ISSN: 0067-2904

Hybrid Resources Allocation Algorithm for Clinical Monitoring System

Taghreed Abdelhamid Essa *, Karim Q. Hussain
Department of Computer Science, College of Science, University of Mustansiriya, Baghdad, Iraq

Received: 11/9/2024 Accepted: 14/ 8 /2025 Published: 30/ 7/2026

Abstract

One of the most significant technologies that businesses currently rely on is cloud computing. Cloud computing has emerged as a critical need that cannot be ignored. From this moment on, cloud computing advancement was evaluated by enhancing efficiency and resolving issues. In this paper, the problem of resource allocation was reviewed. Cloud computing resources have a role in scheduling incoming tasks on existing resources and exploiting them optimally, as clients need more sharing resources and on-demand services, a hybrid particle swarm optimization and tabu search algorithms (PSO-TS) is proposed to allocate tasks to resources in healthcare originations, which aims to reduce the cost and makespan over the homogeneous resources in cloud computing environment, results shows a decrease of the average completion time and cost needed for execution these tasks and reduce the makespan for scheduling algorithms in comparison with FCFS, SJF, RR, and PSO algorithms, as well as, the results are improved with the proposed hybrid algorithm which shows faster solution and higher quality and energy efficient compared to other algorithms.

Keywords: Hybrid algorithms; particle swarm optimization; Tabu search; resource allocation; cloud computing

خوارزمية هجينة لتخصيص الموارد في المراقبة الصحية

تغريد عبد الحميد عيسى * ، كريم قاسم حسين
قسم الحاسوب، كلية العلوم، الجامعة المستنصرية، بغداد، العراق

الخلاصة

تعتبر الحوسبة السحابية من أهم التقنيات التي تعتمد عليها الشركات في الوقت الحالي، حيث أصبحت الحوسبة السحابية ضرورة ملحة لا يمكن الاستغناء عنها. ومن هذا المنطلق، تم الاهتمام بالتطوير في الحوسبة السحابية من خلال تحسين الأداء وحل المشكلات. في هذا البحث تم استعراض مشكلة تخصيص الموارد لموارد الحوسبة السحابية والتي لها دور مهم في جدولة المهام الواردة على الموارد الموجودة واستغلالها على النحو الأمثل، حيث يحتاج العملاء إلى مزيد من مشاركة الموارد والخدمات حسب الطلب والتي يتم توفيرها من بيئة الحوسبة السحابية. تم انجاز خوارزمية هجينة بين (particle swarm optimization) وبين خوارزمية (Tabu search) لتطبيقها في منشآت الرعاية الصحية، والذي يهدف إلى تقليل التكلفة والوقت المطلوب لإنجاز المهام في بيئة الحوسبة السحابية، حيث اظهرت النتائج انخفاض التكلفة اللازمة لتنفيذ هذه المهام مع تقليل متوسط وقت انجاز المهام وقد تم مقارن النتائج بخوارزميات FCFS و SJF و RR و PSO، تم تحسين

*Email: taghreedabdelhameedessa@gmail.com

النتائج باستخدام الخوارزمية الهجينة المقترحة والتي تظهر حل أسرع وجودة أعلى وكفاءة في استهلاك الطاقة مقارنة بالخوارزميات الأخرى.

Introduction

The field of cloud computing has been getting great attention recently. The sharing concept of computing resources on the cloud over the internet, such as software, storage, and infrastructure, becomes critical for companies and individuals, with the benefits of eliminating the need for physical resources and paying only for the types of services [1-2]. Cloud computing technology refers to these services, which are provided as platform as a service (PaaS), infrastructure as a service (IaaS), and software as a service (SaaS)[3-5].

Intelligent monitoring is one of the most important virtualized services and resources in cloud computing architecture, besides scalability, transparency, and security[6-7]. A cloud computing system is considered a hybrid design that combines event-driven architecture (EDA) and service-oriented architecture (SOA) [8- 9]. In the current time, demanding technologies are rising on new strategies and features with advanced cloud solutions with great capability [10-11].

A lot of scientific workflow domains, such as healthcare organizations, consist of sequences of tasks that process a set of data from initiation to completion through a specific time and path[12]. To process these types of big data from medical records, transaction patient status, emails, and other applications, there is a need for a highly flexible cloud that can distribute the resources and be able to allocate them efficiently[13],[4]. Cloud computing revolutionizes the healthcare industry by offering a scalable solution, with applications depending on cloud computing technology, such as telemedicine, health information exchange, medical imaging, and electronic health records (EHRs) [14-15].

Technology that effectively distributes cloud resources to different jobs, services, or applications across the internet is referred to as cloud scheduling. It results in improved system load balancing, performance optimization, energy conservation, and time and cost savings [16]. Resource allocation in a cloud computing system involves balancing the demands across multiple virtual machines, servers, and containers. Resource allocation has two types: dynamic resource allocation, which is resources that can be adjusted based on current workload and demand, and static resource allocation, which refers to resources that can be predefined for certain applications or certain tasks [17-18].In cloud computing, resource allocation techniques are strategies and methods that are used to distribute the resources, such as CPU, memory, storage, and network bandwidth, among multiple tasks required from applications or services with optimal performance, cost, and efficiency[19].

The major contribution of this work is as follows:

- A hybrid Particle Swarm Optimization and Tabu Search algorithms (PSO-TS) are proposed to allocate task resources in healthcare organizations.
- A simulation-based performance for the proposed hybrid PSO-TS scheduling technique is conducted, with different basic algorithms, and results are discussed, Best makespan, tTotal completion time, Average waiting time, Total cost, Energy efficiency.

The environment of task scheduling:

A system created to effectively manage task execution, guaranteeing optimal resource usage, meeting deadlines, and managing dependencies, is known as a task scheduling environment. It is essential in several areas, including workflow management, cloud computing, and operating systems. Three levels make up this environment. The first level is the task scheduling stage, which consists of a number of smaller tasks that are divided based on user demands using a cloud computing algorithm. These tasks are then assigned to available resources, processed, and the results are returned to the users. In order to attain the optimal minimum makespan, the second scheduling level is in charge of assigning these tasks to appropriate resources. All of these functions are carried out by a group of virtual machines (VMs) at the third level as shown in Fig 1[18].

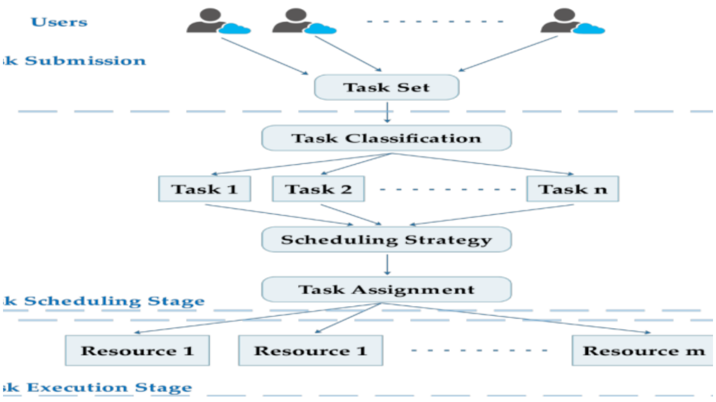

Figure 1: Task scheduling environment stages

Meta-heuristic algorithms:

Meta-heuristic algorithms are high-level problem-independent algorithmic frameworks that provide a set of guidelines or strategies to develop heuristic optimization algorithms. These are typically used to find near-optimal solutions for complex optimization problems where traditional methods (like linear programming) are not effective or feasible [20].

The proposed scheduling algorithms in CloudSim done to ensure realistic and reproducible evaluation using synthetic workloads, and to demonstrate the potential of hybrid metaheuristic algorithms in improving the efficiency, cost-effectiveness, and sustainability of cloud data center operations, also, to evaluate and the performance of proposed hybrid algorithm with swarm intelligence technique (PSO with Tabu) and traditional scheduling algorithms (FCFS, SJF, RR) based on different key performance metrics [21].

Particle Swarm Optimization (PSO)

Proposed by Kennedy and Eberhart in 1995, a popular meta-heuristic optimization technique, basically, a swarm-based algorithm that exploits social behavior like birds flocking, is particularly effective for solving discrete and continuous optimization problems. Each bird, while searching randomly for food, and all birds in the flock can share their discoveries to help other birds find the best hunt. In mathematical equations, this means each bird tries to find the best solution in the space. This is a heuristic solution because the global optimal solution can’t be proven for real. However, the solution found by the PSO algorithm is quite close to the global optimum[22].

A vast number of homogeneous agents, which are known as particles (n), which are interacting locally with themselves and with the environment, are typically known as swarms (S_n). Each particle has its velocity (V_i) and its potential (P_i) to view the problem when flying in the problem space. When a combination of particles is reached with distribution in their location to find the best current solution (P_{best}) and (g_{best}) to the whole swarm, which leads with probability to approaching the best function optimum $\int$ [23-24].
Recently, the swarm-based techniques family, which depends on nature-inspired algorithms, has proven its ability to produce low-cost, fast, robust solutions for complex problems, which are classified as a major category of artificial intelligence, which refers to tasks that require human intelligence [25].

Two equations are involved with the PSO algorithm, the velocity equation, where each particle updates its velocity using the individual current influence and the best global position:

$$V_i^{(t+1)} = V_i^t + c_1r_1 (pbest_i^t - p_i^t) + c_2r_2 (gbest^t - p_i^t) \quad \ldots\ldots\ldots\ldots (1)$$

The second equation for particles to update its positions:

$$p_i^{t+1} = p_i^t + V_i^{t+1} \quad \ldots\ldots\ldots\ldots (2)$$

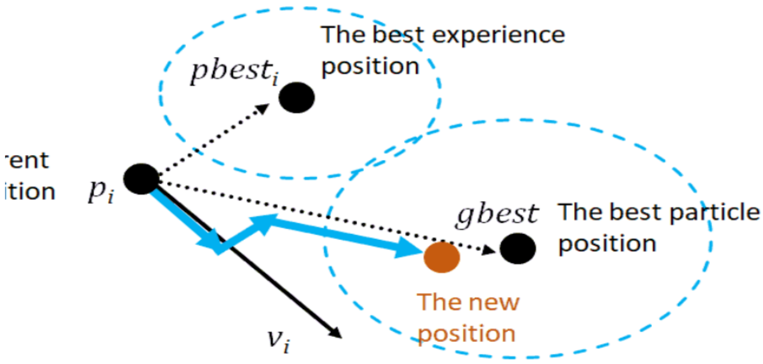

Figure 2: Particle movement in the Particle Swarm Optimization algorithm

Where c_1, c_2: acceleration constants, r_1, r_2: random numbers between (0 and 1), t: iteration number. There is co-dependence between velocity and position parameters. Fig. 2 illustrates the particle in the swarm[26], while Fig.3 shows PSO pseudocode.

Algorithm 1 PSO Algorithm Pseudocode
Input: n= particles, V_i= particle velocity, P_i= particle position, c_1= acceleration constant, c_2== acceleration constant, r_1= random number, r_2= random number, t= iteration number, P_{best} = best current position, g_{best}=globel best position, i=D-dimention vector of position **Output**: A swarm S of size n, n=*50,100,.....n* 1: **Start**: Initialize population (n) *A*: ***Repeat***: Generate solution S *randomly* *B*: ***until size of pop*** = n 2: **Repeat** *A*: ***For*** each particle, calculate fitness function *B*:*1*- **if** solution S is infeasible, then P_i = max_value 2- **else** Evaluate the fitness of the particle: P_i 3- **if** (f) > (P_{best}), Set current fitness value as P_{best} 3: **Update** 1- particle with best fitness value as (g_{best}). 2- **For** each particle calculate new V_i using eq(1), update position using eq(2). 4: **Until** (stop criterion) 1- **If** (t<maximum iterations), then t=t+1 2- **If maximum** iterations attained, **post** result 5: **end procedure**

Figure 3. Particle swarm optimization algorithm pseudocode

The flowchart of the PSO algorithm shows the steps of execution [27], as in Fig. 4.

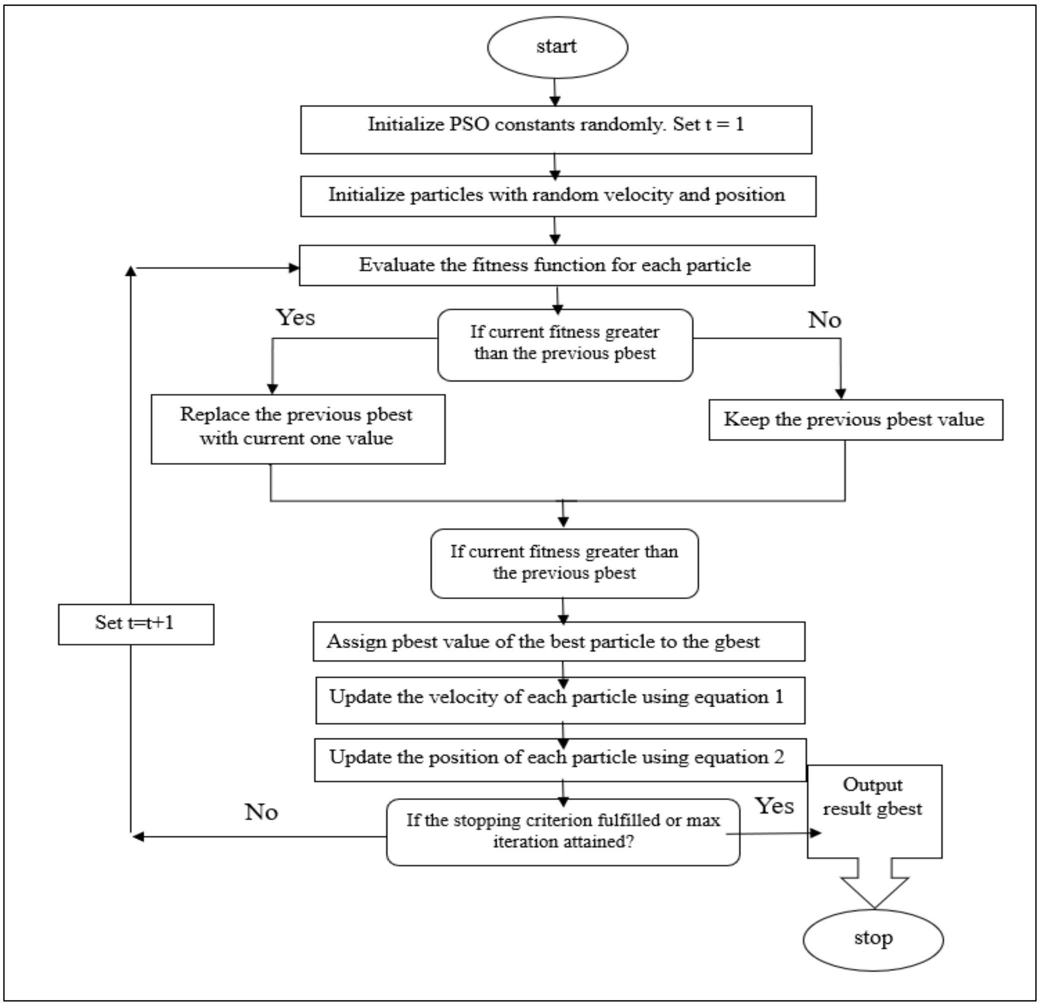

Figure 4: Particle swarm optimization flowchart

Tabu-Search algorithm (TS)

Tabu search algorithm is a robust optimization technique that leverages memory-based strategies, which balance between exploring new regions and intensifying the search around promising solutions, making it a very powerful tool. It is also considered one of the metaheuristic algorithms originally suggested by Glover[28], which is designed to solve combinatorial optimization. It is particularly useful for problems where the solution space is large and complex, making traditional optimization methods ineffective [29].

Its search algorithm starts with the neighborhood concept, which sets all possible solutions with the current position with a small change; this step is called initialization. Then a memory structure starts with the Tabu search list, which stores all previously visited solutions to prevent revisiting them again, which prevents cycles and encourages exploration in other areas. The last step is aspiration criteria, where a condition is put to compare with the results of the tabu search list, usually when it results in a better objective function value than any previously found, as shown in Figure 5.

Algorithm 2 Tabu Search Algorithm Pseudocode
1: **Procedure** Tabu Search 2: Generate initial solution S_t 3: Initialize the Tabu list 4: **While** (set of candidate solution S_t is not complete) **do** 5: Generate candidate solution S_{best} from current solution S_t 6: Add S_{best} 7: **end while** 8: Select the best solution S_{best} from S_t 9: **if** (fitness S_{best}) > (fitness S_t) 10: $S_t = S_{best}$ 11: **end if** 12: **if** (termination condition met) **then** finished, **otherwise** go to step 4 13: **end procedure**

Figure 5: Tabu search algorithm Pseudocode

Related work

In this section, which presents a forward view of the basic cloud environment system, as these researchers have proposed different techniques for enhancing task scheduling to obtain an effective service model, we review some literature associated with our study in this paper, which is provided as follows:

Authors in [30], proposed a hybrid ABC-MSA, Artificial Bee Colony Algorithm, and Modified Simulated Annealing algorithm for optimizing green cloud computing. The new hybrid algorithm has been implemented in Cloudsim toolkit for a simulation study of resource optimization and green cloud environment. The study shows the effect of using scheduling to minimize the makespan and cost, total completion time for executing tasks, and CPU utilization. The results show that the proposed hybrid ABCMSA algorithm attains reduced power consumption and response with improved energy efficiency, CPU utilization

The authors in [31], propose NDWPSO, a hybrid improved PSO algorithm with multiple strategies. They compare it with 8 nature-inspired algorithms (5 intelligent algorithms and 3 PSO variants). On 23 benchmark tests and three practical engineering problems, the simulation results proved better results, which were obtained by the NDWPSO algorithm compared with other sets of algorithms.

In [32], a hybrid PSO-DE algorithm was developed, based on feasibility rules, to improve PSO The authors verify the efficiency by applying 12 classical benchmarks and 57 optimization problems. The results show the advantages of PSO-DE compared to PSO and the differential evolution algorithm.

According to the authors in [33], a hybrid PSO algorithm is proposed, a selection method that can be used for multi-class data. The PSO-SVM algorithm was tested on a UCI dataset. The results in this study show the effectiveness of the hybrid PSO algorithm with a large number of datasets compared to the MVO-SVM method, with a better accuracy rate.
In [34], the authors introduce a new approach for the PSO algorithm for solving optimization problems. by combining the merits of free PSO (pf-PSO) with extrapolated PSO (e-PSO), the goal is to accelerate obtaining a global optimal solution.

In [35], the authors proposed a hybrid PSO with GSA (gravitational search algorithm) to improve the accuracy of the GSA algorithm while searching in a big database.
The author proposed in [36] a hyper-heuristic algorithm for a centralized cloud architecture using various existing scheduling algorithms such as FCFS, PSO, GA, and simulated annealing algorithm.

The author proposed in [37] a comparison between three algorithms for task scheduling, these algorithms are PSO and modified PSO, and the genetic algorithm, where SJFP is merged with PSO to reduce the makespan. the initial population generated as a modified PSO algorithm exceeds the two other algorithms in this comparison.

Hybrid PSO-Tabu Algorithm

In this study, A PSO-Tabu hybrid algorithm has been introduced, which combines the strengths of Particle Swarm Optimization and Tabu search to enhance the performance in solving complex optimization problems such as resource scheduling in cloud computing environments. The PSO population-based algorithm, inspired by social behavior, which consists of a number of particles that move to find the best solution in the problem space, with fast convergence at global search, but PSO can get stuck in local optimum, especially with improper parameters.

On the other hand, the TS algorithm is a local search engine that uses memory structures with tabu lists to avoid visiting already visited areas. With this frequency list, the search process is forced to cover new areas and prevent early convergence to the local optimal solution. This advantage of the TS algorithm is combined with the PSO algorithm to introduce a hybrid PSO-TS algorithm for task scheduling in a cloud computing environment, Fig. 5. Shows the flowchart for the hybrid PSO-TS algorithm.

As in eq. (1) and (2), the hybrid algorithm starts with the PSO basic algorithm, where particles update their position and velocity, and to avoid the PSO from falling into the continuous search; moreover, TS with its strong local searching, which increases the PSO accuracy, and after the swarm updates its position and velocity, the swarm goes into the TS best solution. This swarm considers the best possible solution, which, when updated with time, represents the best global optimum solution, to output the best solution as shown in Figure 6, Fig.ure7.

Algorithm 3 PSO-TS Algorithm Pseudocode

Input: n= particles, V_i= particle velocity, P_i= particle position, c_1= acceleration constant, c_2== acceleration constant, r_1= random number, r_2= random number, t= iteration number, P_{best} = best current position, g_{best}=globel best position, i=D-dimention vector of position
Output: A swarm S of size n, n=50,100,.....n
1: **Start**: Initialize population (n)
A: ***Repeat***: Generate solution *S randomly*
B: ***until size of pop** = n*
2: **Repeat**
A: ***For*** each particle, calculate fitness function
B:1- **if** solution S is infeasible, then P_i = max_value
2- **else** Evaluate the fitness of the particle: P_i
3- **if** (f) > (P_{best}), Set current fitness value as P_{best}
3: procedure Tabu Search ()
Define a neighborhood function N and select a local search algorithm
local Search () for N;
t := 0;
S_t := Generate a starting solution;
$S_{best} := S_t$;
$H_t := (S_t, S_{best}, t)$; // initialize the memory data structure
4: **while** (¬ termination criteria(S_t, H_t))
t := t + 1;
Depending on (S_t, H_t), generate a subset $N_t^H(S_t) \subseteq N(S_t)$;
S_t := Best $\{s \mid s \in N_t^H(S_t)\}$ with respect to ʃ or some $ʃ_t^H = F(ʃ, S_t, H_t)$;
if ($ʃ(S_t) < ʃ(S_{best})$)
$S_{best} := S_t$;
Update the memory data structure H_t;
end while
return S_{best};
5: **Until** (stop criterion)
1- **If** (t<maximum iterations), then t=t+1
2- **If maximum** iterations attained, **post** result
6: **End procedure**

Figure 6: hybrid PSO-TS algorithm Pseudocode

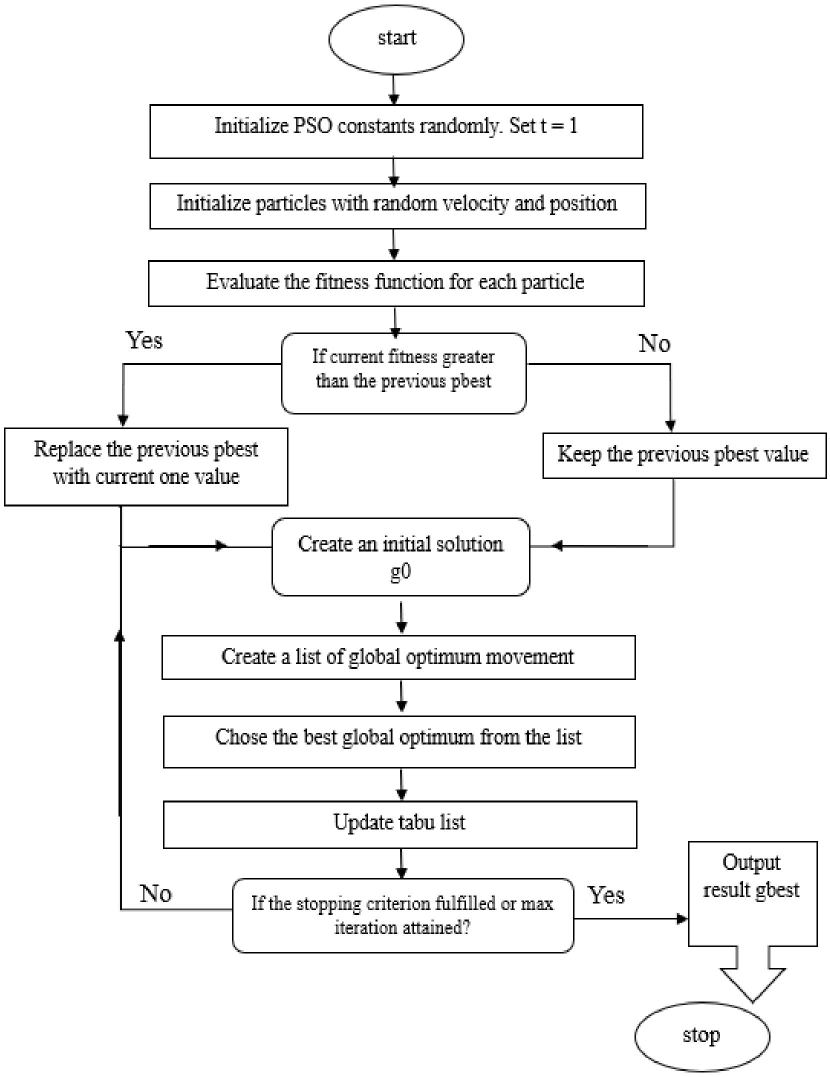

Figure 7: Flowchart for hybrid PSO-TS algorithm

7- Problem statement

Medical computing systems, with their growth towards more dependence on cloud services such as managing, storing, and digital libraries. Shared database medical records, these clinical decision support systems (CDSS) need more cloud-powered repositories to overcome the limitations and improve the cloud services quality (QoS), one of the first solution that implement in modern operation cloud system is cloud scheduling algorithms which adopted hybrid scheduling techniques which improve the efficiency, reduce starvation, and gives better CPU utilization to optimize the resources and enhanced system performances. In this work hybrid PSO-TS algorithm has been implemented and compared with traditional cloud scheduling algorithms such as FCFS, SJF, RR, and the PSO algorithm. Goal is to minimize the makespan, as well as minimize the total completion time, total cost, and energy efficiency.

8- Experimental part

The hybrid PSO-TS algorithm was implemented using CloudSim 3.0.3 tool-kit simulator, which has been chosen for simulating different scenarios as it is an open-source framework that provides different cloud environment modes. These components allow developers to evaluate new strategies in utilization of clouds (balancing, scheduling, policies, and mapping, etc.).

The IDE used in this study is Eclipse 4.31.0 with JDK 17. Computer specification: 4th generation Intel Core i7 Processor, MSI 70 2P Apache Pro, with DDR3 memory (16GB).
CloudSim is used for efficiency evaluation from different perspectives, including cost, speed, and time for different applications, also, it supports of evaluation of green policies. There are two policies available in the base CloudSim simulator toolkit: the default policies, the first VM space-shared scheduler, which means that each VM is assigned to one CPU core. When these VMs complete the process, the CPU is assigned to another VM. That means by this policy, in time, there is one process executed in VM, with no context switch, and tasks execution continues to finish. The second policy, VM Time-shared scheduler, arranges all jobs that are assigned to VMs with the same time; the result, context switch happens within tasks while execution [38].

The CloudSim structure consists of a number of components that simulate different parts of a cloud computing environment. These components allow researchers to model and evaluate the performance of cloud resource allocation, scheduling, and management strategies.
The main components in CloudSim are shown in Fig. 8:

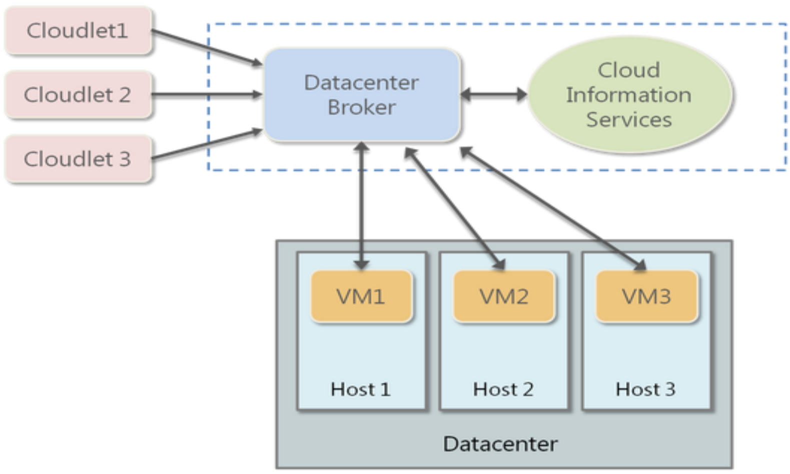

Figure 8: Cloudsim toolkit cloud Components

• Data Center: consists of a group of hosts that represent the physical infrastructure, each host consists of one or more VMs that are responsible for processing the requests or tasks.
• Datacenter Broker: one of the most important components in CloudSim, which is a class that functions as policies that can be expanded to accommodate user experiments.
• Host: It manages virtual machines (VMs); it supplies them with memory, bandwidth, and processor components; it is connected to the datacenter.
• Virtual machines, or VMs, are software that runs programs and functions similarly to a real system. Each VM allocates host resources across activities based on a policy that has been put in place.
• Cloudlet: the datacenter broker class implements scheduling policies that govern this class, which is called a task [32].
The following is the project's experimental setup:

- The specifications for Cloudlet are: long output size = 300, long file size = 300.
- VMs number = 20 homogeneous.
- Time policy: sharing policy among the host and task levels.

- PSO population = 25 particles, PSO iteration = 500.
- System: Linux, Long image size = 10000, name: Xen, process speed = 1000 MIPS.
- VM memory (RAM) = 1000000 MB.
- Band width = 200000 MB.

Several task scheduling algorithms, including FCFS, SJF, RR, and PSO, are implemented over the CloudSim in this research for comparison with the suggested hybrid PSO-TS.

- **FCFS (First come, First served)**: Its non-preemptive, where tasks are completed in the order in which they received, this means, the scheduler in datacenter broker selects tasks that arrive first and run it to completion, this straightforward approach and simple implemented, make this algorithm used as a default scheduling method when no priorities are specified. The advantages of using FCFS in cloud scheduling are simplicity and fairness, as all tasks are executed in the order they arrive. While unlikely in FCFS, if there are constant arrivals of large jobs, smaller jobs could be delayed for long periods, leading to inefficiency in overall scheduling
- **SJF (Short Job First)**: may be either non-preemptive or preemptive; In the SJF, the tasks are ascending in order of their burst time, the scheduler allocates the resources depending on the shortest task execution time. This type of scheduler has advantages in both preemptive and non-preemptive versions. The average time compared between FCFS and SJF shows a reduction in SJF type, as this algorithm optimizes the turnaround time to a considerable degree, and the execution time of each process is estimated.
- **RR (Round Robin)**: a scheduling preemptive algorithm, the scheduler allocated the resources to execute the tasks with fixed slice time in rotation till all tasks finished [39].
- **PSO** (**Particle Swarm Optimization**: a popular meta-heuristic optimization technique, basically, a swarm-based algorithm that exploits social behavior like birds flocking, is particularly effective for solving discrete and continuous optimization problems.

8- Results and dissection

In comparison to other standard scheduling, the suggested hybrid PSO-TS scheduling algorithm effectively reduced the scheduling workflow time and cost. To confirm the hybrid PSO-TS algorithm's superiority in terms of time and cost, the algorithms were tested in the following scenario, each with 200 tasks:

1- **Makespan**: The most important and commonly used performance indicator to find the best schedule that can minimize the maximum completion time to execute the tasks. The results from Table 1, and Fig 10 confirm the effect of using a hybrid PSO-TS scheduler to minimize the total makespan.

Table 1: Makespan comparison for executing 200 tasks using different algorithms

FCFS	SJF	RR	PSO	PSO-Tabu
9703.943115	9204.448336	13171.15319	9113.113926	7022.566703

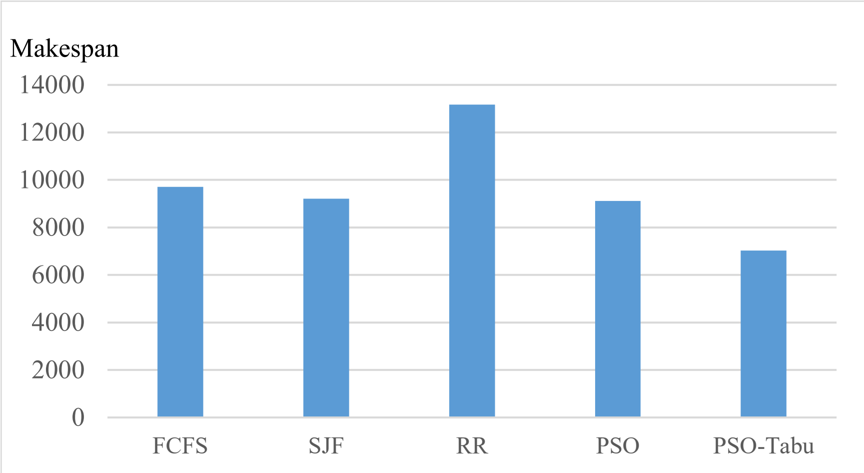

Figure 10: Makespan sketched of 200 tasks for different algorithms

2- **Total Completion Time (TCT)**: refers to the sum of all completion times for all tasks. The results, as shown in Table 2 and Fig. 11, confirm the role of the hybrid PSO-TS algorithm for minimizing total completion time.

Table 2: TCT comparison for executing 200 tasks using different algorithms

FCFS	SJF	RR	PSO	PSO-Tabu
678239.353	658374.065	701801.411	562575.296	457481.558

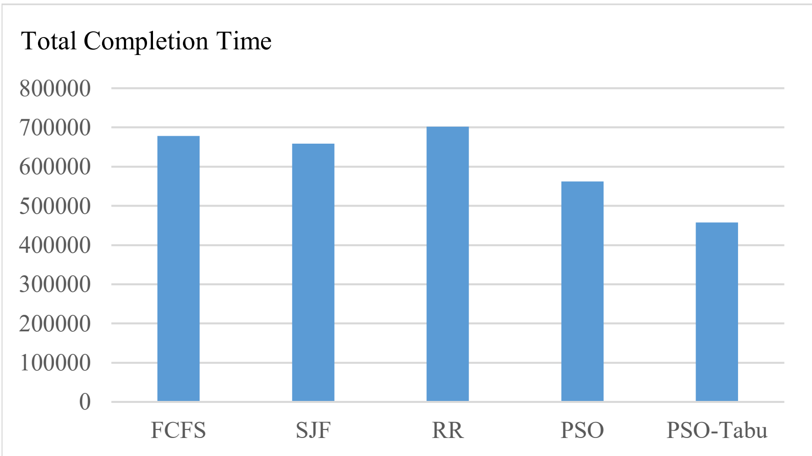

Figure 11: Total completion time comparison of 200 tasks

3- **Average Waiting Time (AWT)**: refers to the summation of all waiting time tasks divided by the number of available resources, as shown in Table 3, and Fig. 12 shows the results, which confirm that the lowest AWT is achieved with hybrid PSO-TS.

Table 3: AWT results using different scheduling algorithms

FCFS	SJF	RR	PSO	PSO-Tabu

2814.64628	2734.907755	2946.397905	2469.98198	2268.5996

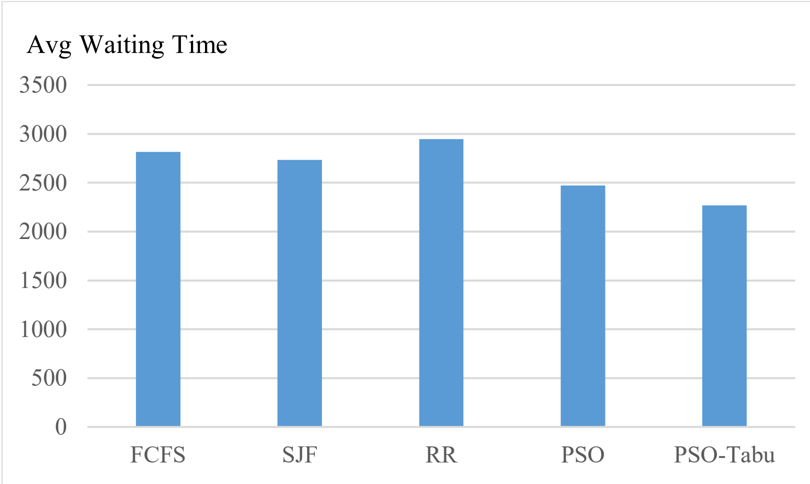

Figure 12: Average waiting time comparison sketched of 200 tasks for different algorithms

4- **Total cost**: The implemented data used for simulating the cost is:
Time Zone = 10.0
One CPU cost in the process = 3.0, Memory cost in the process = 0.05
Storge cost in the process = 0.1, Bandwidth cost in the process = 0.1
It is clear from Table 4 and Fig. 13, that the PSO-TS achieved the lowest value in cost for 200 tasks executions.

Table 4: Total cost comparison for executed 200 tasks

FCFS	SJF	RR	PSO	PSO-Tabu
345930.291	334177.542	337565.49	261192.519	226813.4928

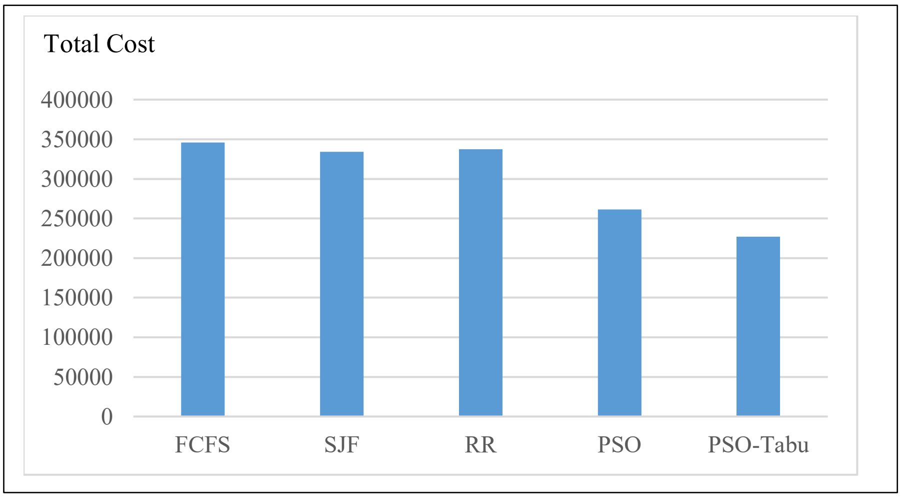

Figure 13: Total cost comparison sketched for 200 tasks for different algorithms

5- **Energy efficiency**:

The proposed hybrid PSO-Tabu algorithm consumes less energy per 200 tasks and keeps CPU usage high, all while reducing total completion time, making it ideal for green data centers. The Tabu-enhanced PSO avoided unnecessary resource idling and overloading, contributing to a more balanced energy draw. Table 5 and Fig. 14 show the total energy comparison for executing 200 tasks using different algorithms.

Table 5: Total energy consumed for executing 200 tasks

FCFS	SJF	RR	PSO	PSO-TS
552.99	619.54	582.64	474.29	433.19

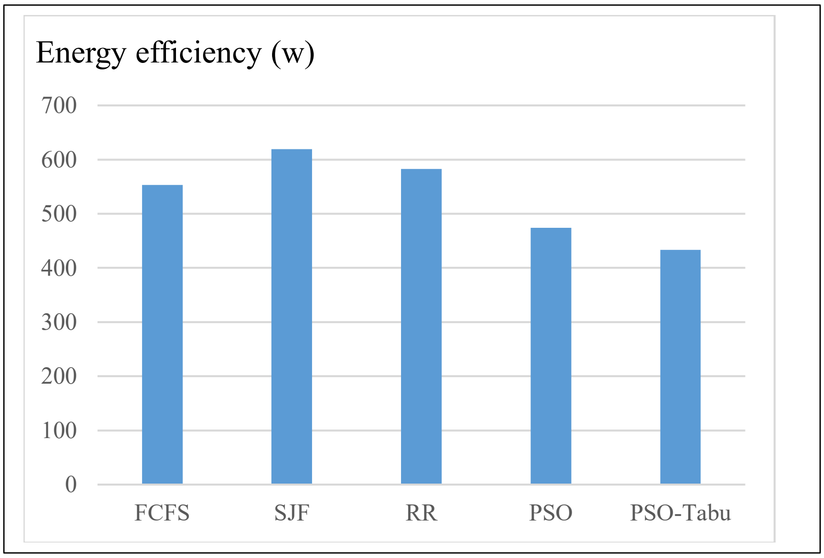

Figure 14: Energy efficiency comparison sketched of 200 tasks for different algorithms

Conclusion

These days, cloud computing systems must provide high-quality services; minimizing the amount of time needed to complete tasks on interactive cloud systems, which is crucial by lowering the makespan, cost, and average completion time for scheduling algorithms. The hybrid PSO-TS algorithms in this study demonstrate their function in minimizing makespan when compared to other algorithms, as well as minimizing overall cost and energy efficiency for completing 200 tasks.

Hybrid PSO-TS algorithm combine the global search ability of PSO with the local refinement ability of Tabu Search (TS), to enhance task scheduling performance in cloud computing, to overcome the falling into local optimum in PSO algorithm, this study of hybrid PSO-TS algorithm with its strong local searching that implemented from TS algorithm, which increases the PSO accuracy.

References

[1] M. Zeedan, G. Attiya, and N. El-Fishawy, "Enhanced hybrid multi-objective workflow scheduling approach based on artificial bee colony in cloud computing," *Computing*, vol. 105, no. 1, pp. 217–247, Jan. 2023,.

[2] A. Salim Ali Abbas, "Improving Data Storage Security in Cloud Computing Using Elliptic Curve Cryptography," vol. 17, no. 4, pp. 48–53, 2015.

[3] C. A. Mayfield *et al.*, "Using cloud-based, open-source technology to evaluate, improve, and rapidly disseminate community-based intervention data," *Journal of the American Medical Informatics Association*, vol. 27, no. 11, pp. 1741–1746, Nov. 2020.

[4] M. Taha and Khalid Ali Hussein, "Design Sequential Hybrid Lightweight Algorithms For Encryption Data Using RC6 Algorithm," *Mustansiriyah Journal of Pure and Applied Sciences*, vol. 3, no. 1, pp. 1–11, Jan. 2025.

[5] S. Husam, Haider Kadhim Hoomod, and Khalid Ali Hussein, "Hybrid Algorithm (DES-Present) for encryption image color using 2D-Chaotic System," *Mustansiriyah Journal of Pure and Applied Sciences*, vol. 3, no. 1, pp. 49–63, Jan. 2025.

[6] B. Wu, C. Wang, and H. Yao, "Security analysis and secure channel-free certificateless searchable public key authenticated encryption for a cloud-based Internet of things," *PLoS One*, vol. 15, no. 4, 2020.

[7] S. Ali and R. Alubady, "A Survey of Fog Computing-Based Resource Allocation Approaches: Overview, Classification, and Opportunity for Research," *Iraqi Journal of Science*, vol. 65, no. 7, pp. 4008–4029, 2024.

[8] W. A. Almahdi, H. A. Lafta, and Y. H. Ali, "Intelligent Task Scheduling Using Bat and Harmony Optimization," *Iraqi Journal of Science*, vol. 64, no. 8, pp. 4187–4197, 2023.

[9] E. I. Abdul kareem and S. A. Hussein, "Optimal CPU Jobs Scheduling Method Based on Simulated Annealing Algorithm," *Iraqi Journal of Science*, vol. 64, no. 8, pp. 3640–3651, 2022.

[10] M. I. Malik, "CLOUD COMPUTING-TECHNOLOGIES," *International Journal of Advanced Research in Computer Science*, vol. 9, no. 2, pp. 379–384, Apr. 2018.

[11] M. Zeedan, G. Attiya, and N. El-Fishawy, "Enhanced hybrid multi-objective workflow scheduling approach based artificial bee colony in cloud computing," *Computing*, vol. 105, no. 1, pp. 217–247, Jan. 2023.

[12] O. Hajjar, E. Mekhallalati, N. Annwty, F. Alghayadh, I. Keshta, and M. Algabri, "Performance Assessment of CPU Scheduling Algorithms: A Scenario-Based Approach with FCFS, RR, and SJF," *Journal of Computer Science*, vol. 20, no. 9, pp. 972–985, 2024.

[13] I. M. I. Et. al., "Task Scheduling Algorithms in Cloud Computing: A Review," *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*," vol. 12, no. 4, pp. 1041–1053, Apr. 2021.

[14] S. M. Khudaier and B. A. Mahmood, "A Review of Assured Data Deletion Security Techniques in Cloud Storage," *Iraqi Journal of Science*, vol. 64, no. 5, pp. 2492–2511, 2023.

[15] A. A. Maryoosh, R. S. Mohammed, and R. A. Mustafa, "Subject Review: Cloud Computing Security Based on Cryptography," *International Journal of Engineering Research and Advanced Technology*, vol. 05, no. 09, pp. 20–23, 2019.

[16] S. Haboubi and O. Ben Salem, "Energy Consumption Prediction of Smart Buildings by Using Machine Learning Techniques," *Iraqi Journal of Science*, vol. 64, no. 12, pp. 6509–6521, 2023.

[17] M. F. Manzoor, A. Abid, M. S. Farooq, N. A. Nawaz, and U. Farooq, "Resource allocation techniques in cloud computing: A review and future directions," *Kauno Technologists Universitates* Dec. 01, 2020.

[18] H. A. Mahmood, "Network Intrusion Detection System (NIDS) in Cloud Environment based on Hidden Naïve Bayes Multiclass Classifier," *Al-Mustansiriyah Journal of Science*, vol. 28, no. 2, pp. 134–142, Apr. 2018.

[19] B. Zakaria, K. Abouelmehdi, A. Beni-Hssane, and H. Khaloufi, "NEW HYBRID ALGORITHM FOR TASK SCHEDULING IN CLOUD COMPUTING," *J Theor Appl Inf Technol*, vol. 31, p. 24, 2021.

[20] N. V. Janagoudar, D. G. Narayan, and M. M. Mulla, "Multi-Objective Scheduling using Logistic Regression for OpenStack-based Cloud," in *Procedia Computer Science*, Elsevier B.V., vol. 4, pp. 1429–1438, 2020.

[21] M. Abdulredha, B. Attea, and A. Jabir, "Heuristic and Meta-Heuristic Optimization Models for Task Scheduling in Cloud-Fog Systems: A Review," *Iraqi Journal for Electrical and Electronic Engineering*, vol. 16, no. 2, pp. 103–112, Dec. 2020.

[22] N. A. Al-Thanoon, O. S. Qasim, and Z. Y. Algamal, "A new hybrid firefly algorithm and particle swarm optimization for tuning parameter estimation in penalized support vector machine with application in chemometrics," *Chemometrics and Intelligent Laboratory Systems*, vol. 184, pp. 142–152, Jan. 2019.

[23] W. Deng, R. Yao, H. Zhao, X. Yang, and G. Li, "A novel intelligent diagnosis method using optimal LS-SVM with improved PSO algorithm," *Soft comput*, vol. 23, no. 7, pp. 2445–2462, Apr. 2019.

[24] E. H. Houssein, A. G. Gad, Y. M. Wazery, and P. N. Suganthan, "Task Scheduling in Cloud Computing based on Meta-heuristics: Review, Taxonomy, Open Challenges, and Future Trends," *Swarm Evol Comput*, vol. 62, Apr. 2021.

[25] H. Huang, L. Lv, S. Ye, and Z. Hao, "Particle swarm optimization with convergence speed controller for large-scale numerical optimization," *Soft comput*, vol. 23, no. 12, pp. 4421–4437, Jun. 2019.

[26] S. Chen, J. qiang Wang, and H. yu Zhang, "A hybrid PSO-SVM model based on clustering algorithm for short-term atmospheric pollutant concentration forecasting," *Technol Forecast Soc Change*, vol. 146, pp. 41–54, Sep. 2019.

[27] A. G. Gad, "Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review," *Archives of Computational Methods in Engineering*, vol. 29, no. 5, pp. 2531–2561, Aug. 2022.

[28] F. Glover, " Tabu Search and Adaptive Memory Programming — Advances, Applications and Challenges," computer science interface, vol. 7, no.8, pp- 4615-4102, 1997.

[29] H. Alazzam, E. Alhenawi, and R. Al-Sayyed, "A hybrid job scheduling algorithm based on Tabu and Harmony search algorithms," *Journal of Supercomputing*, vol. 75, no. 12, pp. 7994–8011, Dec. 2019.

[30] T. A. Essa and K. Q. Hussain, "A Hybrid ABC-MSA Algorithm for Optimizing Green Cloud Computing Solutions", 1st International Conference on Applications of Future Technologies for Sustainable Education- Al-Mastaqbal University, vol. 1, pp. 286-300, 2025.

[31] J. Qiao *et al.*, "A hybrid particle swarm optimization algorithm for solving engineering problem," *Sci Rep*, vol. 14, no. 1, Dec. 2024.

[32] E. Guo, Y. Gao, C. Hu, and J. Zhang, "A Hybrid PSO-DE Intelligent Algorithm for Solving Constrained Optimization Problems Based on Feasibility Rules," *Mathematics*, vol. 11, no. 3, Feb. 2023.

[33] J. Wang, X. Wang, X. Li, and J. Yi, "A Hybrid Particle Swarm Optimization Algorithm with Dynamic Adjustment of Inertia Weight Based on a New Feature Selection Method to Optimize SVM Parameters," *Entropy*, vol. 25, no. 3, Mar. 2023.

[34] G. Ramana Murthy, M. Senthil Arumugam, and C. K. Loo, "Hybrid particle swarm optimization algorithm with fine tuning operators," *International Journal of Bio-Inspired Computation*, vol. 1, no. 1–2, pp. 14–31, 2009.

[35] M. Hosseini and M. S. Navabi, "Hybrid PSO-GSA based approach for feature selection," JIEMS, vol.10, no1, pp 1-15,2023.

[36] S. Kabirzadeh, D. Rahbari, and M. Nickray, "A Hyper Heuristic Algorithm for Scheduling of FogNetworks." *Conference of Open Innovations Association (FRUCT)*, pp. 148-155, 2017.

[37] S. Potluri, A. A. Hamad, D. Godavarthi, and S. S. Basa, "Enhanced Task Scheduling Using Optimized Particle Swarm Optimization Algorithm in Cloud Computing Environment," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 11, no. 3, pp. 1–5, 2024.

[38] M. A. Shahid, M. M. Alam, and M. M. Su'ud, "A Systematic Parameter Analysis of Cloud Simulation Tools in Cloud Computing Environments," *Applied Sciences (Switzerland)*, vol. 13, no. 15, Aug. 2023.

[39] Nidhi, M. Nagle, and V. Nagar, "Cloud Computing: Optimization using Particle Swarm Optimization to Improve Performance of Cloud," *EAI Endorsed Transactions on Internet of Things*, vol. 10, 2024.