Advances in Computer Science and Information Technology (ACSIT)
p-ISSN: 2393-9907; e-ISSN: 2393-9915; Volume 10, Issue 2; April-June 2023, pp. 55-60

http://www.krishisanskriti.org/Publication.html

Optimisation of Task Scheduling in Cloud using Bio-Inspired Hybrid Algorithms based on Meta-Heuristic Approach

Siddhesh Bhujbal[1] and Dr. Tanuja R. Pattanshetti[2]

B.Tech, Computer Engineering COEP Technological University Pune, India
Assistant Professor, Computer EngineeringCOEP Technological University Pune, India
E-mail: [1]bhujbalsa19.comp@coep.ac.in, [2]trp.comp@coep.ac.in

Abstract—*Cloud computing is the latest technological advancement in information technology that assures an on-demand delivery of various computing resources like storage, databases, software, networking, servers as a service instead of a product over a internet. Cloud services has acquired business popularity by providing many benefits including cost reduction, improved performance, flexibility and scalability. As millions of users has to be served simultaneously, it must have the ability to meet all users requirements with high performance. Therefore, there is a necessity to implement a robust task scheduling algorithm to efficiently meet these requirements. As a cloud performance mainly depends on task scheduling, it poses a major concern in cloud computing environment. It is an NP-hard combinatorial optimisation problem to schedule tasks for the cloud. For the purpose of solving it, numerous meta-heuristic techniques have been proposed. Here, we attempt to improve cloud performance by minimising makespan by integrating bio-inspired meta-heuristic algorithms like Artificial Bee Colony (ABC) Optimisation and Ant Colony Optimisation (ACO). The CloudSim modelling and simulation framework is used to simulate the aforementioned hybrid ACO-ABC algorithm. The experimental results shows that the hybrid ACO-ABC algorithm reduces makespan of given task set as compared to ACO and ABC optimisation algorithms.*

Index Terms*: Task Scheduling, Cloud Computing, Ant Colony Optimisation, Artificial Bee Colony Optimisation.*

There are three main models of cloud computing: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). IaaS offers virtualised server, storage, and networking resources. A set of tools to develop, test and deploy applications is offered by PaaS. Software as a service (SaaS) companies offer online-accessible software applications. Organisations must select the service model that best meets their needs among the various options because each has advantages and disadvantages of its own. SaaS offers simple access to software apps without a requirement for installation or maintenance, PaaS simplifies application development and deployment and IaaS offers more control and flexibility. The level of responsibility between the cloud provider and the customer is determined by the cloud service models, which are essential to cloud computing.

INTRODUCTION

Cloud Computing

Cloud computing is a paradigm shift in the field of computing that has revolutionized the way computing resources are utilized and consumed. It has become a significant technological innovation that provides on-demand access to a pool of configurable computing resources that are shared, including servers, storage, software, and services. This model eliminates the need for businesses to maintain their own physical infrastructure by enabling users to access computing resources on a pay-per-use basis. The cloud computing model has gained immense popularity due to its numerous benefits such as scalability, flexibility, agility, reduced costs, and increased efficiency. It has become an essential part of modern businesses and has a significant impact on how computing services are delivered and managed.

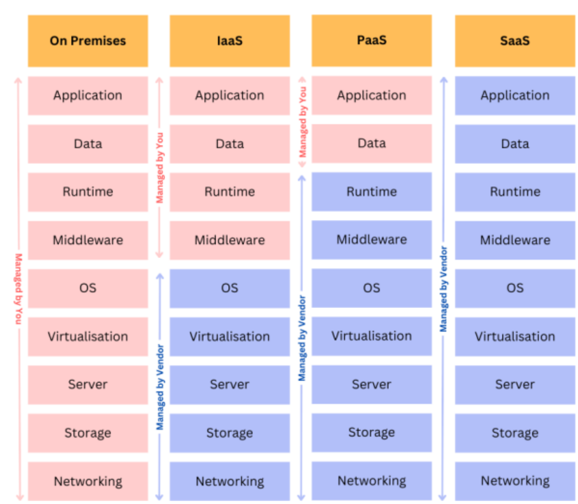

Fig. 1. Cloud Service Models

The various methods for deploying or delivering cloud computing services to customers are referred to as cloud deployment models. Public, private, hybrid, and community clouds are the four primary cloud deployment models. Using the internet, the general public can access computing resources that are owned and managed by outside service providers in the public cloud. Private clouds can be hosted on-premises or by external service providers and offer computing resources solely to one specific organisation. A hybrid cloud is created by combining the infrastructure of public and private clouds. Organisations that need to strike a balance between flexibility, security, and cost may find hybrid clouds to be helpful. A particular user group with similar needs or interests will all use the same cloud deployment model, which is known as the community cloud. A group of organisations or a third-party service provider can jointly own and operate community clouds.

Virtualisation as a Concept of Cloud Computing

It is possible to run multiple virtual machines (VMs) on a single physical machine thanks to the cloud computing concept of virtualization. It allows for the abstraction of computing resources, including servers, storage, and networking, from the physical hardware on which they run. This abstraction provides several benefits, including improved resource utilization, reduced costs, and increased flexibility. Each virtual machine (VM) in a virtualized environment runs its own operating system and applications just like a separate physical computer would. With each VM operating independently of the others, this enables numerous VMs to share the same physical resources like CPU, RAM, and storage. Because it lets the setting up of virtualized environments that can be scaled up or down rapidly and efficiently, depending on demand, virtualization is a crucial part of cloud computing. It also enables organizations to consolidate their IT infrastructure, reduce hardware costs, and simplify management and maintenance. By offering virtualized resources, cloud providers can offer their customers the ability to create and manage their own virtual machines, and to run their own operating systems and applications within the virtualized environment.

Task Scheduling in Cloud

Task scheduling in cloud computing is the process of assigning computational tasks to available computing resources in a cloud environment. It is an essential component of cloud computing that enables efficient utilization of resources and ensures that tasks are completed in a timely and cost-effective manner. Cloud task scheduling can be challenging due to the dynamic nature of cloud computing environments. The availability of computing resources can change rapidly, and the workload can vary significantly, making it difficult to ensure that tasks are completed on time and within budget. Efficient task scheduling is essential for achieving optimal performance and cost savings in cloud computing environments. To address these challenges, various scheduling algorithms have been developed. These algorithms aim to optimize task scheduling by considering factors such as task dependencies, resource availability, and resource utilization. By effectively scheduling tasks, cloud providers can ensure that resources are used efficiently, and customers can complete their workloads in a timely and cost-effective manner.

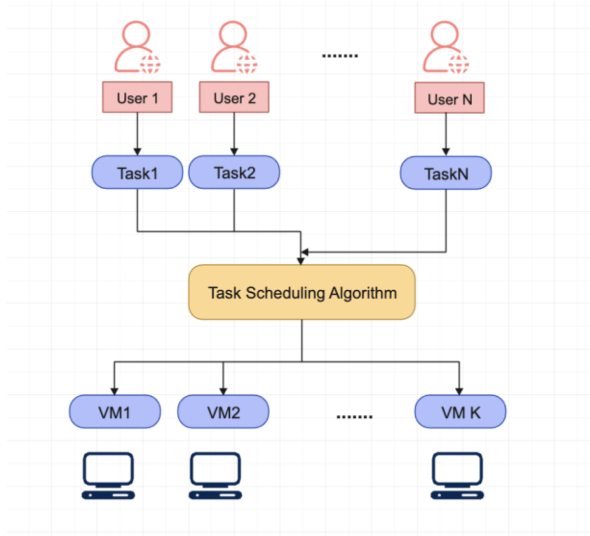

Fig. 2. Cloud Computing Task Scheduling Schematic Diagram

BACKGROUND AND RELATED WORK

The process of allocating the set of tasks submitted by user to the available VMs for execution is known as task scheduling. The tasks scheduling in a cloud environment falls under the heading of NP-hard problem. Sincere attempts have been made to investigate an effective solution to this issue, but since cloud computing is still a comparatively new subject, there is still need for fresh research.

Arora and Anand [5] in 2014 have proposed a custom batch algorithm based on Shortest First Job (SFJ) technique. This strategy has proved to be more efficient over traditional scheduling algorithms i.e. reduced waiting time and processing time with optimum resource utilisation and minimum overhead for the same.

Alworafi et al. [4] in 2016 have integrated Shortest Job First algorithm and Round Robin algorithm. The outcomes of this proposed method have been better in regards to average wait time, context switches that are needed, and average turnaround. As the processes with less remaining CPU burst time are given higher priorities, the problem of starvation has been significantly reduced.

Alhaidari and Balharith [3] in 2021 has developed improved Shortest First Job algorithm. The main concept behind this algorithm is that jobs with long execution times will be sent to virtual machines with high configurations, and jobs with short execution times will be sent to machines with low configurations. By doing so, they are able to maximise resource utilisation while minimising the average response time and makespan for the final task. This algorithm outperforms Shortest First Job algorithms and First Come First Serve.

Derakhshan and Bateni [6] in 2018 have integrated MAX-MIN and MIN-MIN algorithms. Additionally, they employ the prioritisation method to set priorities and the median number to determine whether to apply the min-min or max-min strategy. They took into account priority-based work scheduling, which is something that most researchers overlook. As a result, they cut down on task wait times and makespan. The authors of this research have utilised the Max-Min and Min-Min algorithms to create a new effective approach.

According to studies inspired by nature, complex problems can be resolved through the cooperation of groups of related agents operating in the same environment. So many studies are centred on bio inspired nature when it comes to task scheduling.

Tawfeek et al. [8] in 2013 have used a random optimisation search method called Ant Colony Optimisation is used to assign incoming jobs to virtual machines. This approach mimics how natural ant colonies behave in pursuit of food and communicate with one another by leaving pheromones along their courses of movement. This method reduces makespan by assigning requested jobs to the most potent VMs. Compared to First Come First Serve and Round Robin algorithm, this technique has shown satisfactory results.

Gitanjali et al. [7] in 2018 have presented task scheduling strategy based on Bat's algorithm. This method is based on how bat animals use sonar waves during echolocation to detect prey, avoid obstructions, etc. The purpose of this study was especially to remove problems with First Come First Serve and Round Robin algorithms. The outcomes demonstrate how stable this method is under high loads.

Tawfik et al. [9] in 2015 have focused on using Artificial Bee Colony (ABC) algorithm for task scheduling. The ABC algorithm is based on how bees forage for nectar in the wild. Using this technique, they were able to shorten the makespan of the given set of tasks. The results showed that it outperformed Ant Colony Optimisation and the Fastest Processor to Largest Task First technique.

Agarwal and Srivastava [2] in 2019 have presented Particle Swarm Optimisation algorithm for task scheduling. This technique is based on swarm intelligence and uses the information-sharing capabilities of individual birds in bird groups to find the best answer. They suggested a method to quantify inertia in order to prevent pointless research and convergence at local best, which strikes a balance between local and global search. The proposed PSO outperforms the greedy and genetic algorithms. As complexity rises, the proposed algorithm's performance gets better and better. It has boosted resource utilisation by 20% while reducing makes-an by 30%. When used in various environments, such as grid computing and fog computing, this method has proven to be reliable.

Agarwal and Srivastava [1] in 2018 have published a task scheduling strategy based on Cuckoo Search Algorithm. Laying eggs in the nest of the other host bird is an aggressive reproductive tactic used by birds, particularly cuckoo birds. This allowed them to reduce response times and task durations, and they were able to outperform the First Come First Serve and Greedy algorithms in this regard.

Further research is needed to develop more efficient and effective task scheduling algorithms that can fully optimize cloud resources. This will improve the performance and efficiency of cloud computing systems, and help to meet the increasing demands of cloud users.

ANT COLONY OPTIMISATION ALGORITHM

Ant Colony Optimization (ACO) is one of the metaheuristic algorithms that is inspired by the behavior of ants in nature. In order to determine the shortest path from their nest to a food source, ants communicate with one another through chemical signals called pheromones. The concept of ACO is to mimic this ant behaviour in order to solve optimisation problems.

In the ACO algorithm, ants are placed in a problem space and each ant builds a solution by choosing a path based on the pheromone trails deposited by other ants. The probability of picking a path depends on its pheromone concentration as well as the distance between the current position and the final destination. The routes that ants follow are covered with pheromone as they move. The amount of pheromone deposited directly affects the quality of the solution obtained. Over time, the pheromone concentration on the shortest path increases, making it more likely that future ants will follow this path.

One of ACO's primary advantages is its capacity to quickly identify effective solutions, even for complex issues with numerous variables and constraints. However, one of the limitations of ACO is that it can get trapped in local optima, leading to suboptimal solutions.

ARTIFICIAL BEE COLONY OPTIMISATION ALGORITHM

The Artificial Bee Colony algorithm is a swarm intelligence optimization technique that was inspired by the foraging behavior of honey bees. The algorithm is based on the behavior of three types of bees: employed bees, onlooker bees, and scout bees. Employee bees are tasked with searching the search space and locating food sources that could possibly be used to address the optimisation issue. Onlooker bees observe the waggle dance of the employed bees, which indicates the quality of the food sources, and choose which source to visit based on its attractiveness. Scout bees explore new areas of the search space that have not been explored by the employed bees.

The ABC algorithm starts by randomly generating a population of employed bees, which are assigned to search for solutions within the search space. The fitness of each employed bee's solution is calculated and best solution is stored as the global best solution. Onlooker bees then choose which employed bee to follow based on the quality of its solution. This is determined by the probability of the employed bee's fitness compared to the fitness of all employed bees. The onlooker bee then generates a new solution based on the employed bee's solution and its position in the search space.

If an onlooker bee finds a better solution than its employed bee, it becomes the new employed bee. Otherwise, the employed bee stays in its current position. Scout bees are also introduced to the algorithm to prevent it from getting stuck in local optima. These bees explore new areas of the search space that have not been explored by the employed bees and can potentially find new global best solutions.

Until a stopping criterion, such as a maximum number of iterations or a convergence criterion, is satisfied, the ABC algorithm keeps running. The final solution is the global best solution found throughout the iterations. It has been shown to be a competitive optimization algorithm compared to other metaheuristic algorithms.

PROPOSED METHODOLOGY

A Hybrid Metaheuristic Algorithm is an optimization algorithm that combines two or more different metaheuristic algorithms to overcome their respective limitations and improve the overall performance of the algorithm. Metaheuristic algorithms are search algorithms that are used to solve optimization problems where the objective is to find the best solution among a large set of possible solutions. Metaheuristic algorithms do not guarantee finding the global optimum solution, but they are capable of finding high-quality solutions in a reasonable amount of time.

There are various approaches to designing a hybrid metaheuristic algorithm. One approach is to use two or more algorithms in a cooperative way, where one algorithm generates a set of candidate solutions, and the other algorithm evaluates and improves them. Another approach is to use a sequential approach, where one algorithm is used to improve the solutions generated by the other algorithm.

These algorithms have shown promising results in terms of solution quality and convergence speed compared to using each algorithm independently. However, designing an effective hybrid metaheuristic algorithm requires careful consideration of the characteristics of each algorithm and the problem being solved.

A. Hybrid ACO-ABC Algorithm

In this research, we introduce the Hybrid ACO-ABC metaheuristic algorithm that combines two well-known optimization algorithms: Ant Colony Optimization and Artificial Bee Colony Algorithm. This hybridization aims to leverage the strengths of both algorithms and overcome their respective limitations. In the Hybrid ACO-ABC Algorithm, the ACO algorithm is used to construct candidate solutions, while the ABC algorithm is used to evaluate and update these solutions. Specifically, the ACO algorithm constructs a set of candidate solutions by depositing pheromone trails on a graph or a network. The ABC algorithm then evaluates the candidate solutions based on a fitness function, and the best solutions are selected to update the pheromone trails. ACO might become stuck in an alternative local optimal solution to the global optimal one. In order to avoid falling into the trap of a local optimum solution by ACO and to quickly find the global

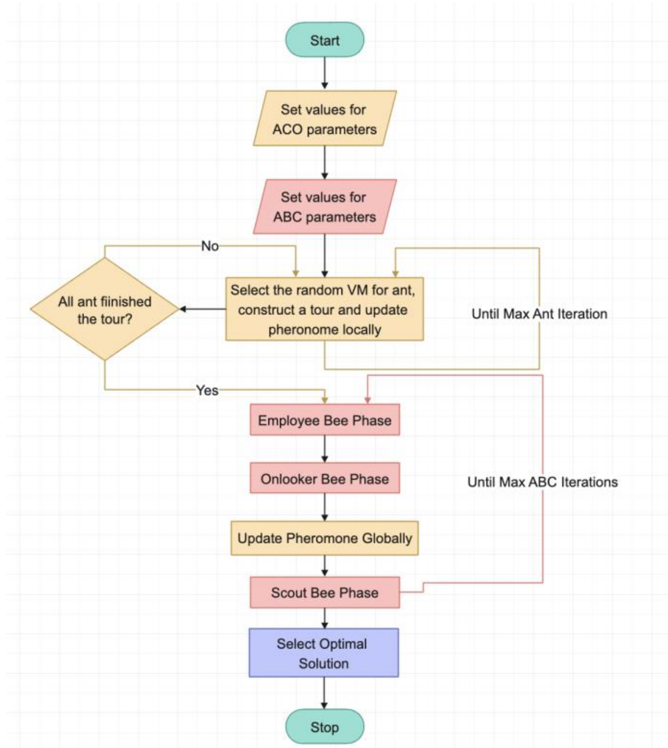

Fig. 3. Hybrid ACO-ABC Algorithm

optimal solution, the ABC algorithm is used. The main goal of hybrid ACO-ABC algorithm in task scheduling is to reduce makespan of tasks leading to improve the cloud performance.

CLOUDSIM

CloudSim is a framework for modeling and simulation of cloud computing infrastructures and services. It can be used to represent a big scaled cloud platform's data centers, service brokers, hosts, VMs, allocation, and scheduling policies. It offers fundamental classes that describe data centres, virtual machines, applications, users, computational resources, and management rules for various system components.

Core Components in CloudSim

- **Cloud Information Service**: It is responsible for maintaining information about the current status of the cloud environment like the status of hosts, VMs and data centers. The Cloud Information Service acts as a central registry for information that is needed by other entities in the simulation, such as the Datacenter Broker and the VM scheduler.
- **Datacenter**: This class represents a data center in a cloud computing environment. It includes information such as the number of host machines, the amount of storage, and the network topology.

- **Datacenter Broker**: It represents an entity that acts as a mediator between cloud consumers and providers ina cloud environment. The Datacenter Broker is meant for managing the allocation of cloud resources, such as virtual machines and cloudlets, to cloud consumers.
- **Host**: This abstract class represents a physical machine in a data center. It includes information like the CPU core numbers, amount of RAM and the amount of storage.
- **Vm**: This class represents a virtual machine in a cloud environment. It includes information such as the amount of CPU, RAM, and storage allocated to the VM, as well as the VM's ID and the ID of the host machine it is running on.
- **Vm Allocation**: This class represents a policy for allocating virtual machines to host machines in a data center. It includes methods for selecting a host machine based on available resources and load balancing.
- **Vm Scheduler**: This abstract class represents a policy to allocate and manage virtual machines on physical hosts. The class provides a set of methods for allocating and deallocating VMs, as well as for determining the status of the VMs and hosts.
- **Cloudlet**: This class represents a task that is executed on a virtual machine in a cloud environment. It includes information such as the length of the task and the amount of CPU and RAM required.
- **Cloudlet Scheduler**: This class represents a policy for scheduling cloudlets on virtual machines. It includes methods for allocating resources to cloudlets, such as time-shared and space-shared scheduling.

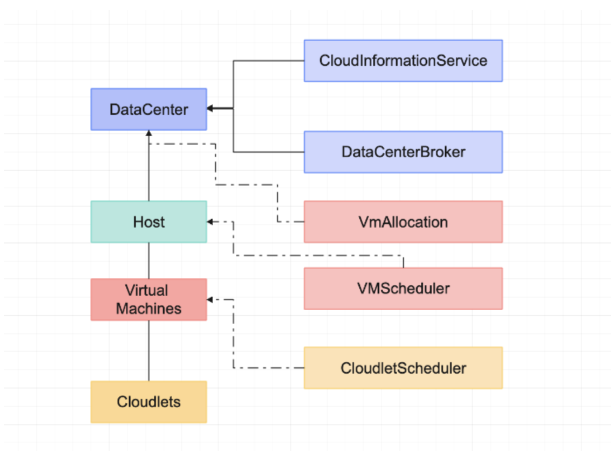

Fig. 4: Core CloudSim Components and their Relationship

Simulation Execution Steps

Launching CloudSim

Creating datacenters and service broker to coordinate and allocate resources and create virtual machines

1) Adding VMs to the virtual machine list
2) Submitting the virtual machine list to the service broker
3) Creating cloudlets and adding them to the cloudlet list
4) Submitting the cloudlet list to the service broker
5) Run the simulation and obtain results

IMPLEMENTATION AND RESULT DISCUSSION

We considered that tasks are independent from one another, that is, there is no restriction on priority between them, they are not pre-emptive and their execution cannot be halted or transferred to another processor.

CloudSim Parameters

TABLE I

CLOUDSIM PARAMETERS FOR SIMULATION

Hybrid ACO-ABC Algorithm Parameters

Entity Type	Parameters	Parameters Value
Datacenter	Architecture Operating System Virtual Machine Manager Number of Hosts	x86 Linux Xen 5
Cloudlet	Million Instructions Tasks Output File Size Input File Size Number of Processors	11000 - 12000 10 - 1000 1000 1000 1
Virtual Machine	MIPS Number of Virtual Machines Number of Processors Bandwidth RAM	1000 - 5000 5 1 1000 512

TABLE II: HYBRID ACO-ABC ALGORITHM PARAMETERS

Parameters	Parameters Value
Ants Count	10
α	0.3
β	0.1
ρ	0.4
Number of Employee Bees	5
Number of Onlooker Bees	5
Limit for Abandoning Food Source	100
Max Ant Iterations	100
Max ABC Iterations	10000

where :

α = Parameter that determines the relative importance of the pheromone trail and the heuristic information in the decision of the ants

β = Parameter that controls the relative importance of the heuristic information and the distance between two VMs in the decision of the ants

ρ = Parameter that determines the evaporation rate of the pheromone trail

The average makespan of the hybrid ACO-ABC algorithms has been when compared to the makespans of the Artificial Bee Colony Optimisation and Ant Colony Optimisation algorithms with different task sets in the graph below. In Fig. 5, the average makespan of the ABC, ACO, and hybrid ACO-ABC algorithms are displayed. As seen in Fig. 5, the hybrid ACO-

ABC algorithm completes the given task set in comparably less time (ms) than ACO and ABC. This suggests that the proposed algorithm outperforms ACO and ABC algorithms. The primary cause of this improvement is the addition of the ABC algorithm, which prevents ACO from falling victim to the trap of a local optimum solution.

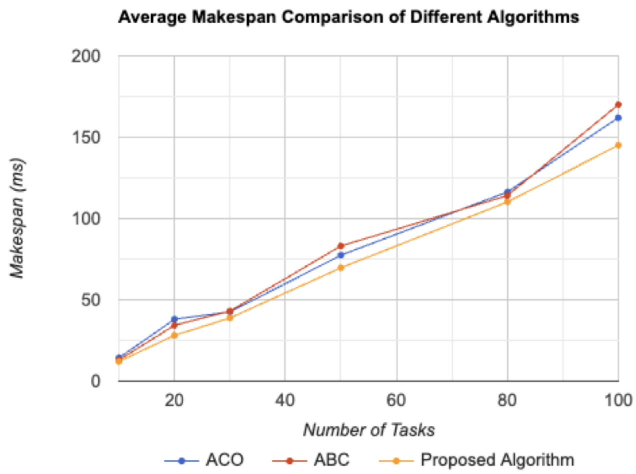

Fig. 5. Average Makespan Comparison between ACO, ABC and hybrid ACO-ABC

CONCLUSION AND FUTURE WORK

This paper provides a concise introduction about various concepts of cloud computing and how it can be implemented using CloudSim for enhancing cloud performance. It goes on to discuss how scheduling is crucial to the effectiveness of cloud systems. It gives an insight about the research work proposed in the field of cloud computing. Also, several work scheduling methods were studied in order to better grasp the idea. It goes into further detail on CloudSim and its core components, which provide an environment that can be used to directly mimic cloud computing scenarios.

This paper proposes bio-inspired hybrid meta-heuristic algorithm based on Ant Colony Optimisation and Artificial Bee Colony Optimisation algorithms. The core objective behind implementing a bio inspired hybrid ACO-ABC technique was to reduce the makespan of tasks set. As hybrid ACO-ABC algorithm takes less time to schedule given tasks set to VMs, the hybrid ACO-ABC algorithm outperfoms ACO and ABC Optimisation algorithms.

In future prospects, we will be considering priority and sequence based tasks scheduling and making proposed algorithm multi-objective by considering other objectives like cost minimisation, etc.

REFERENCES

[1]. Mohit Agarwal and Gur Mauj Saran Srivastava. A cuckoo search algorithm-based task scheduling in cloud comput- ing. In Sanjiv K. Bhatia, Krishn K. Mishra, Shailesh Tiwari, and Vivek Kumar Singh, editors, *Advances in Computer and Computational Sciences*, pages 293–299, Singapore, 2018. Springer Singapore. ISBN 978-981-10- 3773-3.

[2]. Mohit Agarwal and Gur Mauj Saran Srivastava. A pso algorithm-based task scheduling in cloud computing. In Kanad Ray, Tarun K. Sharma, Sanyog Rawat, R. K. Saini, and Anirban Bandyopadhyay, editors, *Soft Computing: Theories and Applications*, pages 295–301, Singapore, 2019. Springer Singapore. ISBN 978-981-13-0589-4.

[3]. Fahd Alhaidari and Taghreed Zayed Balharith. Enhanced round-robin algorithm in the cloud computing environ- ment for optimal task scheduling. *Computers*, 10(5), 2021. ISSN 2073-431X. doi: 10.3390/computers10050063.

[4]. Mokhtar A. Alworafi, Atyaf Dhari, Asma A. Al-Hashmi, A. Basit Darem, and Suresha. An improved sjf schedul- ing algorithm in cloud computing environment. In *2016 International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT)*, pages 208–212, 2016. doi: 10.1109/ICEEC-COT.2016.7955216.

[5]. Sumit Arora and Sami Anand. Improved task schedul- ing algorithm in cloud environment. *International Jour- nal of Computer Applications*, 96:7–12, 06 2014. doi: 10.5120/16772-6342.

[6]. Majid Derakhshan and Zohreh Bateni. Optimization of tasks in cloud computing based on max-min, min- min and priority. In *2018 4th International Conference on Web Research (ICWR)*, pages 45–50, 2018. doi: 10.1109/ICWR.2018.8387236.

[7]. J. Gitanjali, E. Sathiyamoorthi, A. Mummadisetti, R.S. Dammu, and R.R. Kadari. Scheduling tasks using bat's algorithm in cloud sim. *International Journal of Pure and Applied Mathematics*, 119:475–483, 01 2018.

[8]. Medhat A. Tawfeek, Ashraf El-Sisi, Arabi E. Keshk, and Fawzy A. Torkey. Cloud task scheduling based on ant colony optimization. In *2013 8th International Conference on Computer Engineering Systems (ICCES)*, pages 64–69, 2013. doi: 10.1109/ICCES.2013.6707172.

[9]. Medhat Tawfik, Ashraf Bahgat, Arabi Keshk, and F. Torkey. Artificial bee colony algorithm for cloud task scheduling. *IJCI. International Journal of Computers and Information*, 4(1):1–10, 2015. ISSN 1687-7853. doi:10.21608/ijci.2015.33956.