Improving Quality of Service Based on Tasks Scheduling Approach in Cloud Computing

AbdulHakim Hekmat*, Vahid Ranjbar **, Mahdi Aqa Sarram***

DOI: 10.29322/IJSRP.16.11.2026.p17922
https://dx.doi.org/10.29322/IJSRP.16.11.2026.p17922

Paper Received Date: 23rd October 2026
Paper Acceptance Date: 22nd November 2026
Paper Publication Date: 27th November 2026

Abstract- Cloud computing is one of the most acceptable emerging technologies, which involves the allocation and de-allocation of the computing resources using the Internet as the core technology to compute the tasks or jobs submitted by the users. Task scheduling is a major problem in cloud computing that reduces system performance. An efficient scheduling algorithm is needed to improve system performance. It is believed that a proper scheduling method can have a significant impact on reducing costs, increasing resource productivity and efficiency of service quality parameters. In this paper, a particle swarm optimization (PSO) mechanism is proposed to improve service quality and efficient scheduling of tasks among virtual machines (VMs). The proposed algorithm using the CloudSim simulator is compared with genetic and greedy algorithms. The simulation results clearly show that the PSO-based task planning mechanism clearly works better than the genetic and greedy algorithms. because it creates approximately 20% cost reduction and 20% increase in resource utilization.

key words: IJSRP Research Journal

I. INTRODUCTION

As the demand and benefits of cloud computing infrastructure increase, so does the movement speed is increasing toward the cloud.Tasks need to be optimally timed to virtual machines to increase responsiveness and ensure cloud user satisfaction. Task scheduling is one of the key issues in cloud computing and many efforts have been made to solve this problem. Cloud computing is an approach, development and application of Internet-based computer technologies that provides IT-related capabilities as a service or service to the user and allow users to access Internet-based services. Without having specialized information about this technology [1].

The cloud computing environment has a large volume of different types of resources that provide different services or services based on user requests. Increasing user demand leads to increased load on the cloud. Many technologies such as optimized hardware, optimized network, optimized databases and other related technologies can be used for load management. One of them is scheduling tasks performed by users to virtual machines in a data center. Cloud computing is a service-based structure that is actually a set of services that communicate with each other. These services are categorized into system architecture layers. In the general and most comprehensive type of architecture, cloud computing consists of three main layers. Software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS). [2].

Task scheduling is a vital step in enhancing cloud computing performance. Although many scheduling algorithms are used in the cloud, sensible and efficient algorithms have not been used yet [3]. Therefore, an approach is proposed that has a better performance than previous algorithms according to the simulation results. This study examines how using cloud scheduling in cloud computing can ensure service quality and reduce costs. The application and importance of this research will be specifically in scheduling tasks as well as improving the quality of services for users in the cloud computing environment.

The organization of content in this Paper is as follows: in the second part an overview of research conducted in this field, in the third part the proposed method to schedule and improve service quality, in the fourth part a description of the quasi-environment and evaluation of the proposed algorithm and finally Conclusion The article will be reviewed.

This publication is licensed under Creative Commons Attribution CC BY.

II. RELATED WORKS

Significant research has been done on scheduling tasks in cloud computing using evolutionary algorithms, each of which will be examined in more detail below.

Hamad & Omra [4] has proposed an approach to scheduling tasks based on the Genetic Algorithm (GA) to perform tasks, the purpose of which is to minimize the completion time and cost of tasks and maximize the use of resources. Fatemeh Rastkhadio and Kamran Zamanifar [5] propose a new cloud scheduling approach based on load balancing using the artificial bee algorithm, one of the most popular bio-inspiring methods. The artificial bee cloning algorithm is an optimized metaphorical algorithm based on a specific intelligent bee swarm behavior. The schedule is designed to achieve a balanced load between virtual machines to maximize power and deliver maximum power. The simulated experimental results show its effectiveness in optimizing load balance and work scheduling compared to FCFS and ACO. The proposed approach reduces the time required to complete the work, and the work is done faster. Their approach increases efficiency and the degree of imbalance, distributes tasks, and balances loads between virtual devices. As a result, the proposed algorithm increases the efficiency and productivity of resources.

Nidhi Bansal et al. [6] introduce an approach that optimizes task scheduling based on the priority of virtual devices. This algorithm does good results for load balancing but is not cost-effective.

Octane Rani and Harperit Kavir [7] proposed the Raven Roosting (RRO) optimization algorithm to solve task scheduling problems in the cloud. In Raven Roosting optimization algorithm, hens are information centers that are inspired to solve problems. This technique is good enough to perform a certain number of overload tasks in virtual machines (VMs) by determining the capacity of VMs. Randomly assigning Raven Roosting Optimization (RRO) from virtual machines to Cloudlets will lead to a huge change in the structure of the VM to which it is assigned. The simulation results are likely to show better compatibility, average response time, average waiting time, and number of tasks migrated through the Raven Roosting optimization algorithm.

Ben Sal et al. [8] has proposed a scheduling algorithm based on the priority of virtual machines to balance the load and reduce costs. As well as increasing resource usage and reducing costs, both are necessary factors

for scheduling algorithms in cloud computing. In this paper, the authors have tested the proposed algorithm by considering the load balancing and cost parameters, and have shown that the proposed algorithm performed well for load balancing but was not effective in reducing costs.

A summary of previous research can also be found in Table 1.

Table 1: Summary of research history

Proposed Algorithm	Resources	Year of publication	Scheduling category			quality of service (QoS)			Description of the algorithms used	Performance of the proposed algorithms
			static	dynamic						
				Online	Offline	Cost	Makespan Time	performance		
Lyapunov optimization	[9]	2016		✓			✓		An approach to planning has been proposed. They use Lyapunov optimization to develop an online distributed approach that can minimize the average time and reduce the number of queues in the system significantly.	This approach can significantly reduce the average time as well as the number of queues in the system.

This publication is licensed under Creative Commons Attribution CC BY.

Min-Min Algorithm	[14]	2013	✓				✓	✓	In this approach, a work scheduling algorithm for network computing is introduced. The proposed algorithm is based on a general adaptive planning exploration discovery, which includes improving service quality.	These task scheduling algorithms have been used in cloud computing to improve service quality.
DTSWB model	[10]	2018			✓		✓		The Task Scheduling Algorithm is based on the proposed model (DTSWB) which can consider both tasks and providers. And the DTSWB model performs better than other existing programming algorithms.	The DTSWB model performs better than other algorithms. These models are used to improve runtime in scheduling tasks.
Genetic Algorithm (GA)	[11]	2016			✓	✓	✓	✓	Provides an approach to scheduling tasks based on the Genetic Algorithm (GA) for assigning and executing program tasks, the purpose of which is to minimize completion time and cost of tasks and maximize the use of resources.	The purpose of this proposed algorithm is to minimize the time and cost of work and maximize the use of resources.
Artificial bee algorithm	[5]	2016			✓			✓	They propose an approach for cloud scheduling based on load balance, using an artificial bee algorithm. The artificial bee cloning algorithm is an optimized metaphorical algorithm based on a specific intelligent bee swarm behavior.	The proposed approach increases efficiency and balances the load between virtual devices. It also increases resource efficiency and productivity.
FCFS Algorithm	[13]	2016	✓			✓			Introduces an approach that optimizes task scheduling based on the priority of virtual devices. This algorithm does good results for load balancing but is not cost effective for performance.	The proposed approach works well for load balancing, but is not cost effective for performance.
Raven Roosting (RRO) optimization algorithm	[6]	2017			✓		✓		The Raven Roosting (RRO) optimization algorithm is proposed to lighten the load balance and also to solve the tasks of scheduling tasks in the cloud environment. In Raven Roosting optimization algorithm, hens are information centers or it can be said that servers and features are ordinary ravens that are inspired to solve problems.	The simulation results show better compatibility, mean response time, mean waiting time, and number of tasks migrated through the Raven Roosting optimization algorithm.
Max-Min Algorithm	[12]	2019	✓				✓		This algorithm can solve the problem of execution time, similar to the least execution time-least scheduling algorithm, except that among the unscheduled tasks, the work with the longest execution time is assigned to a source that has the least completion time. .	It performs better than other scheduling algorithms in terms of runtime and resource usage. The main advantage of this algorithm is the simplicity of implementation.
Algorithm (CSA)	[16]	2020		✓				✓	In this paper, Crow Search Algorithm (CSA) is proposed for scheduling tasks in cloud computing. This algorithm is inspired by crow eating habits. In fact, the crow looks to his other mates to find a better food source than the current food source. In this way, CSA finds a suitable virtual machine for this task and minimizes the speed of construction.	CSA finds a suitable virtual machine for this task and minimizes the speed of construction.

This publication is licensed under Creative Commons Attribution CC BY.

III. SUGGESTED METHOD

In this section, our proposed approach consists of three sections, the Implementation model, proposed mathematical model and the proposed algorithm, each of which we discuss in detail.

- **IMPLEMENTATION MODEL**

Figure (1) shows the implementation model in the proposed method. In the proposed model, there is a discussion about the complete cycle of resource management and optimization of tasks scheduling in the cloud environment. Because fluctuations in resource productivity cause many problems in the cloud environment, end users and service recipients, each as an important entity in the cloud, strive to make a sensible decision that can meet the quality of service required to meet the needs of users. Present. People who have a valuable role in the cloud environment are: end users, application service providers (ASP) and cloud service providers (CSP).

In cloud computing, end users are people who end up using cloud services but do not support the service. In the proposed model, the end-user entity must generate user requests and send them to ASP at random intervals. ASP receives user requests and executes them on virtual machines according to the existing scheduling policy and sends the received result to the users. Therefore, in the proposed model, ASP implementation requires the development of a Broker in CloudSim. Cloud Service Providers (CSPs) are companies that provide cloud computing services to end users. In the proposed model, in order to establish the CSP layer, we set up 2 data centers, each containing 5 hosts.

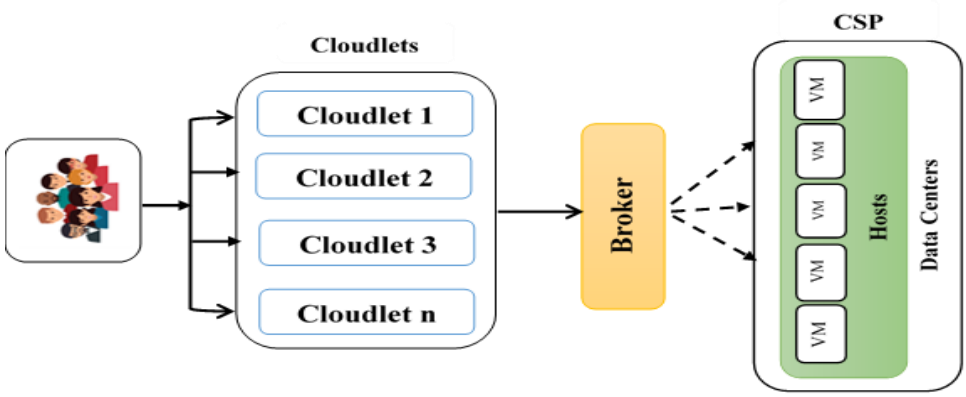


Figure 1: Implementation model

- **FORMULATION OF THE PROBLEM**

Given the defined issues, our goal is to improve service quality performance in cloud computing based on task scheduling methods. To do this, we need to consider various QoS metrics such as runtime, which by minimizing this parameter can also minimize the completion time and cost associated with the VM, while reducing these parameters to increase resource efficiency and improve system performance. To minimize these parameters, we will define the fit function and formulate and calculate each of these parameters mathematically. Consider that the number of n requests sent in the set T_i = {T1, T2, T3… Tn} by users, each of which is independent of each other, each task has $L(Ti)$ specified length. The purpose of this Paper is to reduce the execution time to perform all tasks and improve the quality of service. Given that in the particle swarm optimization algorithm, the efficiency of each particle is based on its fitness function, we defined the fitness function as equation (1) with the aim of reducing the makespan time of all tasks and increasing productivity.

$$F = \theta TotalCost + (1-\theta)MakespanTime \quad (1)$$

According to Equation (1), here α is the weight given to the total cost and the $(\alpha - 1)$ weight given to the total processing time, $0 \le \alpha \le 1$.

One of the goals of task scheduling is to try to minimize the execution time of tasks in virtual machines. Execution time means the amount of time spent by the system to perform each task until the end. Task execution time T_i can be calculated according to formula (2).

$$ET_{ij} = \sum_{i=1}^{n} X_{ij} + C_{ij} \quad (2)$$

This publication is licensed under Creative Commons Attribution CC BY.

So that X_{ij} the finish time is on virtual machines and C_{ij} the sum of input and output requests. C_{ij} Can be obtained from Equation (3).

$$C_{ij} = \frac{(TaskIsize + TaskOsize)}{Bandwidth} \tag{3}$$

In this case, it is an attempt to minimize the total execution time of all tasks in the virtual machine. The execution time T_i of all tasks can be calculated according to formula (4).

$$TET_{ij} = \sum_{i=1}^{n} ET_{ij} * \alpha_{ij} \tag{4}$$

ET_{ij} Indicates the execution time and α_{ij} indicates that if $\alpha = 1$ in this case the tasks are mapped to a virtual machine, otherwise it will be $\alpha = 0$. Also, the makespan time, which is the maximum time required to perform all input tasks in a system, can be calculated according to formula (5) [15].

$$Makespan = \max\left\{\sum_{i=1}^{n} TET_i\right\} \tag{5}$$

Resource productivity, the capacity that is used during implementation, is known as the percentage of resource productivity. The resource productivity percentage is calculated according to formula (6) [15].

$$U_j = \sum_{j=1}^{m} \frac{ET_{ij}}{MakespanTime} \tag{6}$$

One of the parameters that improves service quality is cost minimization, which can be used to calculate the cost of performing tasks performed on virtual machines according to Equation (7).

$$VmExeCost = (VmPrice) * (VmHours) * (ET_{ij}) \tag{7}$$

Here is $VmPrice$ the amount of initial cost determined by the service provider for each VM, $VmHours$ the length of time that Vm is available to the user. ET_{ij} How long work is being done on VMs. Also, the total execution cost can be obtained according to Equation (8).

$$TotalExeCost = \sum_{i=1}^{n} VmExe\,\mathrm{Cost}_{ij} * \alpha_{ij} \tag{8}$$

$VmExe\,\mathrm{Cost}_{ij}$ Indicates the execution cost for each Vm and α_{ij} indicates that if $\alpha = 1$ in this case the tasks are mapped to a virtual machine, otherwise $\alpha = 0$ it will be. According to Equations (4) and (8), the total cost can be calculated according to Equation (9).

$$Total\ \mathrm{Cost} = \beta Max \sum_{i=1}^{n} TET_{ij} + \delta \sum_{i=1}^{n} T\ \mathrm{cost}_{ij} \tag{9}$$

Here β and δ represents the weights assigned to FT_{ij} and FC_{ij} that is the sum β and δ can be equal to one ($\beta + \delta = 1$).

Equation (10) shows that the Performance improvement rate (PIR %) of our proposed algorithm is improved by several percent compared to the genetic algorithm and the greedy algorithm.

This publication is licensed under Creative Commons Attribution CC BY.

$$PIR = \left(\frac{Makespan(GA) - Makespan(PSO)}{Makespan(GA)} \right) * 100 \quad (10)$$

Description of the proposed algorithm

The main idea of the particle swarm optimization algorithm was first proposed by Kennedy and Eberhard in 1995. The PSO algorithm is a nature-based evolutionary calculation algorithm based on iteration. PSO is one of the optimization algorithms based on random generation of the initial population. In this algorithm, it is constructed by modeling and simulating the group flight behavior of birds or the group movement of fish. Each member in this group is defined by the velocity vector and the position vector in the search space. In each time iteration, the new position of the particles is defined according to the velocity vector and the position vector in the search space [17].

$$V_i(t) = w * V_i\,(t-1) + c_1 * rand_1 * (P_{i\,.best} - X_i\,(t-1)\,) + c_2 * rand_2 * (P_{g\,.best} - X_i\,(t-1)\,)$$

$$X_i = X_i\,(t-1) + V_i\,(t)$$

W: Weight factor of inertia (moving in its own way) which indicates the effect of the previous vector velocity vector ($V_i(t)$) on the velocity vector in the current iteration ($V_i(t+1)$).

C1: Fixed training coefficient (moving in the direction of the best particle value)

C2: Constant training coefficient (moving in the direction of the best particle found among the whole population)

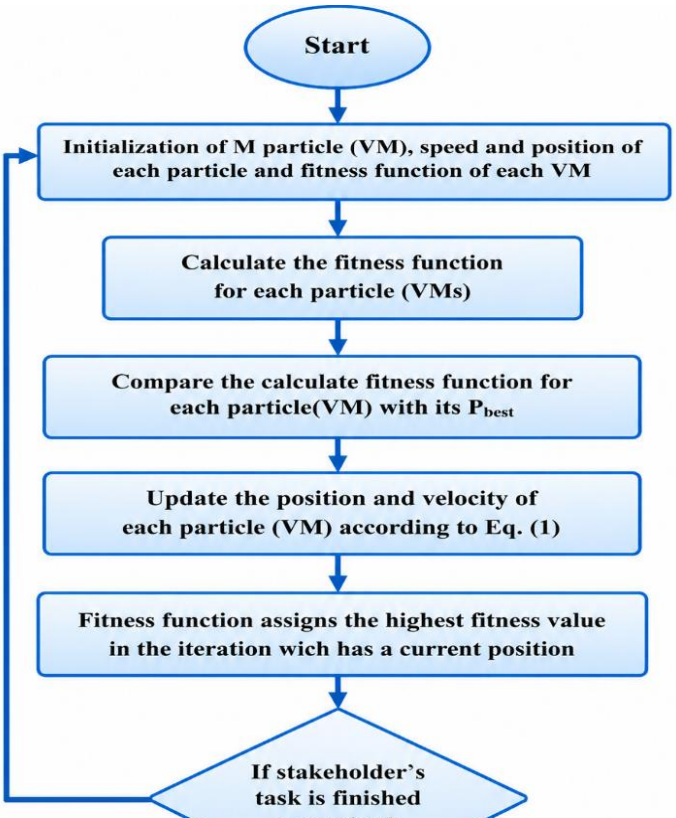
$rand_1$، $rand_2$: Two random numbers with uniform distribution in the range 0 to 1.

$V_i\,(t-1)$ velocity vector in iteration $(t-1)$

$X_i\,(t-1)$ position vector in iteration $(t-1)$

To prevent the velocity of a particle from moving too fast from one location to another (velocity of the velocity vector), we limit the velocity changes to the range V_{max} to V_{min}; Ie $V_{min} \leq V \leq V_{max}$. The upper and lower speed limits are determined by the type of problem [18].

In the implementation stages of PSO algorithm, sometimes there are differences in different authorities in how to separate the implementation steps of the algorithm; That is, in some cases, the steps are mentioned more separately, and in some cases, they combine two or more steps and turn them into one step. But this does not cause any problems in the programming done.



This publication is licensed ur

Figure 2 Steps of implementing PSO

IV. SIMULATION AND EVALUATION

In the section, we evaluate the algorithm (PSO) by performing simulations in the cloud computing platform. To test the proposed algorithm, the CloudSim-3.0.3 simulator platform running in Eclipse 1.0 software is used and by performing various experiments, the results obtained with The proposed algorithms for scheduling tasks are compared. Due to the fact that the CloudSime simulator has been used more widely and comprehensively, CloudSim has also been used for the simulation.

1-4- Parameter setting: In this paper, experiments have been performed with 10 data centers with 5 virtual machines and 10-100 tasks under the cloud simulation platform. Task lengths vary from 1,000 to 10,000 million instructions. The configuration of the cloud simulator is shown in Table (1) and the specifications of the parameters required for the simulator are shown in Table (2). In this paper, to optimize task scheduling to virtual machines, we have implemented the particle swarm optimization algorithm approach and performed several experiments.

Table 2: characteristics of Host

Bandwidth (MB / s)	External memory (MB)	Internal memory (MB)	Processor speed (number of instructions per second)	Number of CPU cores	Host number
10000	2024	1024	1000-5000	2	1-5

This publication is licensed under Creative Commons Attribution CC BY.

Table 3: Necessary parameters for simulation

10	Initial population size	Particle swarm optimization algorithm(PSO)
100	Number of Iteration	
0.2	C_1	
0.2	C_2	
Random number between 0 and 1	$r_{1,2}$	
0.95	W	
10-100	In variable intervals	Number of Tasks
1000-10000	Millions instructions(MI)	Length of tasks
5	On demand	Number of virtual machines supplied

V. EXPERIMENT AND RESULT ANALYSIS

Execution time is one of the most important parameters for scheduling tasks in cloud computing. Although we have less execution time, users will have to wait a short time to get the result of the request in the cloud.

Figure (3) shows the comparison, the average execution time for each task on the virtual machines with the proposed algorithm (PSO), genetic algorithm (GA) and greedy algorithm, which also increases with increasing number of tasks, the average execution time in the algorithm (PSO) The execution time increase rate is less than the genetic algorithm (GA) and the greedy algorithm, the result of which is clearly visible in Figure (3).

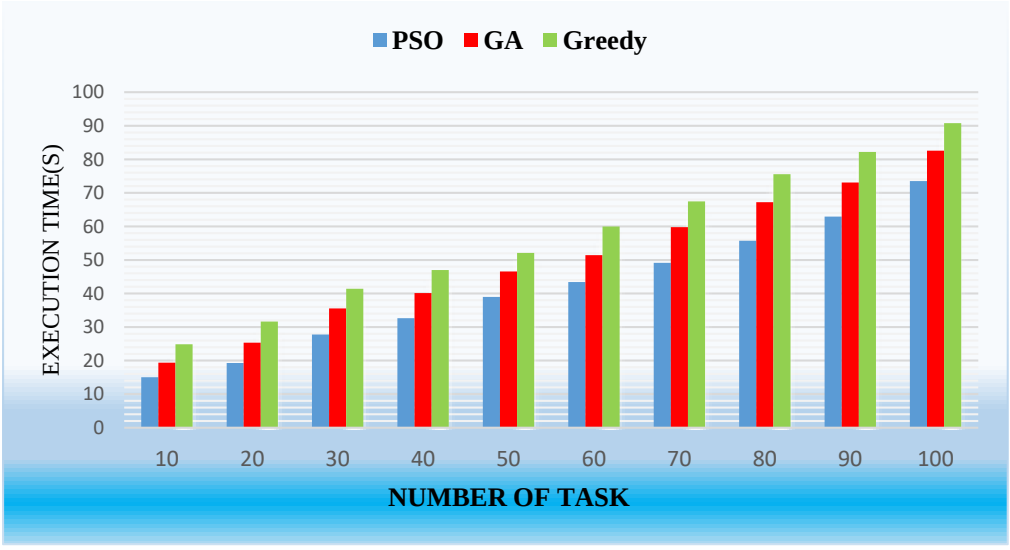


Figure 3: Comparison of Execution time between algorithms (PSO), genetics and greed

Figure (4) shows a comparison of the total execution time for all tasks on the virtual machines with the proposed algorithm (PSO), genetic algorithm (GA) and greedy algorithm. (GA) and the greedy algorithm, the result of which is clearly visible in Figure (4).

This publication is licensed under Creative Commons Attribution CC BY.

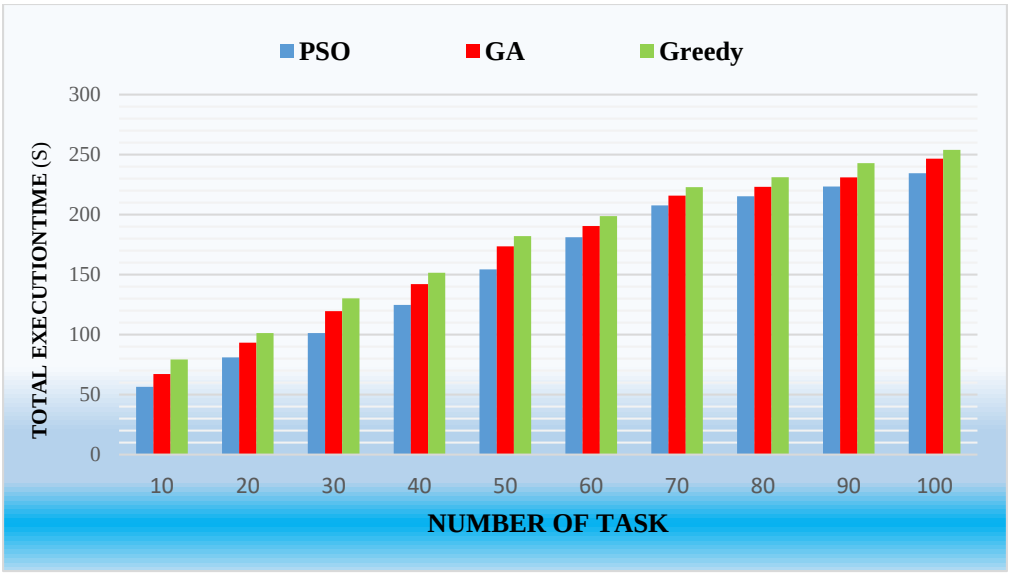


Figure 4: Comparison of total Execution time between algorithms (PSO), genetics and greed

Figure (5) shows the comparison, the average makespan time for each task on the virtual machines with the proposed algorithm (PSO), genetic algorithm (GA) and greedy algorithm.

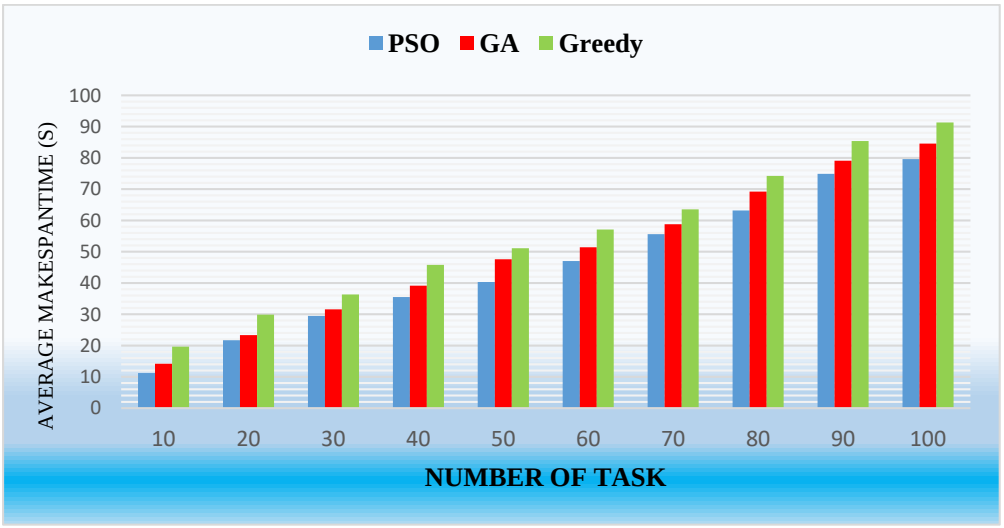


Figure 5: Comparison of makespan time between algorithms (PSO), genetics and greed

Figure (6) shows the comparison of Resource utilization with the proposed algorithm (PSO), genetic algorithm (GA) and greedy algorithm. As can be seen in the figure, the proposed algorithm (PSO) has better efficiency than the genetic algorithm (GA) and the greedy algorithm.

This publication is licensed under Creative Commons Attribution CC BY.

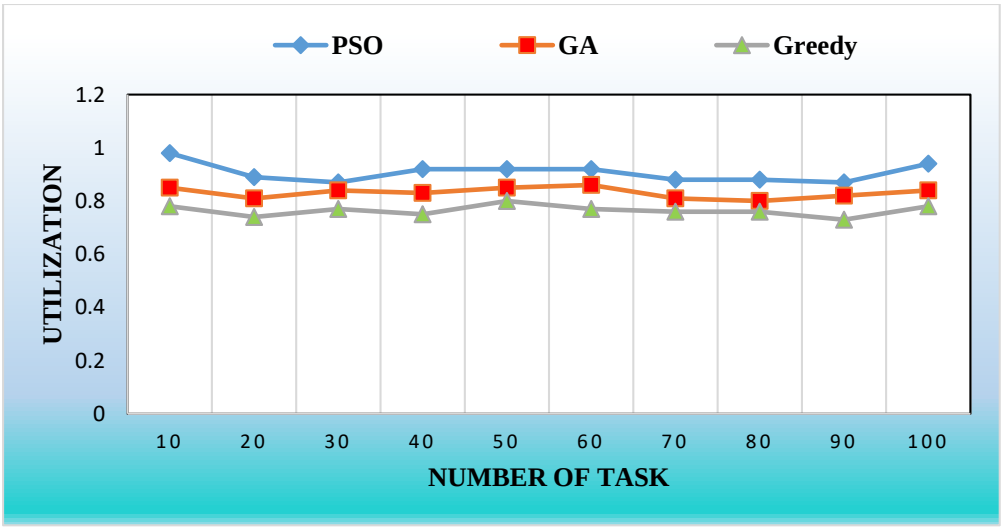


Figure 6: Comparison of Resource utilization between algorithms (PSO), genetics and greed

Figure (7) shows a comparison of the execution cost for each task on the virtual machines with the proposed algorithm (PSO), genetic algorithm (GA) and greedy algorithm, which as the number of tasks increases, the execution cost also increases in the algorithm (PSO) The rate of increase of execution cost is less than genetic algorithm (GA) and greedy algorithm, the result of which is clearly visible in Figure (7).

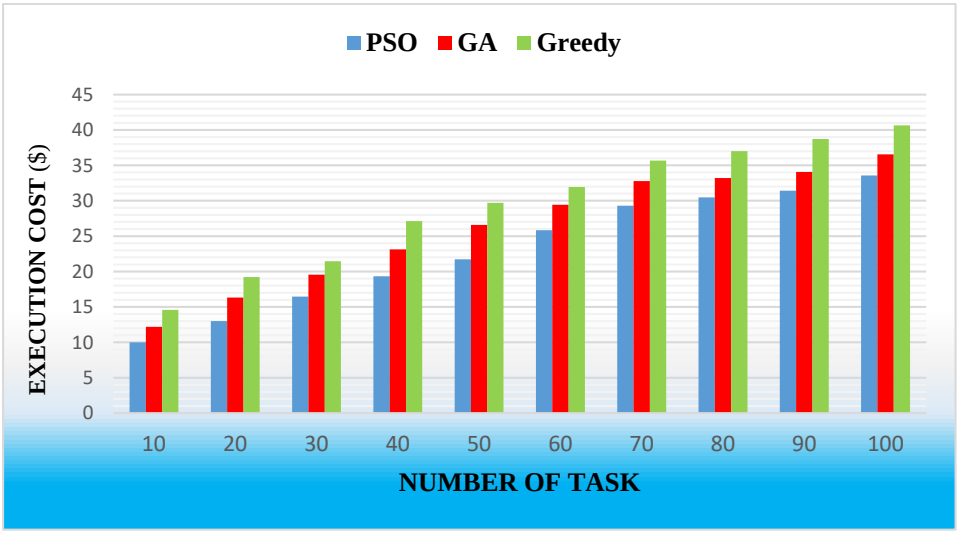


Figure 7: Comparison of execution cost between algorithms (PSO), genetics and greed

Figure (8) shows the comparison, the total execution cost for all tasks on virtual machines with the proposed algorithm (PSO), genetic algorithm (GA) and greedy algorithm. The PSO algorithm works better than the other two algorithms.

This publication is licensed under Creative Commons Attribution CC BY.

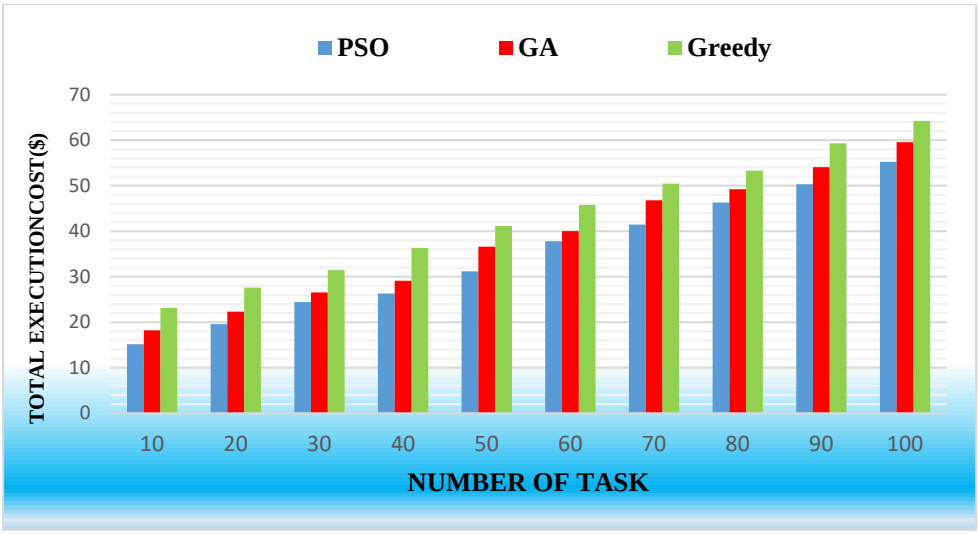


Figure 8 Comparison of total execution cost between algorithms (PSO), genetics and greed

Figure (9) shows a comparison of the total cost for all tasks on virtual machines with the proposed algorithm (PSO), genetic algorithm (GA) and greedy algorithm.

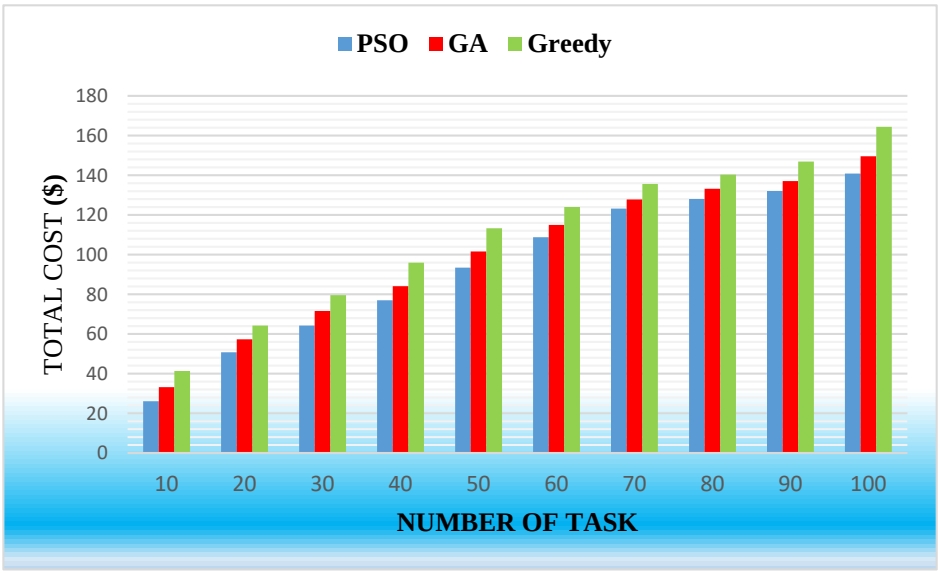


Figure 9: Comparison of total cost between algorithms (PSO), genetics and greed

Figure (10) shows a comparison, Performance improvement rate (PIR %) for all tasks on virtual machines with the proposed algorithm (PSO), genetic algorithm (GA) and greedy algorithm.

This publication is licensed under Creative Commons Attribution CC BY.

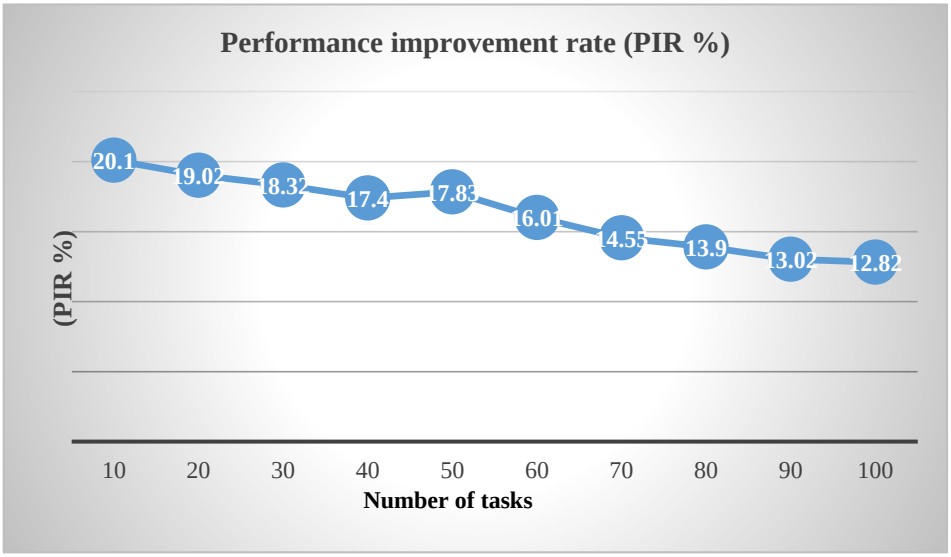


Figure 10: Comparison of PIR between algorithms (PSO), genetics and greed

VI. Conclusion and Future Works

In this paper, a particle swarm optimization algorithm is presented to solve the task scheduling problem with the aim of minimizing service quality parameters in the cloud computing environment. To minimize service quality parameters based on task scheduling in cloud computing, we provide a method that minimizes runtime, completion time, and system cost and solves it by the PSO algorithm. Experimental results show that the PSO algorithm has both an optimal solution and faster convergence than the other two algorithms. The particle swarm algorithm has fewer parameters than the genetic algorithm and is used with the aim of maximizing the efficiency of effective resources and global optimization. The following results are obtained by evaluating and performing different experiments.

The higher the number of requests, the longer the execution time, and the higher the execution time, the higher our cost, so we can have less cost in virtual machines by reducing execution time.

Using the PSO algorithm reduces both execution time and completion time, as well as achieving optimal cost and better productivity. The PSO algorithm has a high convergence speed, so it will get results faster. From the simulation results, we saw that by increasing the number of requests or the number of tasks, completion time, cost and execution time of the proposed algorithm is less than other algorithms, so we can say that the PSO algorithm is more efficient than the genetic algorithm.

Given the growing use of the cloud, the following suggestions will be made in the future. Consideration of energy consumption criteria in cloud computing in order to reduce energy consumption. Development of a proposed algorithm to provide flexible resources in cloud computing. Consider load balancing dynamic task scheduling resources in cloud computing.

VII. References

[1] N. Devi et al., “A systematic literature review for load balancing and task scheduling techniques in cloud computing,” Artificial Intelligence Review, vol. 57, article 276, 2024.

[2] J. Bhagwan and S. Kumar, “Performance Analysis of Tasks Scheduling Algorithms in Homogeneous Cloud Environment,” vol. 10, no. 2, pp. 901–912, 2018.

[3] “A New Task Scheduling Algorithm using Firefly and Simulated Annealing Algorithms in Cloud Computing.” pp. 195–202, 2018.

[4] G. Wang and Y. Zhao, “Interactive fuzzy Bayesian search algorithm for task scheduling in cloud computing environments,” Journal of Engineering and Applied Science, vol. 73, Art. no. 85, 2026,

This publication is licensed under Creative Commons Attribution CC BY.

[5] A. Kamalinia and A. Ghaffari, “Hybrid Task Scheduling Method for Cloud Computing by Genetic and DE Algorithms,”, vol. 97, no. 4, pp. 6301–6323, 2017.

[6] M. Agarwal, Srivastava, G. M. S. “A PSO Algorithm-Based Task Scheduling in Cloud Computing.” International Journal of Applied Metaheuristic Computing, vol. 10, no. 4, pp. 1-17, 2019

[7] A. K. Yadav and H. L. Mandoria, “Study of Task Scheduling Algorithms in the Cloud Computing Environment: A Review,”, vol. 8, no. 4, pp. 462–468, 2017.

[8] O. L. Abraham, M. A. Ngadi, J. M. Sharif, and M. K. M. Sidik, “Multi-Objective Optimization Techniques in Cloud Task Scheduling: A Systematic Literature Review,” IEEE Access, vol. 13, pp. 12255–12291, 2025.

[9] K. Wu, P. Lu, and Z. Zhu, “Distributed Online Scheduling and Routing of Multicast-Oriented Tasks for Profit-Driven Cloud Computing,”, vol. 20, no. 4, pp. 684–687, 2016.

[10] T. Wang, X. Wei, T. Liang, and J. Fan, “Dynamic tasks scheduling based on weighted bi-graph in Mobile Cloud Computing,”, vol. 19, no. March, pp. 214–222, 2018.

[11] F. A. O. Safwat A. Hamad, “Genetic-Based Task Scheduling Algorithm in Cloud Computing Environment,” Int. J. Adv. Comput. Sci. Appl., vol. 7, no. 94, pp. 550–556, 2016.

[12] J. Britto, V. Chaudhari, D. Mehta, A. Kale, and J. Ramteke, “A Priority-Based Max-Min Scheduling Algorithm for Cloud Environment Using Fuzzy Approach,” Int. Conf. CNCT., vol. 15, pp. 819–828, 2019.

[13] S. C. Satapathy, J. K. Mandal, S. K. Udgata, and V. Bhateja, “Task Scheduling Algorithms with Multiple Factor in Cloud Computing Environment,” Adv. Intell. Syst. Comput., vol. 433, pp. 619–627, 2016.

[14] S. Dubey and S. Agrawal, “QoS Driven Task Scheduling in Cloud Computing,” Int. J. Comput. Appl. Technol. Res., vol. 2, no. 5, pp. 595–600, 2013.

[15] M. Agarwal. and G. M. S. Srivastava, “A PSO Algorithm-Based Task Scheduling in Cloud Computing,” IJAMC., vol. 10, no. 4, pp. 1–17, 2019.

[16] Kumar, K. R.Prasanna, and K. Kousalya. “Amelioration of Task Scheduling in Cloud Computing Using Crow Search Algorithm.” Neural Computing and Applications, vol. 32, no. 10, pp. 5901–5907, 2020.

[17] Masdari, Mohammad, et al. “A Survey of PSO-Based Scheduling Algorithms in Cloud Computing.” Journal of Network and Systems Management, vol. 25, no. 1, pp. 122–158, 2017.

[18] Kumar, Mohit, and Subhash C. Sharma. “PSO-Based Novel Resource Scheduling Technique to Improve QoS Parameters in Cloud Computing.” Neural Computing and Applications, vol. 32, no. 16, pp. 12103–12126, 2020.

This publication is licensed under Creative Commons Attribution CC BY.

AUTHORS



First Author – AbdulHakim Hekmat, Assistant Professor, Faculty of Communication Engineering - Department of Telecommunication Technology, Rahe-Abreshume University, Kabul, Afghanistan.

Email: a.hekmat1995@gmail.com

Second Author – Vahid Ranjbar, Assistant Professor, Faculty of Computer Engineering - Department of Information Technology, Yazd University, Yazd, Iran.

Third Author – Mahdi Aqa Sarram, Professor, Faculty of Computer Engineering, Yazd University, Yazd, Iran.

Correspondence Author: AbdulHakim Hekmat, a.hekmat1995@gmail.com,

contact number: +93795334383

This publication is licensed under Creative Commons Attribution CC BY.