Fuzzy Based Ant Colony Optimization Scheduling in Cloud Computing

Article *in* Computer Systems Science and Engineering · September 2021

CITATIONS
7

READS
4,202

6 authors, including:

Garima Verma
DIT University
73 PUBLICATIONS **520** CITATIONS
SEE PROFILE

Dilip Kumar Sharma
Jaypee University of Engineering and Technology Guna
148 PUBLICATIONS **3,987** CITATIONS
SEE PROFILE

Sudhakar Sengan
PSN College of Engineering and Technology
189 PUBLICATIONS **2,992** CITATIONS
SEE PROFILE

Computer Systems Science & Engineering
DOI:10.32604/csse.2022.019175
Article

Tech Science Press

Fuzzy Based Ant Colony Optimization Scheduling in Cloud Computing

K. Rajakumari[1,*], M.Vinoth Kumar[2], Garima Verma[3], S. Balu[4], Dilip Kumar Sharma[5] and Sudhakar Sengan[6]

[1]Department of Computer Science and Engineering, School of Engineering, Avinashlingam Institute for Home Science and Higher Education for Women, Coimbatore, 641043, Tamil Nadu, India
[2]Department of Information Science and Engineering, Dayananda Sagar Academy of Technology and Management, Bangalore, 560082, India
[3]School of Computing, DIT University, Dehradun, 248009, Uttarakhand, India
[4]Department of Computer Science and Engineering, Paavai Engineering College, Pachal, 637018, Tamil Nadu, India
[5]Department of Mathematics, Jaypee University of Engineering and Technology, Guna, 473226, M.P., India
[6]Department of Computer Science and Engineering, PSN College of Engineering and Technology, Tirunelveli, 627152, Tamil Nadu, India
*Corresponding Author: K. Rajakumari. Email: raji1anjuphd@gmail.com
Received: 05 April 2021; Accepted: 10 May 2021

Abstract: Cloud computing is an Information Technology deployment model established on virtualization. Task scheduling states the set of rules for task allocations to an exact virtual machine in the cloud computing environment. However, task scheduling challenges such as optimal task scheduling performance solutions, are addressed in cloud computing. First, the cloud computing performance due to task scheduling is improved by proposing a Dynamic Weighted Round-Robin algorithm. This recommended DWRR algorithm improves the task scheduling performance by considering resource competencies, task priorities, and length. Second, a heuristic algorithm called Hybrid Particle Swarm Parallel Ant Colony Optimization is proposed to solve the task execution delay problem in DWRR based task scheduling. In the end, a fuzzy logic system is designed for HPSPACO that expands task scheduling in the cloud environment. A fuzzy method is proposed for the inertia weight update of the PSO and pheromone trails update of the PACO. Thus, the proposed Fuzzy Hybrid Particle Swarm Parallel Ant Colony Optimization on cloud computing achieves improved task scheduling by minimizing the execution and waiting time, system throughput, and maximizing resource utilization.

Keywords: Cloud Computing; scheduling; ant colony optimization; fuzzy logic

1 Introduction

Cloud computing is a fast-growing technology that allocates distributed modern computing systems and resources to hardware and software, allowing for more efficient resource utilization. Cloud computing discharges with minimal downtime efficiently. Cloud computing possesses dynamic provisioning, and this technique is not only applicable for the cloud service, but it can also compute the capability, storage,

networking, and information technology infrastructure. Cloud services are helpful to everyone using a digital wallet, with the framework supporting the increasing and diminishing software in response to requests and charges for the period during which these tools were used. Fig. 1 depicts the different types of interconnected computers, including laptops, desktop computers, mobiles, tablets, servers, and databases. Data will be stored, and programs will be run so that Internet-connected applications will access cloud data. Several research areas have been developed to express complex computational issues that can be efficiently processed on cloud computing infrastructures. The primary goal of the research is to enhance the conventional task scheduling algorithms that is essential for resource utilization on cloud computing systems.

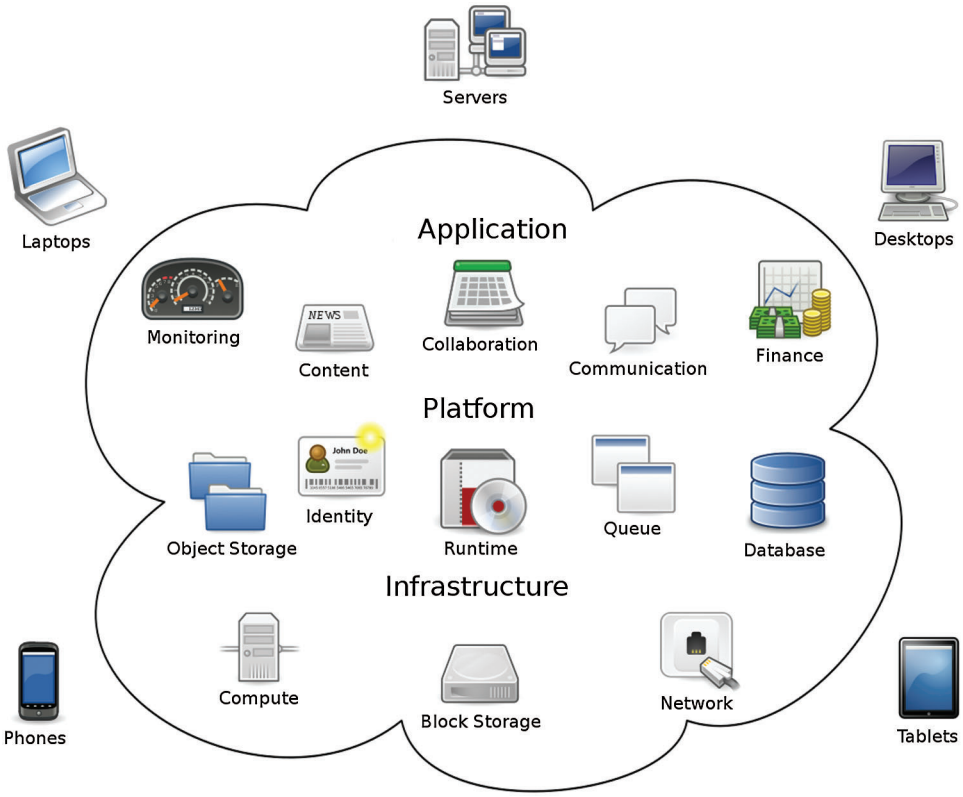

Figure 1: A model of cloud computing platform

Task scheduling cannot be accomplished by focusing on a single criterion. Still, task scheduling can be performed by arrangements such as Quality of Service (QoS) to the client based on the cloud service providers' key roles. However, a significant number of tasks is running on the cloud service provider's role. Task scheduling could be thought of as a quest for the optimization to assign a sequence of subtasks from different tasks to the available virtual servers to attain the intended task scheduling goal. As a result, this paper investigates conventional task scheduling algorithms, which are improved using a hybrid system that can help cloud services achieve an excellent cloud service level.

As in the Information Technology field, cloud computing is a new booming field that has emerged as an authenticity. There are a few elements of the cloud that can be changed. Task scheduling is among the primary sources of concern. Since big data management is becoming more popular in the cloud computing environment, it must process the information effectively. The assigning of tasks to versatile resources according to flexible time helps determine a logical sequence in which the tasks are completed.

Assignments should be planned effectively in a cloud computing environment to minimize reaction time, waiting time, running time, computation time, and resource usage. The task scheduling issue is crucial not only for attaining maximum cloud efficiency but also for meeting the demands of different Cloud users in an appropriate way to improve the overall performance of cloud computing.

The primary motive of task scheduling in task management is to prioritize the tasks in the cloud computing environment to minimize time, avoid losing work, and succeed in the task's deadlines. Task scheduling improves the cloud computing system to optimize the benefit from high-performance computing and best machine performance. The scheduling algorithm distributes the workload across processors, maximizing their efficiency while reducing overall task execution time. This paper's primary goal is to reduce the computation time and execution cost, optimize resource usage, reduce transmission rate, and reduce overall execution.

2 Literature Survey

Multi priority-based quality of service scheduling (2016) [1] was suggested by cloud computing. The recommended approach's key goals were to spread services to the greatest extent. The new principles are selected to allocate the scheduling goals to each role for task clustering mixed among QoS and Vendor costs. The proposed algorithm is used to ensure minimum execution time for all cloud computing systems, less client latency, and the cloud service provider's minimized cost.

Energy management in cloud computing [2] was prompted based on the task scheduling algorithm using game theory. In this paper, the simplified model was proposed for the task scheduling algorithm based on the game theory as a mathematical tool on cloud computing. The task scheduling algorithm has the reliability of the balanced task based on the game theory. The game strategy in the cooperative game model was used for the task in the measurement of rate allocation strategy on the node. However, according to the internal scheduler, the processing cost is required to improve the scheduling performance was not measured.

Provisioning of resources and scheduling plans in the Infrastructure as a Service (IaaS) cloud servers [3] are suggested using the Augmented Shuffled Frog Leaping Algorithm. The optimum task scheduling for collecting dependent tasks in an NP-hard problem was considered in this study. Implementation of Particle Swarm Optimization and Shuffled Frog Leaping Algorithm to resource integration and workflow scheduling in clouds were recently identified. Then, the SFLA was formulated as an augmented variation for obtaining better cost-optimal solutions and congregating deadline constraints. However, the overall execution cost was high.

QoS-driven task scheduling [4] was implied in cloud computing. This study's key objective was to investigate the fixed priority pre-emptive task scheduling algorithms in cloud computing to improve the QoS parameters. This review recommended two fixed-priority scheduling algorithms such as Rate Monotonic and Deadline Monotonic scheduling algorithms. Also, different types of task scheduling algorithms were discussed.

Load balancing for nonpre-emptive based tasks in cloud computing [5] suggested a weighted round-robin algorithm that was enhanced. The introduced algorithm's main goal was to improve VM efficiency by combining static and dynamic load balancing depending on the job duration, resource availability, interdependencies of different tasks, under utilized VM prediction, and overflow elimination of the VM. Thus, the overload on a VM and task migrations were minimized by using the proposed algorithm. However, the job end time was not reduced.

Cloud resource allocation [6] was planned as the nonpre-emptive approach. In this paper, the anticipated algorithm utilized the turnaround time effectively by discriminating it into the gain and loss function for a single task according to priorities. The tendered algorithm was executed on both pre-emptive and

nonpre-emptive approaches. Moreover, further details on scheduling algorithms were also discussed. The computation time of the algorithm was improved by using the Nephele method. However, the resources were not provided dynamically.

3 Dynamic WRR Scheduling in Cloud Computing

In most cases, task scheduling is defined as the method of allocating the resources to complete a task that has been stated by the scheduling method. Digital computing components such as threads, processes or information flows may be included in the mission, which is then scheduled into hardware resources such as processors, network links, etc. The task scheduler is a computer operating system that determines which tasks will be accepted by the system, and those tasks will be executed immediately. Cloud computing has become a real competitor in the information technology field in recent decades. They nevertheless need to be developed more. Since extensive data analysis is progressively taking place in cloud environments, it must provide data quickly. Task scheduling requires flexible resources based on the flexible time that includes finding an appropriate sequencing in which assignments can be performed when needed. Tasks should be organized efficiently in such states to minimize reaction time, wait period, processing time, makespan, and resource usage.

Task scheduling is significant for achieving the highest cloud service performance and satisfying the different cloud users' demands equitably. Therefore, the overall performance of the cloud computing infrastructure is enhanced. The task scheduling is accomplished in this method depending on various task scheduling algorithms in the cloud computing framework. Different scheduling algorithms addressed here are First-Come First-Served, Round Robin, Weighted Round Robin, and Dynamic Weighted Round Robin. Eventually, the comparative analysis reveals that these algorithms' performance efficiency is different for different tasks.

3.1 Proposed Dynamic Weighted Round Robin (DWRR) Scheduling

The suggested DWRR scheduling is focused on every task's additional dynamic weighting factor. The recommended DWRR is used to set priorities to the most appropriate VM focusing on VM data such as processor speed, load on the VM, and the duration and importance of the tasks that have been distributed. The WRR algorithm's static scheduling uses the VM's computing energy, the number of new tasks, and each task's period to decide the best VM for the job. WRR's dynamic scheduling often considers the load on each VM, as well as VM details, when determining which VM should be assigned to a mission. There is a possibility that in some cases, the process may require more extended execution time than the initial time due to the execution of a more significant number of cycles on the exact instructions based on the complex execution time information.

In such cases, the load balancer euthanizes the scheduling controller and reconfigures the tasks from heavy load VM as per the idle slot available in another unused VM. The load balancer recognizes the new VM via resource prober when a job is completed in any VM. If no unused VM remains, the load balancer will not occupy any VM task transfer. If the load balancer detects some unused VM, it moves from overloaded VM to a new VM. The load balancer assesses the VM load only after completing any of the VM tasks. The load balancer never investigates VM load to bypass overhead on VM. It will help to decrease the number of task migrations among VM and resource probe execution in VM.

3.2 Algorithm for DWRR Scheduling

Input: Set of tasks and VM

Output: Scheduling of tasks to VM

1. Consider the M set of tasks
2. Assign N set of VM
3. Consider $w_{i,k}$ as the dynamic weight counter of queue i at round k
4. Consider $q_{i,k}$ as the current task of queue i at round k
5. Assume $cw_{i.k}$ as the weight counter of queue i at round k
6. $cw_{i.k} \leftarrow 0 \;\; (i = 0, 1, 2, \ldots, n-1)$
7. $k \leftarrow 0$
8. *For* $i \leftarrow 0$ *to* $n-1$ Do
9. *If* $\left(q_{i,k} \neq NULL\right)$ Then
10. Compute $w_{i,k}$

$$w_{i,k} = w_{i,k} W_i$$

$$W_i = \frac{MRT_i}{\sum_{i=1}^{N} MRT_i}$$

11. $cw_{i.k} \leftarrow w_{i,k}$
12. *If* $\left(\left(q_{i,k} \neq NULL\right)\&\&(cw_{i.k} \neq NULL)\right)$ Then
13. Provide task from $q_{i,k}$ to CPU using WRR
14. Else If
15. $k \leftarrow k + 1$
16. End If
17. End For
18. End

4 Hybrid Particle Swarm Parallel Ant Colony Optimization Algorithm

To obtain optimized task scheduling in cloud computing, a hybrid PSO and PACO are proposed in this paper. Here, the hybrid algorithm combines Particle Swarm Optimization and Parallel Ant Colony Optimization [7]. For *'n'* tasks and *'m'* VMs, each particle represents a practical scheduling method. PSO evaluates each particle's fitness value using a parallel ACO algorithm and then finds particle best and global best for the optimal solution. Initially, PSO randomly initializes tasks, position, and velocity vector. Then PSO computes the cost, makespan time, and processor utilization. In this research work, parameters are considered as fitness values [8]. The fitness value of each particle is estimated by using a parallel ACO algorithm. In the parallel ACO algorithm, the ant colony population is partitioned into sub-ant colonies. Each ant sub-colony consider the particle best and global best. Based on the fitness value of each ant subcolonies, the particle and global best of each particle in PSO are updated. The fitness value is measured by the weighted sum of the cost of computing, makespan time, and processor utilization. It is represented in Fig. 2.

In Eq. 1, c represents the cost of computing, T is makespan, and U is processor utilization.

$$Fitness = \sum w_1 \frac{1}{C} + w_2 \frac{1}{T} + w_3 U \qquad w_1 + w_2 + w_3 = 1 \tag{1}$$

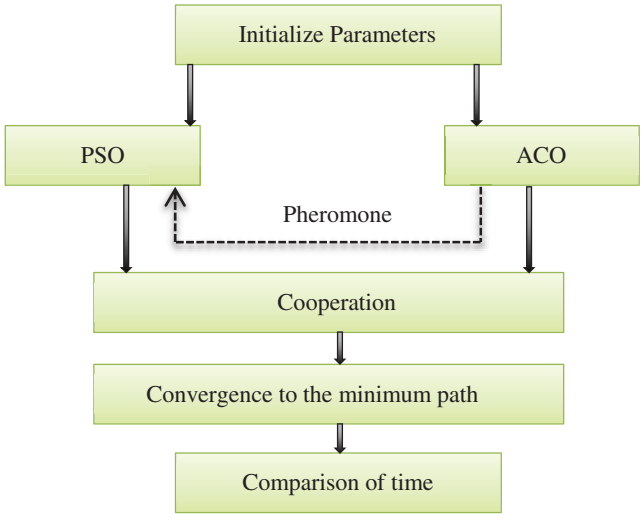

Figure 2: System Architecture of HPSPACO

4.1 Algorithm of Hybrid Particle Swarm Parallel Ant Colony Optimization (HPSPACO) Algorithm

Input: Set of tasks, position, velocity, list of VMs

Output: Scheduled Task

1. Set particle dimension is equal to the size of prepared tasks in $\{t_i\} \in T$
2. Initialize population and velocity randomly
3. Assess the fitness value for each particle using Parallel ACO
4. Initialize number of ants i with a set of particles in PSO
5. For (i = 1; i < = k; i++)
6. Initialize starting pheromone information by scheduling task in sub-ant colonies
7. Find the probability of m^{th} ant choosing VM 'b' for the next task, denoted in Eq. 2.

$$P_{ab}^{m} = \frac{\left(\tau_{ab}^{\alpha}\right)\left(\eta_{ab}^{\beta}\right)}{\sum\left(\tau_{ab}^{\alpha}\right)\left(\eta_{ab}^{\beta}\right)} \tag{2}$$

8. While (Max(Fitness)) Do
9. If task scheduling is Normal Generation
10. Create 'm' solutions from the graph
11. If migration generation, then
12. Send the information about the best solution in the sub-ant colony to all neighboured ant colonies
13. Find the solutions and select 'm'; best solutions among them
14. Update each particle's velocity and position based on *m* best solutions
15. End While
16. End If
17. End

5 Fuzzy Hybrid Particle Swarm Parallel Ant Colony Optimization (FHPSPACO) in Cloud Computing

Task scheduling determines the best order in which the tasks could be completed while adhering to transaction logic constraints. The PSO and PACO are basic task scheduling algorithms that require less computing time in a cloud computing environment [9]. The PSO and PACO have a flaw in proposed algorithms; they need to seek the best solutions because they lack a framework for parameter adaptations. The new approach presented here combines Fuzzy Logic with PSO (FPSO-Fuzzy Particle Swarm Optimization) and PACO for resolving the Scheduling issue. A Fuzzy System (FS) is recommended for the Inertia weight upgrade in the PSO algorithm; however, a current fuzzy system for the pheromone trail upgrade's weighting coefficient is applied the PACO algorithm [10].

Fuzzy-HPSPACO optimization is suggested where the scheduling decision is made by assessing the complete set of tasks in the job scheduling. The proposed fuzzy controller uses fuzzy logic to assign elements based on binary values of 0 to 1. The location and velocity updating formulas are used to conduct the fuzzy PSO, incorporating neighborhood information about communication strategy. PACO is used unless the most substantial particle, the global best, does not change after a certain number of generations [11]. The best person from the fuzzy PSO algorithm proposals value valued PACO results during the hybrid search method. The Fuzzy-based PSO achieves better task scheduling, and Fuzzy-based HPSPACO reduced PACO convergence [12].

5.1 Hybrid Algorithm with Fuzzy Hybrid Particle Swarm Parallel Ant Colony Optimization

Input: **Set of** tasks, position, velocity, m

Output: scheduled tasks

1. Set of particle dimension is equal to the size of ready tasks in $\{t_i\} \in T$
2. Initialize population, position, and velocity of particle randomly
3. Evaluate the fitness value of each particle
4. Evaluate the Normalized fitness value of the current best position for each particle
5. Update the position and velocity of the particle
6. Create fuzzy rules using Normalized fitness of the current best position of the particle and current value of the inertia weight
7. Defuzzify the rules by centroids set of methods
8. For (n=0; n<m; n++)
9. If (g_{best} is Updated)
10. Go to Step 3
11. Else If
12. Go to Step 15
13. End If
14. End For
15. Initialize set of ants *i* with different scheduled tasks
16. For (i=1; i<=k; i++)
17. Initialize starting pheromone information by scheduling tasks in sub-ant colonies

18. Move the ants from A to B state with probability using the following Eq. 3

$$P_{ab}^{m} = \frac{\left(\tau_{ab}^{\alpha}\right)\left(\eta_{ab}^{\beta}\right)}{\sum\left(\tau_{ab}^{\alpha}\right)\left(\eta_{ab}^{\beta}\right)} \tag{3}$$

19. Calculate the Normalized weight of the current best position of the ant
20. Create Fuzzy rules for PACO based on the Normalized weight of the current best position of ant and the weighting coefficient α_b^a
21. While (Max (Fitness)) Do
22. If this is an average generation
23. Create *'m'* solutions from graph
24. If this is a migration generation
25. Send the information about the best solution in sub ant colony to all neighbored ant colonies
26. Collect solutions and select *'m'* best solutions among them
27. Update the pheromone information depending on the fitness of the solution
28. End While
29. End If
30. End

6 Results and Discussion

In this section, the performance of the proposed DWRR, HPSPACO, and FHPSPACO task scheduling methods is evaluated and compared among the proposed task scheduling methods. In the description of the experimental cloud setup in this section, simulation is performed using CloudSim [13]. The proposed DWRR, HPSPACO, and FHPSPACO based task scheduling are compared in resource utilization, execution time, waiting time, throughput, and makespan.

6.1 Resource Utilization

Fig. 3 demonstrates the resource usage distinction within suggested DWRR and HPSPACO, with the Y-axis indicating resource utilization [14]. The suggested FHPSPACO has higher resource consumption than other task scheduling methods.

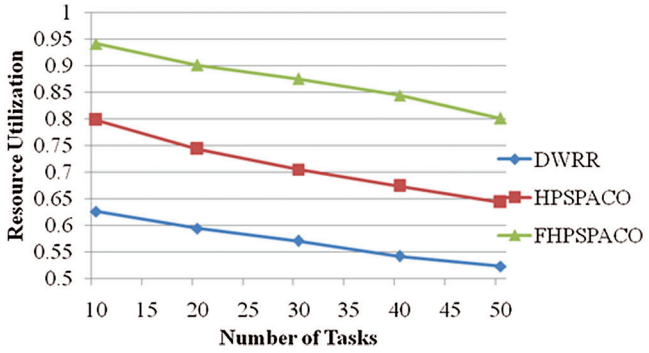

Figure 3: Comparison of Resource utilization: DWRR, HPSPACO, and FHPSPACO

6.2 Execution Time

Fig. 4 provides a comparative study of DWRR, FHPSPACO, and HPSPACO task scheduling methods in terms of execution time. The number of virtual machines is represented by the X-axis, and Y-axis represents the execution time in *minutes*. It has been demonstrated that the proposed FHPSPACO takes less time to execute than other task scheduling methods.

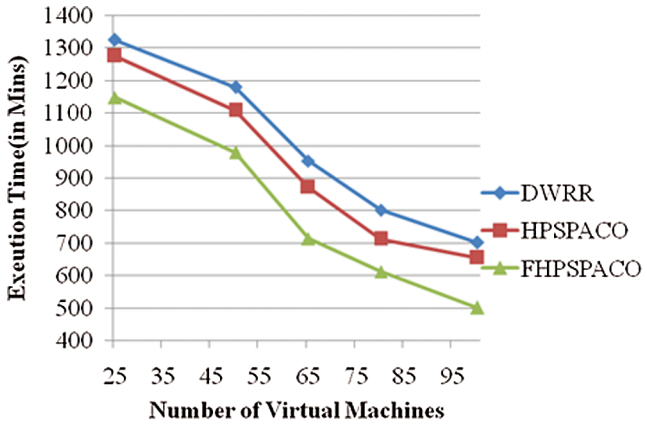

Figure 4: Comparison of Execution Time: DWRR, HPSPACO, and FHPSPACO

6.3 Waiting Time

Fig. 5 compares the waiting times for the task scheduling methods DWRR, HPSPACO, and FHPSPACO [15]. It has been shown that the suggested FHPASO requires less time than other task scheduling strategies.

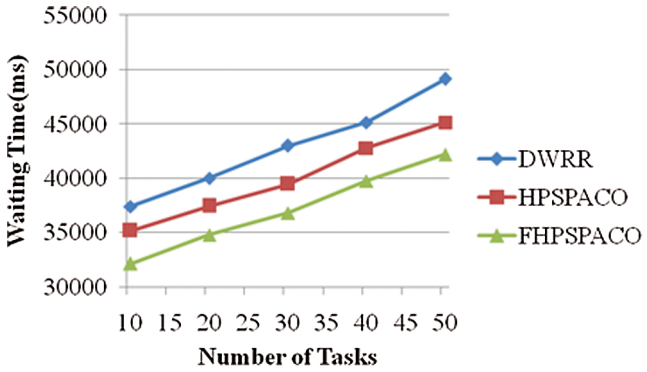

Figure 5: Comparison of Waiting Time: DWRR, HPSPACO, and FHPSPACO

6.4 Throughput

Fig. 6 demonstrates the comparison of throughput between DWRR, HPSPACO, and FHPSPACO task scheduling methods. It is proved that the proposed FHPSPACO has better throughput than the other task scheduling methods.

6.5 Makespan

Fig. 7 compares the make spans of the task scheduling methods DWRR, HPSPACO, and FHPSPACO. The suggested FHPSPACO has been shown to have a shorter makespan than other task scheduling strategies.

The cumulative efficacy of the suggested Dynamic Weighted Round-Robin, Hybrid Particle Swarm Parallel Ant Colony Optimization, and Fuzzy Hybrid Particle Swarm Parallel Ant Colony Optimization is exposed in this section. From this section, Makespan is proved that the proposed FHPSPACO has determined resource utilization, which is 34.53% better than DWRR and 18.32% better than HPSPACO, less execution time, which is 20.29% better than DWRR and 14.46% improved than HPSPACO, less waiting timeis 13.53% better than DWRR and 7.16% improved than HPSPACO, least makespan is10.14% better than HPSPACO and minimum throughput is 42.06 % better than DWRR and 27.06% improved than HPSPACO. As an outcome, this section successfully describes the overall performance effectiveness of the proposed method FHPSPACO.

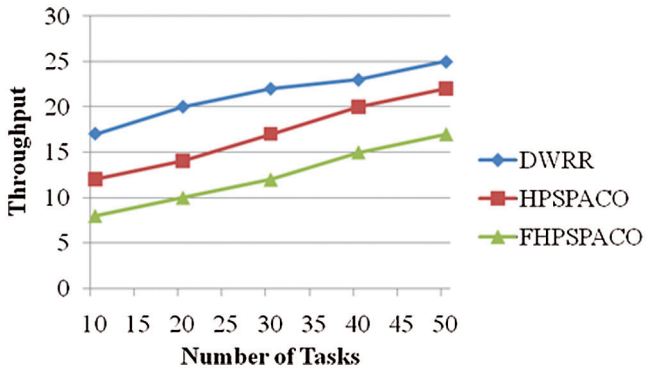

Figure 6: Comparison of Throughput: DWRR, HPSPACO, and FHPSPACO

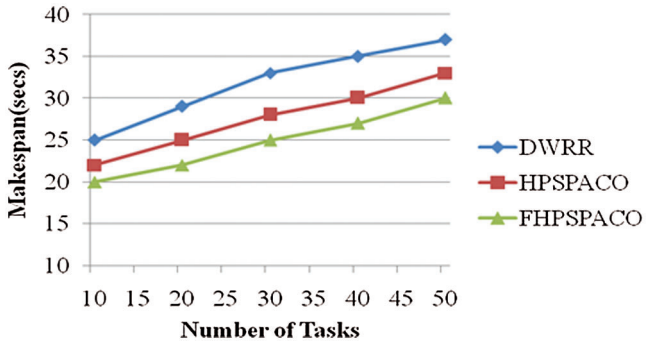

Figure 7: Comparison of Makespan: DWRR, HPSPACO, and FHPSPACO

The task scheduling problem was overcome using the Dynamic Task Scheduling model named Dynamic Dispatch Queue and simulated annealing methodologies [16] in the cloud environment by cloud service providers to improve customers' satisfaction efficiently with particle swarm optimization. A pair-based task scheduling model for the cloud has been using the optimization algorithm [17] with an unequal number of clouds and tasks based on first-come, first-serve, and Hungarian algorithms. Solving NP-hard in a heterogeneous environment and quality of service to schedule tasks in the cloud environment through cloud service providers are made by a parallel multi-objective genetic model [18]. The parallel multi-objective genetic model provides a load balance system with the help of a genetic algorithm with a specialized scheduler and particle swarm optimization.

7 Conclusion

This paper's primary objective effectively to enhance the scheduling of tasks in cloud computing, which is according to fuzzy rules. Different Meta-heuristic algorithms such as Genetic Algorithm, Particle Swarm Optimization, Ant Colony Optimization, Artificial Bee Colony, and Parallel Ant Colony Optimization are developed for task scheduling in the cloud computing environment. Using such algorithms, the tasks are scheduled efficiently in terms of minimized execution, waiting and response time, makespan, and resource utilization. On the other hand, task scheduling is essential for attaining maximum cloud service efficiency and frequent cloud users' demands while also enhancing the complete cloud computing Quality of Service. As an outcome of the improved optimization algorithms used in this investigation, task scheduling performance and efficiency are improved.

8 Future Work

The future extension of this research would be included in optimizing a more significant number of objectives such as task availability, energy efficiency, user's comprehensive QoS, etc. Moreover, an efficient optimal task scheduling could be enhanced by novel optimization methods such as Cat Swarm Optimization, Cockroach Swarm Optimization, Bean Optimization, etc., which may provide the best and optimized results.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] E. Meriam and N. T. Mediatron, "Multiple QoS priority-based scheduling in cloud computing," in *Signal, Image, Video and Communications (ISIVC), International Symposium on IEEE*, Tunis, Tunisia, pp. 276–281, 2016.

[2] J. Yang, H. Xu, L. Pan, P. Jia, F. Long *et al.,* "Task scheduling using Bayesian optimization algorithm for heterogeneous computing environments," *Applied Soft Computing*, vol. 11, no. 4, pp. 3297–3310, 2011.

[3] P. Kaur and S. Mehta, "Resource provisioning and workflow scheduling in clouds using augmented Shuffled Frog Leaping Algorithm," *Journal of Parallel and Distributed Computing*, vol. 101, pp. 41–50, 2017.

[4] S. Dubey and S. Agrawal, "QoS driven task scheduling in cloud computing," *International Journal of Computer Applications Technology and Research*, vol. 2, no. 5, pp. 595–600, 2013.

[5] D. C. Devi and V. R. Uthariaraj, "Load balancing in cloud computing environment using improved weighted round-robin algorithm for non-preemptive dependent tasks," *Scientific World Journal*, vol. 2016, pp. 1–14, 2016.

[6] G. Upadhye and T. Dange, "Cloud resource allocation as non-preemptive approach," in *Current Trends in Engineering and Technology (ICCTET), 2nd International Conference on IEEE*, Coimbatore, India, pp. 352–356, 2014.

[7] P. Mathiyalagan, U. R. Dhepthie and S. N. Sivanandam, "Enhanced hybrid PSO-ACO algorithm for grid scheduling," *International Journal on Soft Computing (IJCS)*, vol. 1, no. 1, pp. 54–59, 2010.

[8] K. Etminani and M. Naghibzadeh, "A min-min max-min selective algorithm for grid task scheduling," in *Internet, ICI 2007, 3rd IEEE/IFIP International Conference in Central Asia on IEEE*, Tashkent, Uzbekistan, pp. 1–7, 2007.

[9] A. K. Bardsiri and S. M. Hashemi, "A review of workflow scheduling in cloud computing environment," *International Journal of Computer Science and Management Research*, vol. 1, no. 3, pp. 348–351, 2012.

[10] R. S. Chang, J. S. Chang and P. S. Lin, "An ant algorithm for balanced job scheduling in grids," *Future Generation Computer Systems*, vol. 25, no. 1, pp. 20–27, 2009.

[11] C. Cheng, L. Li and Y. Wang, "An energy-saving task scheduling strategy based on vacation queuing theory in cloud computing," *Tsinghua Science and Technology*, vol. 20, no. 1, pp. 28–39, 2015.

[12] M. Dorigo, M. Birattari and T. Stutzle, "Ant colony optimization," *IEEE Computational Intelligence Magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[13] M. Habibi and N. J. Navimipour, "Multi-objective task scheduling in cloud computing using an imperialist competitive algorithm," *International Journal of Advanced Computer Science & Applications*, vol. 1, no. 7, pp. 289–293, 2016.

[14] A. E. Keshk, A. B. El-Sisi and M. A. Tawfeek, "Cloud task scheduling for load balancing based on intelligent strategy," *International Journal of Intelligent Systems and Applications*, vol. 6, no. 5, pp. 25–36, 2014.

[15] W. Lin, C. Liang, J. Z. Wang and R. Buyya, "Bandwidth-aware divisible task scheduling for cloud computing," *Software Practice and Experience*, vol. 44, no. 2, pp. 163–174, 2014.

[16] B. A. Hicham, B. A. Said and E. Abdellah, "A dynamic task scheduling algorithm for cloud computing environment," *Recent Advances in Computer Science and Communications*, vol. 13, no. 2, pp. 296–307, 2020.

[17] S. K. Panda, S. S. Nanda and S. K. Bhoi, "A pair-based task scheduling algorithm for cloud computing environment," *Journal of King Saud University - Computer and Information Sciences*, 2018, https://doi.org/10.1016/j.jksuci.2018.10.001.

[18] M. Sardaraz and M. Tahir, "A parallel multi-objective genetic algorithm for scheduling scientific workflows in cloud computing," *International Journal of Distributed Sensor Networks*, vol. 16, no. 8, pp. 1–13, 2020.