid algorithm that combines particle swarm optimization algorithm and simulated annealing algorithm (PSOSA) has been presented in [11] to improve task scheduling performance for resources, taking into account the given bandwidth for each virtual machine. This hybrid algorithm is implemented using the CloudSim platform package. The experimental results show that in terms of task execution time, response time, and performance efficiency, the suggested method beats the standard PSO, bat, and raven roosting optimism algorithms.

Strumberger et al. [4] introduced Cloudlet Scheduling implementations by hybridized and original monarch butterfly optimization algorithms (MBO and MBO-ABC) that belong to the type of swarm intelligence meta-heuristics. The authors used the environment of the CloudSim platform to implement and adapt both algorithms for tackling the cloudlet scheduling problem. The proposed approach has proven to

be efficient and capable of solving the cloudlet scheduling problem with positive implications for cloud management. The study in [12] introduced three distinct optimization algorithms: Cuckoo Search (CS), PSO, and ABC. The goal of the proposed work is to locate the best VMs for cloud users to help work on lowering task execution time, turnaround time, and waiting time for requests from online medical systems. The research was simulated with MATLAB and CloudSim, and the performance was evaluated. The simulation results show that the proposed solution improves the system's real-time performance by lowering the total execution time and increasing system efficiency.

Abdelaziz et al. in [13] used particle swarm optimization (PSO) and parallel particle swarm optimization (PPSO), which are two different meta-heuristics optimization techniques, to develop a smart HCS model in a cloud environment. The new PPSO is a dependent algorithm tested on the CloudSim and MATLAB platforms to confirm its efficacy in solving the task scheduling problem to assist healthcare organizations in smart urban.

El-Shafeiy and Abohany in [1] introduced a new model named swarm intelligence for the Internet of Medical Things (SIoMT) system in healthcare. The proposed framework observes and analyzes the periodic data cluster readings collected from the IoT in smart health centers to recognize tasks and behavior changes. Applying the adaptive Artificial Bee Colony algorithm (ABC) to discover useful data using multiple measurements as an early stage to shorten the time required.

Ramaswamy and Mukherjee [14] introduced a healthcare decision-making support system using swarm intelligence. The proposed system implements (FFS-PSODP), which is a new filter-based feature selection method for particle swarm optimization with digital pheromones that effectively handles both balanced and imbalanced medical datasets. The experimental results reveal that the technique surpasses previous feature selection techniques based on filters and is capable of dealing with imbalanced medical datasets exclusively with recently created fitness functions.

A new PSO method has been presented for enhancing task scheduling efficiency in [15]. Specifically, the authors in that study have proposed a data locality-based ranging-and-tuning-based PSO (RTPSO) method. The "RTPSO" is used to tackle the inertia weight assignment problem that plagues the PSO algorithm for job scheduling. The authors updated RTPSO with a bat pseudocode to boost optimization (RTPSO-B). They used the cloudsim toolkit to develop their algorithm and the findings showed that the proposed method reduced cost and time and enhances utilization of resource as well.

Dinesh Reddy et al. [16] proposed a modified discrete PSO for primary VMs placement. Regarding bandwidth utilisation, size, and memory use of VMs, the suggested approach outperforms recent PSO algorithms. The simulation findings states that the proposed method increases efficiency and lowers service level contract destruction.

In [17], the authors developed a model of swarm intelligence of social spiders with chaotic inertia weight and random selection to shorten the overall makespan and balance load. Their approach avoided local convergence and instead used large scale intelligence searching to get the ideal optimized VMs with balanced resource consumption and a shorter makespan. The authors use a framework called cloudsim to do performance evaluation and simulation for the task supplied by the user. The experimental results stated that the proposed algorithm produced a balanced task distribution with the shortest makespan when compared to other algorithms such as ABC, GA, and PSO.

Boveiri et al. [18] suggested a swarm intelligence technique for scheduling tasks in the Internet of Things applications based on the cloud computing paradigm using the Max-Min Ant System (MMAS). which represents a well-organized variant of the ant colony optimization algorithm family for tackling static task-graph scheduling in homogenous multiprocessor environments and properly manipulating task priority variables to get the best optimal task order. From a performance standpoint, the results demonstrate its efficiency and superiority over traditional competitors.

Jamali et al. in [19] presented particle swarm optimization (PSO) method to schedule tasks in cloud computing compared with first come first serve (FCFS) algorithm and genetic algorithm (GA). The primary goal of this study was to reduce the makespan and waiting time of a given task set. The Cloudsim toolkit package was used to simulate the suggested model, and the results revealed that PSO outperformed GA and FCFS algorithms.

Yuen et al. [31] presented a framework for addressing the comprehensive index tracking problem (IPT) using a collaborative approach based on metaheuristics to optimize this problem globally. The proposed framework's primary process comprises metaheuristics algorithms, which balance a desirable tradeoff between computational resource utilization and solution quality. The GA, PSO, CSO, and DE are applied to the comprehensive ITP in the simulation, and the proposed method achieves a competitive result. Yang et al. [32] developed a dominant cognitive learning particle swarm optimization (PCLPSO) method based on dynamic adjustment strategies to effectively tackle complex optimization problems and improve particle learning effectiveness and diversity. The experimental results confirmed the developed PCLPSO's remarkable feature and demonstrated that the proposed technique is scalable for solving various optimization problems.

Elhady and Tawfeek [20] introduced a hybrid technique for dynamic task scheduling in cloud computing based on swarm intelligence approaches. The goal is to solve the resource management problem by allocating and scheduling cloud computing virtual machines in a way that allows providers to reduce task makespan times. by combining the behaviors of three different swarm intelligence techniques. It takes advantage of the benefits of ant colony behavior (ACO), particle

swarm behavior (PSO), and honeybee foraging behavior (BCO). The experimental results support the proposed hybrid algorithm's strength.

Ramezani et al. in [21] presented multi-objective particle swarm optimization for the cloud computing task scheduling optimization process. Specifically, to give the best solution for the suggested model, the authors devised a multi-objective algorithm based on the multi-objective PSO (MOPSO) method. This is done by extending the Jswarm package into a multi-objective Jswarm (MO-Jswarm) package. In addition, the cloud-sim toolkit is extended by making use of MOJswarm as its task scheduling technique.

Based on these previous studies, it is clear that there is still a great opportunity to make enhancements in health care services based on cloud computing in the IoT environment and find a better way to reduce the makespan time of tasks to provide an efficient method for optimal task scheduling.

### III. METHODOLOGY

This section introduces the SSA and PSO algorithms and then discusses the proposed algorithm (HPSOSSA) in detail.

#### A. SALP SWARM ALGORITHM (SSA)

The Salp Swarm Algorithm (SSA) is a recent meta-heuristic optimization approach proposed by Mirjalili et al. [22]. The main idea of this algorithm is inspired by the swarming behavior of salps in the oceans. The SSA effectively improves the starting population and converges toward the optimal. Even with an undefined search space, it can escape local solutions and subsequently discover a correct approximation of the optimal solution achieved during optimization. SSA strikes a good mix between exploitation and exploration. The salp swarming activity when traveling and exploring for food in the sea is the primary motivating factor of SSA. The Salp's movements and structure are similar to those of the sea organisms, jellyfish. Salps travel in colonies and frequently form a swarm known as a "salp chain". The first salp is called the leader, and the others are called followers. The following mathematical equations (1,2) [22] should be used to determine the leader's position.

$$x_j^1 = \begin{cases} f_j + c_1 ((ub_j - lb_j) c_2 + lb_j) c_3 \geq 0 \\ f_j - c_1 ((ub_j - lb_j) c_2 + lb_j) c_3 < 0 \end{cases} \quad (1)$$

where  $x_j^1$  is the greatest and most feasible solution's position,  $ub_j$ ,  $lb_j$  are the  $j$ th dimension's upper and lower bounds,  $f_j$  is the dimension's food supply location,  $c_2$ ,  $c_3$  are two randomly generated values in the range (0,1) and the most important variable in this algorithm is  $c_1$ , which gradually decreases over generations to allow for high exploration in the initial phases of the optimization procedure, then high exploitation in the final steps.  $c_1$  is defined as:

$$c_1 = 2e^{-\left(\frac{l}{L}\right)^2} \quad (2)$$

where  $L$  is the maximum number of iterations and  $l$  is the current number of iterations, The following mathematical

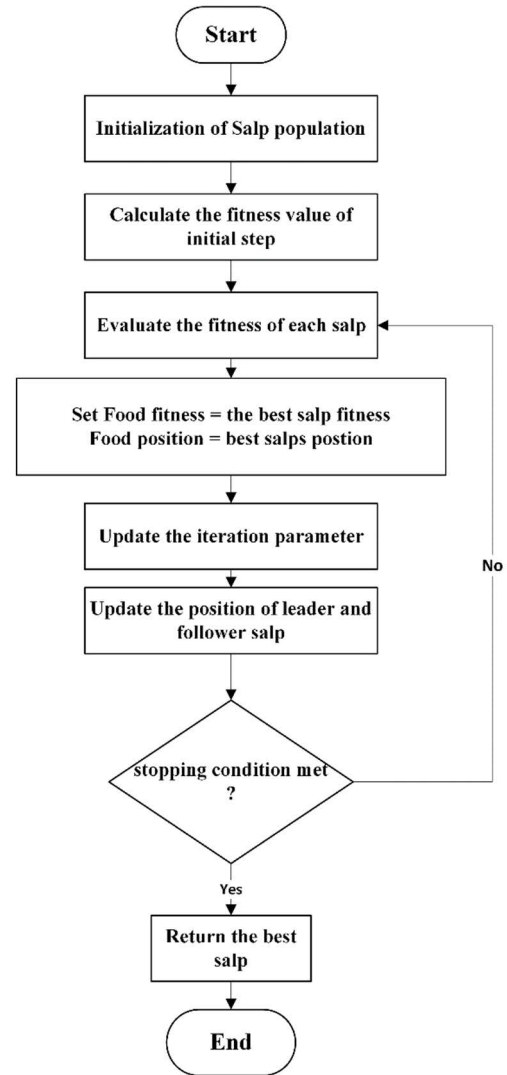


FIGURE 1. Main steps of the SSA.

equation is used to update the locations of the followers:

$$x_j^i = \frac{1}{2} (x_j^i + x_j^{i-1}) \quad (3)$$

where  $x_j^i$  denotes the  $i$ th follower's position in the  $j$ th dimension and  $i \geq 2$ .

SSA starts with the generation of the salp population and then calculates all salps in the initial crowd. Their primary goal is to locate the food source (F), which represents the best salp in the search space. The equation (2) is used to update the location of the food supply on a regular basis. The iteration numbers decrease gradually as the search space is explored and exploitation of the search space begins. The location of the leader is adjusted in reference to the food source, and the position of the following salp is adjusted by using the equation (3) [23]. The salp swarm algorithm (SSA) flow chart is shown in the following Figure 1.

SSA has many advantages, such as combining with other algorithms is surprisingly rewarding. A good rate of



convergence acceleration. Obtaining outstanding solutions in a shorter time frame, appropriate for a wide range of optimization issues and search spaces. Its idea and implementation are easier in concept compared with other heuristic optimization techniques. The time for execution is reasonable. Little adjusting parameters. On the other hand, SSA has some disadvantages, like suffering from premature convergence. There isn't a theoretically convergent frame. The probability distribution changes by generation [24].

### B. PARTICLE SWARM OPTIMIZATION (PSO)

Particle Swarm Optimization (PSO) is a famous meta-heuristic technique to solve optimization issues. The algorithm has been developed in 1995 by Kennedy and Eberhart [25]. PSO's main idea is inspired by particle social behavior such as fish schooling and bird flocking to solve global optimization functions. Furthermore, PSO can handle dynamic task scheduling, workflow scheduling and load balancing. When compared to other meta-heuristic optimization techniques, PSO has the advantage of being simple to build and having few parameters to adjust. In this method, each possible outcome is treated as an agent (particle). Each candidate's velocities and fitness levels are unique. These agents navigate the D-dimensional function region by learning from the agents' previous experiences.

PSO employs particles to perform elect solutions to problems, with each particle attempting to find the most advantageous solution in the search space. The steps to find the optimum value using the PSO algorithm are shown in Figure 2 and can be described as follows:

1. Randomly Initialize and assign the positions and velocities of each particle within a problem space.
2. Evaluate each particle's objective function and calculate the best position and best global position.
3. A particle's position and velocity are updated based on its own inertia and experience, with the goal of detecting the optimal solution to the problem.

An agent demonstrates a probable outcome, which changes its position and position based on mathematical equations. (4) & (5):

$$v_i^{k+1} = w \times v_i^k + c_4 r_1 (p_i^k - x_i^k) + c_5 r_2 (g_{best} - x_i^k) \quad (4)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (5)$$

where  $v_{ij}^k$  denotes the particle velocity when  $i \in [1, 2, \dots, n_x]$  in  $j$  dimension at step of period  $k$ ,  $w$  is the constant of inertia,  $p_i^k$  denotes the agent's personal best position,  $g_{best}$  denotes the neighbor agent's best position,  $x_{ij}^k$  denotes the particle  $i$ 's position in dimension  $j$  at step of period  $k$ . he acceleration coefficients  $w$  and  $c_4$  and  $c_5$  are constants and values of random  $r_{1j}, r_{2j} \in U(0, 1)$  within the range  $[0, 1]$  at step of period  $k$ .

4. Check to see if the solution is optimal. If convergence has been achieved, then it comes to an end; otherwise, it returns to step 2.

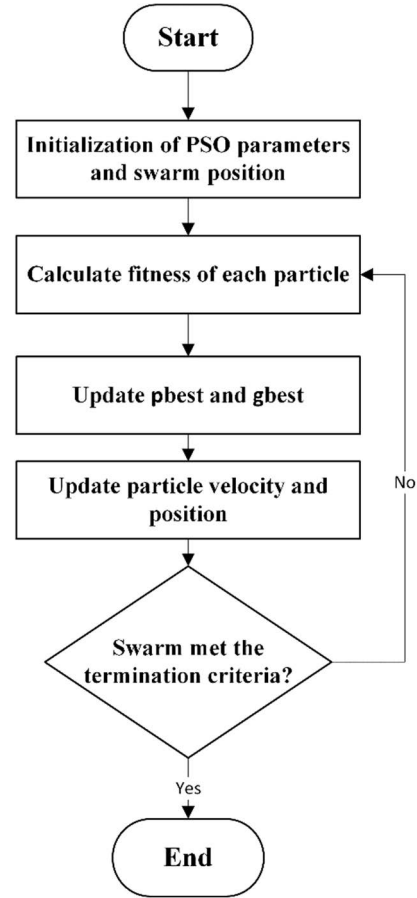


FIGURE 2. Main steps of the PSO algorithm.

PSO offers the advantage of having no overlapping or mutation calculations. Another feature is that the speed of the particles can be used to carry out the search. The speed of research is very fast, and the calculations in PSO are very simple. It has a higher optimization capability and can be completed quickly. In contrast, the PSO algorithm has disadvantages, such as easily suffering from partial optimism, which results in less precise control of its direction and speed. The algorithm thus becomes incapable of solving scattering and optimization problems, as well as non-coordinated system problems [26].

### C. THE PROPOSED HYBRID PSOSSA ALGORITHM

The HPSOSSA is a hybrid optimization algorithm that combines the benefits of the PSO and SSA algorithms to overcome the shortcomings of these two techniques. Where each individual optimization algorithm cannot solve every form of a real-world problem, these standard techniques may fail to detect the target of any complex functions due to their weaknesses in the search area. As a result, in response to the present need for real-world applications, researchers from diverse domains are developing new modified and hybrid versions to fix the flaws in current versions, allowing these

algorithms to deliver the best global optimal solution for complicated problems. According to the literature review and related works to learn the advantages and disadvantages of each optimization technique, we suggest a hybrid particle swarm optimization technique based on the SSA method that directly ignores the impact of velocity on particle position and prevents performance loss brought on by inertia weight, acceleration factor, and other parameters that are unable to establish their optimal value. The proposed HPSOSSA technique will determine the best score for the complex optimization functions by eliminating the PSO algorithm's drawbacks while incorporating the SSA algorithm's benefits, which will help quickly capture the global optimal solution while ignoring local optima in the search area during the search process. As a result, by accelerating the search, we can improve the task scheduling process and get better results.

HPSOSSA begins by initializing the population randomly according to the problem. Then, each search agent in the crowd is rated according to how effective their own position is. The objective function is used to evaluate the fitness value of each search agent. Then, based on its fitness value in the search space, each search agent chooses the next new site. Whereas PSO works on updating the crowd and selecting a preferred solution, SSA updates and improves the selected option. The preferred solution is selected and forwarded to the next iteration. The refinery procedure is carried out and repeated until the stopping conditions are met in order to arrive at the best solution to the task scheduling problem. The flow chart in Figure 3 shows the working process of the proposed HPSOSSA algorithm.

#### IV. THE PROPOSED MODEL

There are a lot of potential enhancement opportunities in the medical field in the Cloud-IoT environment. This can be achieved by integrating cloud and IoT to solve the HCS's issues in many fields, such as smart hospitals, medicine control, and other remote medical services, which can help and save the lives of many people. In this section, we will introduce an intelligent model paradigm for HCS that incorporates certain major elements of cloud computing and

IoT. The key elements of the proposed model are shown in figure 4 and are discussed in the following.

- **The cloud-IoT environment.** Users of HCS use a variety of devices such as (PCs, Laptops, Smartphones, Tablets, digital sensors, and so on) to easily send a variety of medical requests (tasks) through cloud computing to obtain various medical services such as retrieving patients' data, online doctor, disease diagnosis, electronic medical records (EMR), and so on. The IoT environment serves as a data collection where its endpoints collect the data and share it with the cloud system. The cloud infrastructure is organized within the cloud data centers, which are made up of several physical servers (application hosts).

#### Algorithm 1 PSOSSA Algorithm Pseudocode

---

```

1: Set the initial constant;
2: Number of search agents  $s = 50$ ,  $l = 2$ ,  $L = 50 - 500$ ,
 $W_{max} = 0.9$ ,  $w_{min} = 0.2$  and  $v_{max} = 6$ 
3: Initialize population  $x_i$ , ( $i = 1, 2 \dots n$ )
4: For generation from 1 to maxiter
5: do
6: Calculate all salp in the crowd
7: Represent the best salp as F
8: Update the value of  $c_1$  by using the mathematical equation (2)
9: For (all salp(x,)) do
10: Calculate the velocity of the follower's by using the equation (6):
11:

$$v_i^{k+1} = w \times v_i^k + c_1 \times \left( \text{salp}_j^{\text{fitness}} - \text{food}_j^{\text{position}} \right) \left\{ \begin{array}{l} \text{if } v_i^k > v_{\max} \\ \text{then } v_i^k = v_{\max} \\ \text{end} \\ \text{if } v_i^k = -v_{\max} \\ \text{then } v_i^k = -v_{\max} \\ \text{end} \end{array} \right. \quad (6)$$

12: if  $x_i$  is a leader
13: then
14: Update the position of the leader by using the mathematical equation (1)
15: Else
16: Update the position of the follower's by using the mathematical equation (7);

$$x_j^i = \frac{1}{3} \left( x_j^i + x_j^{i-1} + v_i^{k+1} \right) \quad (7)$$

17: end
18: adjust the search agents based on the upper and lower bounds of variables
19: end
20: Return F

```

---

The application host in the cloud data center provides the SaaS and hosts one or more VMs that have a set of resources that can be used to process the coming user's requests (cloudlets).

- **The broker.** according to the proposed model, the broker serves as a bridge between users and the cloud. It accepts and sends the user requests to the task manager component.
- **Task Manager** organizes incoming requests and notifies the user who submitted them of the status of each task. After that, the task manager sends task requests to the task scheduler.
- **Task scheduler** employs the proposed hybrid scheduling algorithm HPSOSSA to allocate incoming cloudlets

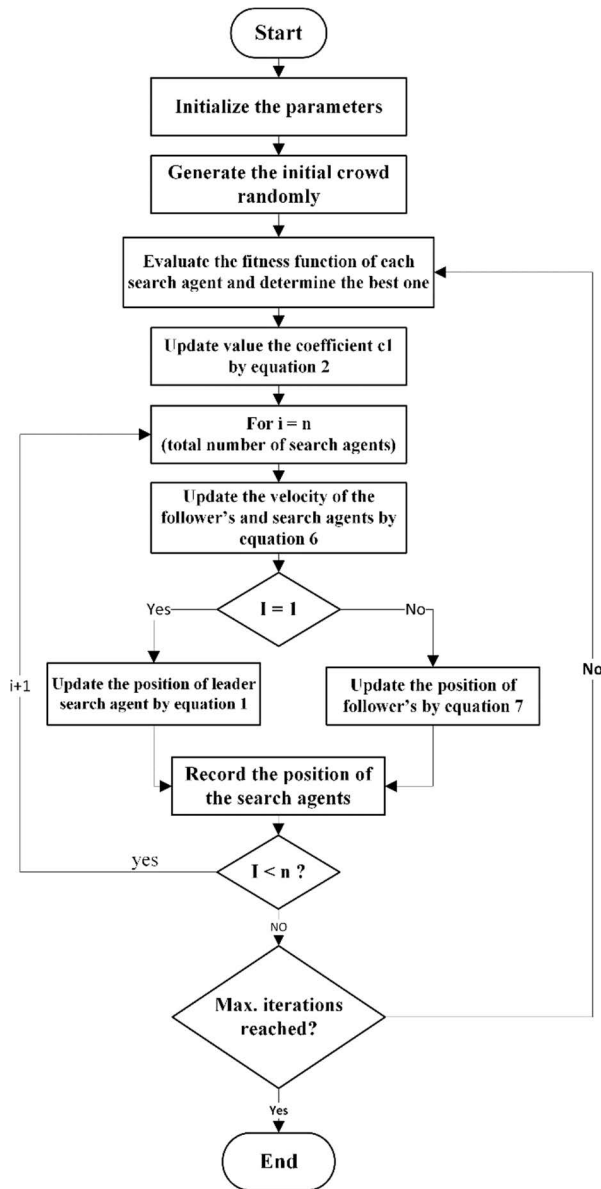


FIGURE 3. Hybrid PSOSSA algorithm.

to the available and appropriate VMs in order to improve the task scheduling process. Doing so helps in minimizing the makespan to support healthcare providers in the IoT by minimizing the time it takes to process medical requests and maximizing the use of medical resources. To acquire a suitable allocation design for the proposed technique, the data center transmits the tasks and virtual machines in the form of two different collections one for cloudlets = {task1, task2, taskn ...} and one for VMs = {vm1, vm2, ...}, each with their own specifications as shown in the following figure 4.

## V. EXPERIMENTAL RESULTS

This section discusses the implementation details and experimental results of the proposed algorithm.

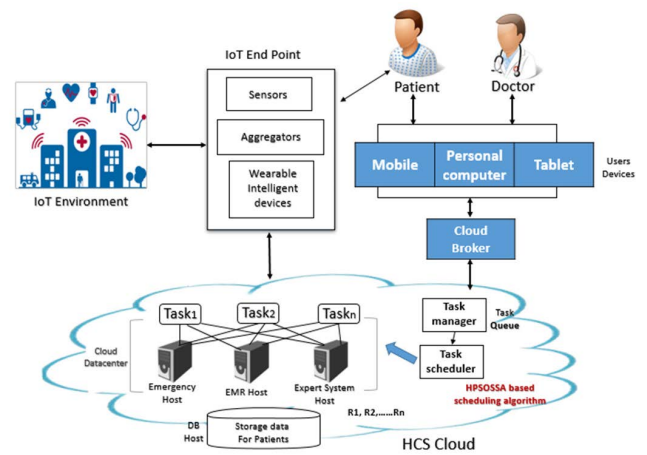


FIGURE 4. HCS Intelligent cloud computing model.

We have performed different scenarios to assess the efficiency of the suggested algorithm HPSOSSA in terms of makespan, waiting time and resource utilization.

### A. DATASET USED AND SIMULATION SETUP

For our experiments, We used an HCSP-based dataset made up of VMs and cloudlets data files [27], that can be used for cloud computing task scheduling procedures. The dataset provides two types of information used for the simulation and has four various circumstances: c-hilo, i-hilo, c-lohi, and i-lohi. Because each dataset has a varied level of task and resource heterogeneity, it contains various types of data like cloudlets (1024). Each value denotes the task or cloudlet's size in MIs, while the processing power is represented by the VM size in available MIPS [28].

CloudSim is a modeling and simulation framework for cloud computing infrastructures and services [26], [27]. To assess the proposed algorithm's effectiveness, algorithms are simulated using the CloudSim 3.0.3 simulation toolkit. All tasks are supposed to run independently, and each task is supposed to run on only one VM at a time. According to the literature review, researchers have usually used constants and standard parameters to simulate their experiments in the CloudSim environment. We carried out a series of experiments to evaluate the performance of HPSOSSA and compare it to the most commonly used meta-heuristics, namely; PSO, SSA, ACO, and HPSO-GA. These algorithms are run in the same simulation environment for a fair comparison, and parameters and fixed values are adjusted, as shown in Table 1.

The number of cloud tasks submitted by users changes in the range of 400–1000, generated randomly from the dataset file we mentioned above, while the number of virtual machines in the cloud data center and other parameters remain unchanged to standardize the test environment. Table 2 shows the parameters for the proposed algorithm's configuration.

**TABLE 1.** Parameters setting of the CloudSim environment.

CloudSim Entity	Parameter	Value
Cloudlets	No. of cloudlets	400 – 1000
	Cloudlet length	100–10000 MIs
	File size	1000 MB
	Output size	1000 MB
VM	No. of VMs	40
	RAM	512 MB
	Storage capacity	10 GB
	VM processing power	800 MIPs
	Bandwidth	1 GB/s
	VM policy	Time-shared
	VMM	Xen
	O.S.	Linux
	No. of CPUs	1
Host	No. of Datacenters	2
	No. of host in datacenter	1
	Host RAM	2 GB
	Storage	1 TB
	Bandwidth	10GB/s

**TABLE 2.** HPSOSSA algorithm parameters.

Parameter	Definition	Values
$s$	No. of search agents	50
$l$	Current iteration	2
$L$	Max no. of Iterations	50 – 500
$w_{max}$	Max – Inertia	0.9
$w_{min}$	Min – Inertia	0.2
$v_{max}$	Max Velocity	6

**TABLE 3.** Makespan values.

Tasks	PSO	ACO	SSA	HPSOGA	HPSOSSA
400	119.52	129.58	126.99	113.41	100.83
500	136.78	146.03	144.26	124.56	110.97
600	147.23	167.22	162.29	131.37	115.4
700	187.92	208.87	200.04	146.02	117.15
800	217.21	227.19	221.04	162.41	123.01
900	231.97	239.24	239.03	178.19	132.18
1000	243.21	263.79	261.06	191.23	139.1

## B. EXPERIMENTS AND RESULT ANALYSIS

To assess the HPSOSSA algorithm's performance, the simulation experiment was executed seven times. At each time, we used a different number of user tasks (cloudlets). The average of the makespan, waiting time, and cost for the HPSOSSA algorithm were compared with that of PSO, ACO, SSA and HPSO-GA algorithms. The results in table 3 show the makespan values as the number of tasks increased from 400 to 1000 for each simulation separately.

The simulation was repeated multiple times, and the results confirm that HPSOSSA performs better with the makespan minimization, as shown in figure 5.

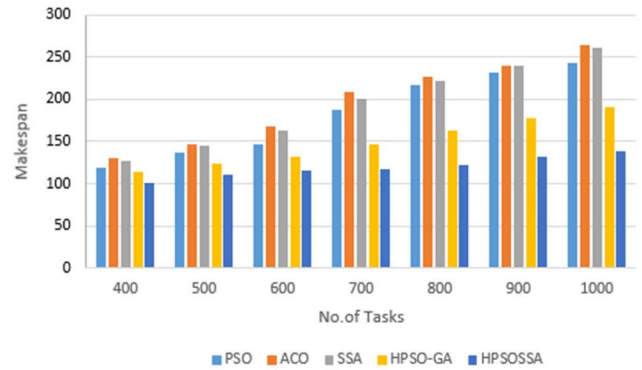
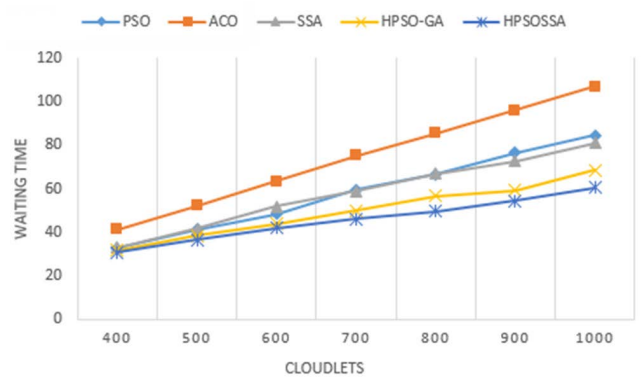
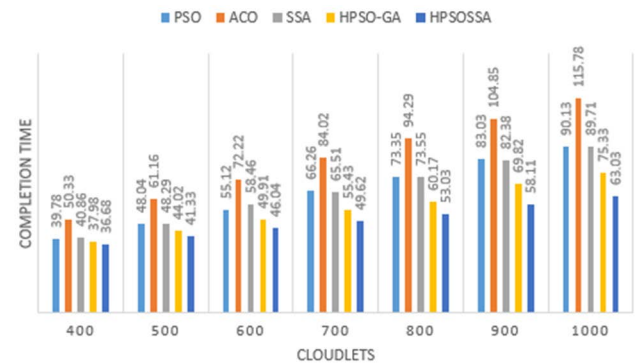
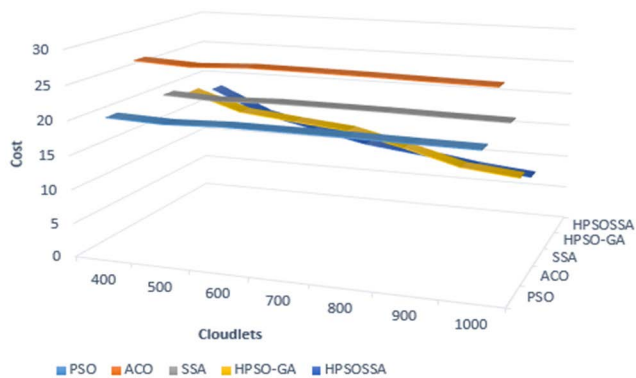
**FIGURE 5.** Makespan comparison for HPSOSSA, HPSO-GA, SSA, ACO, and PSO.**FIGURE 6.** Comparison of the average waiting time for HPSOSSA, HPSO-GA, SSA, ACO, and PSO.**FIGURE 7.** Comparison of the average completion time for HPSOSSA, HPSO-GA, SSA, ACO, and PSO.

Figure 6 shows that when the number of cloudlets is between 400 and 1000 tasks, the average waiting time of HPSOSSA for each experiment is the lowest as compared to PSO, ACO, SSA and HPSO-GA.

Figure 7 compares the average completion time of the proposed algorithm with the other four algorithms, and the results prove that HPSOSSA provides the minimum average completion time in all experiments done.

Figure 8 depicts the association between the average cost and the increasing number of cloudlets. It is clear from the





**FIGURE 8.** Comparison of the average cost value HPSOSSA, HPSO-GA, SSA, ACO, and PSO.

**TABLE 4.** Comparison on PIR (%) based on total average makespan.

	ACO	SSA	PSO	HPSOGA	HPSOSSA
Total average makespan	1381.92	1354.71	1283.84	1047.19	838.64
PIR over ACO		2.01%	7.63%	31.96%	64.78%
PIR over SSA			5.52%	29.36%	61.52%
PIR over PSO				22.59%	53.08%
PIR over HPSO-GA					24.86%

simulation results that the proposed algorithm provides the lowest average cost value of all presented algorithms.

The performance improvement rate (PIR) of the proposed HPSOSSA algorithm over the other four comparable optimization scheduling algorithms using the HCSP-based dataset shows our proposed algorithm has managed to achieve a PIR% of 64.78% for ACO, 61.52% for SSA, 53.08% for PSO and 24.86% for HPSO-GA in terms of total average makespan. The remaining experimental results of the PIR analysis obtained by the four scheduling algorithms are reported in table 4.

These results demonstrate the importance of the proposed algorithm in achieving the best solution more quickly and efficiently than other algorithms by combining features of appropriate diversity and fast convergence. Although it may have some limitations, such as the difficulty of becoming in local or global optima, it is unsuitable for highly complicated problems or functions. It is incapable of dealing with its drawbacks, such as slow convergence speed and slow diversity.

## VI. CONCLUSION AND FUTURE WORK

This paper introduced a meta-heuristic hybrid particle swarm optimization and salp swarm (HPSOSSA) algorithm, as part of an intelligent model. for enhancing the task scheduling problem in HCS based on cloud computing and IoT environment. The proposed HPSOSSA algorithm combines PSO and SSA advantages to find an optimal solution to help HCS

users execute their medical requests with less time and cost. The research was implemented using the cloudsim simulation toolkit and the outcomes of the experiments prove the efficiency of the proposed model compared with other algorithms in terms of makespan, waiting time, and cost. In future work, we are planning to extend the research scale by adding new evaluation parameters and using new optimization techniques to reach the optimal task scheduling.

## REFERENCES

- [1] E. El-Shafeiy and A. Abohany, "A new swarm intelligence framework for the Internet of Medical Things system in healthcare," in *Swarm Intelligence for Resource Management in Internet of Things*. Amsterdam, The Netherlands: Elsevier, 2020, pp. 87–107, doi: [10.1016/B978-0-12-818287-1.00010-3](https://doi.org/10.1016/B978-0-12-818287-1.00010-3).
- [2] M. Elhoseny, A. Abdelaziz, A. S. Salama, A. M. Riad, K. Muhammad, and A. K. Sangaiah, "A hybrid model of Internet of Things and cloud computing to manage big data in health services applications," *Future Gener. Comput. Syst.*, vol. 86, pp. 1383–1394, Sep. 2018, doi: [10.1016/j.future.2018.03.005](https://doi.org/10.1016/j.future.2018.03.005).
- [3] N. Nayar, S. Ahuja, and S. Jain, "Swarm intelligence and data mining: A review of literature and applications in healthcare," in *Proc. 3rd Int. Conf. Adv. Informat. Comput. Res. (ICAICR)*, 2019, pp. 1–7, doi: [10.1145/3339311.3339323](https://doi.org/10.1145/3339311.3339323).
- [4] I. Strumberger, M. Tuba, N. Bacanin, and E. Tuba, "Cloudlet scheduling by hybridized monarch butterfly optimization algorithm," *J. Sensor Actuator Netw.*, vol. 8, no. 3, p. 44, Aug. 2019, doi: [10.3390/jsan8030044](https://doi.org/10.3390/jsan8030044).
- [5] P. Singh, M. Dutta, and N. Aggarwal, "A review of task scheduling based on meta-heuristics approach in cloud computing," *Knowl. Inf. Syst.*, vol. 52, no. 1, pp. 1–51, Apr. 2017, doi: [10.1007/s10115-017-1044-2](https://doi.org/10.1007/s10115-017-1044-2).
- [6] M. Kalra and S. Singh, "A review of metaheuristic scheduling techniques in cloud computing," *Egyptian Informat. J.*, vol. 16, no. 3, pp. 275–295, 2015, doi: [10.1016/j.eij.2015.07.001](https://doi.org/10.1016/j.eij.2015.07.001).
- [7] R. A. Al-Arasi and A. Saif, "HTSCC a hybrid task scheduling algorithm in cloud computing environment," *Int. J. Comput. Technol.*, vol. 17, no. 2, pp. 7236–7246, Aug. 2018, doi: [10.24297/ijct.v17i2.7584](https://doi.org/10.24297/ijct.v17i2.7584).
- [8] R. Jain and N. Sharma, "A QoS aware binary salp swarm algorithm for effective task scheduling in cloud computing," in *Progress in Advanced Computing and Intelligent Engineering*, vol. 1199, C. R. Panigrahi, B. Pati, P. Mohapatra, R. Buyya, and K.-C. Li, Eds. Singapore: Springer, 2021, pp. 462–473, doi: [10.1007/978-981-15-6353-9\\_43](https://doi.org/10.1007/978-981-15-6353-9_43).
- [9] X. Fu, Y. Sun, H. Wang, and H. Li, "Task scheduling of cloud computing based on hybrid particle swarm algorithm and genetic algorithm," *Cluster Comput.*, May 2021, doi: [10.1007/s10586-020-03221-z](https://doi.org/10.1007/s10586-020-03221-z).
- [10] B. Kruekaew and W. Kimpan, "Enhancing of artificial bee colony algorithm for virtual machine scheduling and load balancing problem in cloud computing," *Int. J. Comput. Intell. Syst.*, vol. 13, no. 1, p. 496, 2020, doi: [10.2991/ijcis.d.200410.002](https://doi.org/10.2991/ijcis.d.200410.002).
- [11] E. Safari, S. O. Pourhashemi, and M. Gharahkhani, "A hybrid swarm particle optimization algorithm for task scheduling in cloud computing," *Iran Telecommun. Res. Center, Tehran, Iran, Tech. Rep.*, 2020, vol. 7, no. 4, p. 9.
- [12] P. Kumar and K. Silambarasan, "Enhancing the performance of healthcare service in IoT and cloud using optimized techniques," *IETE J. Res.*, vol. 68, no. 2, pp. 1475–1484, 2019, doi: [10.1080/03772063.2019.1654934](https://doi.org/10.1080/03772063.2019.1654934).
- [13] A. Abdelaziz, A. S. Salama, and A. M. Riad, "A swarm intelligence model for enhancing health care services in smart cities applications," in *Security in Smart Cities: Models, Applications, and Challenges*, A. E. Hassanien, M. Elhoseny, S. H. Ahmed, and A. K. Singh, Eds. Cham, Switzerland: Springer, 2019, pp. 71–91, doi: [10.1007/978-3-030-01560-2\\_4](https://doi.org/10.1007/978-3-030-01560-2_4).
- [14] V. Ramaswamy and S. Mukherjee, "An effective clinical decision support system using swarm intelligence," *J. Supercomput.*, vol. 76, no. 9, pp. 6599–6618, Sep. 2020, doi: [10.1007/s12277-019-02888-5](https://doi.org/10.1007/s12277-019-02888-5).
- [15] R. Valarmathi and T. Sheela, "Ranging and tuning based particle swarm optimization with bat algorithm for task scheduling in cloud computing," *Cluster Comput.*, vol. 22, no. S5, pp. 11975–11988, Sep. 2019, doi: [10.1007/s10586-017-1534-8](https://doi.org/10.1007/s10586-017-1534-8).
- [16] V. Dinesh Reddy, G. R. Gangadharan, and G. S. V. R. K. Rao, "Energy-aware virtual machine allocation and selection in cloud data centers," *Soft Comput.*, vol. 23, no. 6, pp. 1917–1932, Mar. 2019, doi: [10.1007/s00500-017-2905-z](https://doi.org/10.1007/s00500-017-2905-z).

- [17] V. M. Arul Xavier and S. Annadurai, "Chaotic social spider algorithm for load balance aware task scheduling in cloud computing," *Cluster Comput.*, vol. 22, no. S1, pp. 287–297, Jan. 2019, doi: [10.1007/s10586-018-1823-x](https://doi.org/10.1007/s10586-018-1823-x).
- [18] H. R. Boveiri, R. Khayami, M. Elhoseny, and M. Gunasekaran, "An efficient swarm-intelligence approach for task scheduling in cloud-based Internet of Things applications," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 9, pp. 3469–3479, 2019, doi: [10.1007/s12652-018-1071-1](https://doi.org/10.1007/s12652-018-1071-1).
- [19] S. Jamali, F. Alizadeh, and S. Sadeqi, "Task scheduling in cloud computing using particle swarm optimization," Univ. Mohaghegh Ardabili, Ardabil, Iran, Tech. Rep., 2016, p. 7.
- [20] G. F. Elhady and M. A. Tawfeek, "A comparative study into swarm intelligence algorithms for dynamic tasks scheduling in cloud computing," in *Proc. IEEE 7th Int. Conf. Intell. Comput. Inf. Syst. (ICICIS)*, Dec. 2015, pp. 362–369.
- [21] F. Ramezani, J. Lu, and F. Hussain, "Task scheduling optimization in cloud computing applying multi-objective particle swarm optimization," in *Service-Oriented Computing*, vol. 6470, P. P. Maglio, M. Weske, J. Yang, and M. Fantinato, Eds. Berlin, Germany: Springer, 2013, pp. 237–251, doi: [10.1007/978-3-642-45005-1\\_17](https://doi.org/10.1007/978-3-642-45005-1_17).
- [22] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017, doi: [10.1016/j.advengsoft.2017.07.002](https://doi.org/10.1016/j.advengsoft.2017.07.002).
- [23] G. I. Sayed, G. Khoriba, and M. H. Haggag, "A novel chaotic salp swarm algorithm for global optimization and feature selection," *Appl. Intell.*, vol. 48, no. 10, pp. 3462–3481, 2018, doi: [10.1007/s10489-018-1158-6](https://doi.org/10.1007/s10489-018-1158-6).
- [24] L. Abualigah, M. Shehab, M. Alshinwan, and H. Alabool, "Salp swarm algorithm: A comprehensive survey," *Neural Comput. Appl.*, vol. 32, no. 15, pp. 11195–11215, Aug. 2020, doi: [10.1007/s00521-019-04629-4](https://doi.org/10.1007/s00521-019-04629-4).
- [25] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, 1995, pp. 1942–1948, doi: [10.1109/ICNN.1995.488968](https://doi.org/10.1109/ICNN.1995.488968).
- [26] D. P. Rini and S. M. Shamsuddin, "Particle swarm optimization: Technique, system and challenges," *Int. J. Appl. Inf. Syst.*, vol. 1, no. 1, pp. 33–45, Sep. 2011.
- [27] M. Ibrahim, S. Nabi, A. Baz, N. Naveed, and H. Alhakami, "Toward a task and resource aware task scheduling in cloud computing: An experimental comparative evaluation," *Int. J. Netw. Distrib. Comput.*, vol. 8, no. 3, pp. 131–138, 2020.
- [28] M. Ibrahim, "Dataset for task scheduling in cloud using cloudsims," *IEEE Dataport*, Feb. 2020. [Online]. Available: <https://iee-dataport.org/documents/dataset-task-scheduling-cloud-using-cloudsim>
- [29] R. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Softw., Pract. Exper.*, vol. 41, no. 1, pp. 23–50, Aug. 2011, doi: [10.1002/spe.995](https://doi.org/10.1002/spe.995).
- [30] E. Rani and H. Kaur, "Study on fundamental usage of CloudSim simulator and algorithms of resource allocation in cloud computing," in *Proc. 8th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2017, pp. 1–7, doi: [10.1109/ICCCNT.2017.8203998](https://doi.org/10.1109/ICCCNT.2017.8203998).
- [31] M.-C. Yuen, S.-C. Ng, M.-F. Leung, and H. Che, "A Metaheuristic-based framework for index tracking with practical constraints," *Complex Intell. Syst.*, Dec. 2021, doi: [10.1007/s40747-021-00605-5](https://doi.org/10.1007/s40747-021-00605-5).
- [32] Q. Yang, Y. Jing, X. Gao, D. Xu, Z. Lu, S.-W. Jeon, and J. Zhang, "Predominant cognitive learning particle swarm optimization for global numerical optimization," *Mathematics*, vol. 10, no. 10, p. 1620, May 2022, doi: [10.3390/math10101620](https://doi.org/10.3390/math10101620).



computing, artificial intelligence, and related topics.

**KARIM M. HASSAN** received the B.Sc. degree in information systems program from the Higher Institute of Computer Science and Information Technology, El-Shorouk Academy, Cairo, Egypt, in 2015. He is currently pursuing the master's degree in software engineering program with Helwan University, Egypt. He is also a Teaching Assistant with the Higher Institute of Computers and Information Technology, El-Shorouk Academy. His research interests include cloud computing, artificial intelligence, and related topics.



She worked as a Lecturer with the Faculty of Computers and Information, Helwan University, in 2010. She held the position of software engineering program coordinator, from 2016 to August 2020. She supervised many master's and doctoral dissertations in the fields of information systems, machine learning, artificial intelligence, medical informatics, and bioinformatics.

**A. ABDO** received the bachelor's degree in computers and information, the master's degree in information systems, and the Ph.D. degree in bioinformatics from the Department of Information Systems, Faculty of Computers and Information, Helwan University, in 2000, 2004, and 2010, respectively.



research interests include machine learning, big data analytics, and data management.

**AHMED YAKOUB** received B.Sc. degree (Hons.) in information systems, the M.Sc. degree, and the Ph.D. degree in information systems from Helwan University, in 2008, 2014, and 2019, respectively. He worked as a Teaching Assistant with the Faculty of Computers and Information, Helwan University, from 2010 to 2014, where he was a Lecturer Assistant, from 2014 to 2016. He is currently a Lecturer with the Faculty of Computers and Artificial Intelligence, Helwan University. His

...