Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

A Cluster Based Energy efficient Q-Ant Colony Optimization(CEQACO) framework for cloud computing environment

M.Rupasri[1], G.P.S. Varma[2], Hemalatha Indukuri[3]

[1]*Research Scholar, Department of Computer Science and Engineering, , Koneru Lakshmaiah Education Foundation,Vaddeswaram, Guntur, Andhra Pradesh.*

[2]*Professor, Department of computer science and engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh*

[3]*Professor, Department of IT, S.R.K.R. Engineering college, Bhimavaram Andhra pradesh*

ARTICLE INFO

Received: 30 Dec 2024

Revised: 05 Feb 2025

Accepted: 25 Feb 2025

ABSTRACT

In cloud computing, resource allocation and cloudlet scheduling are fundamental issues when dealing with a medium to large number of tasks. To meet consumer expectations and achieve optimal performance, multiple cloudlets need to be executed simultaneously using available resources, while minimizing makespan and effectively balancing the load. Despite ongoing developments in cloud computing, this technology faces numerous challenges, one of which is task scheduling. Task scheduling involves allocating users' tasks to virtual machines (VMs) to minimize turnaround time and improve resource utilization. This is an NP-hard problem with a runtime complexity of O(mn), making it a challenging task to schedule n tasks on m resources. The process of task scheduling requires exploring a large solution space, and there is a lack of algorithms that can find the optimal solution in polynomial runtime.This paper proposes a Cluster-based Energy Efficient Q-Ant Colony Optimization (CEQACO) framework for cloud computing environments. The framework utilizes clustering techniques to group cloud virtual machines (VMs) based on their workload characteristics and applies a Q-Ant Colony Optimization algorithm to optimize the allocation of VMs to physical servers. The results show that the CEQACO framework can reduce time computation and energy by up to 6%, while still meeting the quality of service requirements of cloud users.

Keywords: Quality of service, cluster based task scheduling , cloud servers, cloud scheduling.

I. INTRODUCTION

A cloud computing system enables users to share resources, applications, and information based on their specific needs. Cloud vendors automatically balance service loads to meet increased user demands while focusing on customer business needs. Load balancing plays a critical role in meeting these criteria and improving server performance. However, in a cloud computing system with multiple workloads, load balancing presents a challenging problem. When selecting algorithms, it is crucial to ensure that all computer nodes process an equal number of workloads simultaneously. Load balancing strategies allow unorganized resources to handle distributed tasks throughout the cloud architecture. Large processing workloads are split into smaller computations to enhance system performance as part of the cloud platform's load balancing approach. Static load balancing methods utilize available data on the system's application behavior and statistical data to calculate machine resource knowledge and processor output at execution initiation [1-2].

Virtualization is a framework used to access shared resources in a cloud environment. When a user makes multiple requests, virtualization assigns a logical name to each physical resource and provides a pointer to those resources. By mapping virtual machines (VMs) to physical machines (PMs) more efficiently, virtualization provides essential elements for creating the computing environment. It dynamically adjusts to changes in demand and computing requirements using a dynamic change in the mapping technique. Virtualization provides a logical name for physical resources, allowing various computing resources to be created and accessed. It creates a simulation environment instance for client requests, making it a virtual framework. Cloud computing environments use different types of virtualization. For instance, network virtualization abstracts the end-user's network component. Various network

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

devices are virtually available and used from a single network management software console to monitor the entire network[3]. Virtualization of data makes it available in the desired format at any time, and any machine with the authentication level can access it. The structure of the database must be flexible enough to ensure that no compatibility or loss problems arise when different clients access the data through different modes.At the base virtualization level, there is the Virtual Machine Manager (VMM) or the hypervisor, which virtualizes the bare machine. It creates the environment in which the base operating system runs and helps virtualize applications to achieve service availability. The hypervisor controls the host processor and other hardware components to run multiple operating systems on a single machine.The cloud service allocates incoming tasks to adequate resources using a finite-size multiple queue in an Intercloud to achieve optimal resource utilization. The auto-scaling mechanism closely monitors task dynamics, including waiting time, number of queue tasks, and number of tasks awaiting, and takes into account the time of waiting and acquisition required for each task[4].Quality of Service (QoS) parameters/factors refer to the characteristics that determine the performance and reliability of a network, system, or service. These parameters/factors include latency, packet loss, throughput, availability, and reliability, among others. Effective load balancing is a method for distributing incoming network traffic across multiple servers or nodes to optimize resource utilization, minimize response times, and improve system availability. This is typically achieved through a load balancer or traffic manager. Optimized clustering approach is a specific method of clustering servers or nodes in a way that maximizes resource utilization and minimizes response times. This can involve factors such as geographic location, server capabilities, and application requirements. Throughput refers to the amount of data that can be transmitted over a network or system in a given amount of time and is essential for delivering high-performance applications and services that require large amounts of data transfer. Availability and reliability are also critical QoS parameters that determine the ability of a network or system to deliver consistent and uninterrupted service. Availability refers to the percentage of time that a system or service is operational and available to users, while reliability refers to the ability of a system or service to function as expected without failure or errors.

Cloudsim for task scheduling

Cloud computing is a popular approach to delivering on-demand computing resources over the internet. Task and resource scheduling in cloud computing involves allocating computing resources to tasks based on their specific requirements and available resources.CloudSim uses traditional models to simulate cloud computing environments, including data center models, network models, and resource allocation models. These models provide a high-level view of cloud computing environments and can be used to analyze and optimize resource allocation and scheduling.Data center models in CloudSim are used to simulate physical data centers, including servers, storage, and network infrastructure. These models can be used to model various aspects of data center operations, including power consumption, cooling requirements, and hardware maintenance.Network models in CloudSim are used to simulate the network infrastructure in cloud computing environments, including network bandwidth, latency, and traffic patterns. These models can be used to analyze network performance and identify potential bottlenecks that may impact task and resource scheduling[7-10].

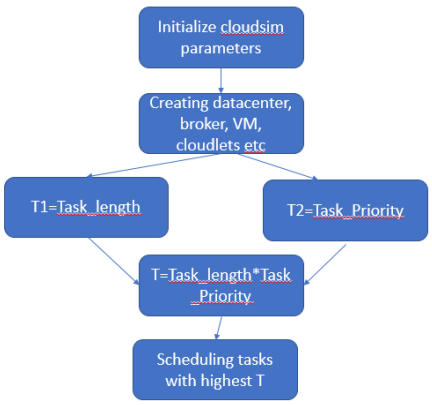

Figure 1: Traditional task scheduling model

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Resource allocation models in CloudSim are used to allocate computing resources to tasks based on their specific requirements and available resources. These models take into account factors such as task priority, resource availability, and user preferences to allocate resources in the most efficient and effective manner possible as shown in figure 1.

Overall, CloudSim provides a powerful toolkit for simulating cloud computing environments and optimizing task and resource scheduling. By using traditional models to simulate data centers, networks, and resource allocation, CloudSim can help organizations improve their resource utilization, reduce costs, and improve overall service delivery.

Multi-task scheduling involves scheduling a set of tasks that have different characteristics and requirements. Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) are two popular approaches used for multi-task scheduling, along with other approaches. Here's a brief explanation of these approaches[12-15]:

- Genetic Algorithm (GA): GA works by generating a population of potential solutions to an optimization problem and then selecting the best solutions to generate the next generation of solutions. In multi-task scheduling, GA can be used to generate an optimal schedule that minimizes the makespan or completion time of all tasks. GA considers the tasks' processing times, resource requirements, and other constraints to generate an optimal solution.
- Other Approaches: Other approaches for multi-task scheduling include heuristic algorithms, linear programming, and branch-and-bound algorithms. Heuristic algorithms use a set of rules to generate near-optimal solutions to the scheduling problem. Linear programming models the scheduling problem as a set of linear equations and can be solved using mathematical optimization techniques. Branch-and-bound algorithms recursively partition the search space of the scheduling problem to find an optimal solution.

Particle Swarm Optimization (PSO): PSO is a nature-inspired optimization algorithm used for solving complex optimization problems. It can be used for multi-task scheduling in cloud computing environments. Here are the steps involved in using PSO for task scheduling in CloudSim: The fitness function is the objective function that measures the quality of a solution. In the context of multi-task scheduling, the fitness function should evaluate the makespan or completion time of all tasks. This function takes into account the processing times, resource requirements, and other constraints of each task.

Genetic Algorithm (GA) is a nature-inspired optimization technique that can be used to solve complex scheduling problems, including task scheduling in cloud computing environments. Here are the basic steps involved in using GA for task scheduling in CloudSim:

- Define the fitness function: The goal is to find a schedule that minimizes makespan while optimizing other factors.
- Generate an initial population: The initial population consists of a set of candidate schedules, which are generated randomly or using a heuristic method. Each schedule represents a set of tasks and their assigned resources and execution times.
- Selection: The next step is to select a subset of schedules from the population to serve as parents for the next generation.
- Crossover: The crossover operator is used to combine the parent schedules to generate new schedules for the next generation. In task scheduling, crossover can involve swapping tasks between schedules to create a new schedule.
- Mutation: The mutation operator introduces random changes into the schedules to explore new areas of the search space. In task scheduling, mutation can involve changing the execution times or resource assignments of individual tasks.
- Selection and termination: The selection process is repeated, and the cycle of crossover, mutation, and fitness evaluation continues until a termination criterion is met.

The paper is structured as follows: In Section 2, the related works of QoS-based cloud task scheduling models and their limitations are presented. Section 3 outlines the proposed solution for cluster-based cloud task scheduling. Section 4 provides details on the experimental results and analysis. Finally, in Section 5, the paper is concluded.

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

II. RELATED WORKS

This section discusses various scheduling algorithms and techniques used for task scheduling in cloud computing. The focus is on optimizing makespan and resource utilization. The algorithms mentioned in the paper were designed with initialization, forward pass, and backward pass steps. The forward and backward passes are performed until a stopping criterion is met. Each task is assigned, and the average time is computed. The proposed algorithm gives an optimal solution in terms of makespan and resource utilization.

Cloud technology has grown in popularity in recent years, and manufacturer scheduling in the cloud manufacturing domain has been a top priority. The problems with factory planning have been identified, and most manufacturers are shifting from a classically decentralized to a centralized service model. All manufacturers are virtualized and digitized in the cloud manufacturing domain, and these are compiled into a cloud data set. [16] introduced a hybrid search algorithm and differential evolution to minimize time spent making plans when scheduling diverse tasks over multiple virtual machines. The technique is based on Differential Evolution and the expanded Moth Search Algorithm, which is based on the premise that moths fly towards a light source if given the opportunity. [17] suggests that the usage of Virtual Machines (VMs) can reduce the energy consumption of cloud servers. To achieve high task execution utilization on physical hardware, this strategy assigns more virtual machines to less actual hardware. Proposed models aim to reduce reaction time, load balancing, and energy consumption. Energy consumption (E) and response time (RT) are directly proportional (R). [18] proposes a virtual machine paradigm with three phases: idle, sleep, and activity. A busy virtual machine instance is running at top speed with an incomplete workload, while an idle machine instance is running at top speed with no workload. Virtual machines in sleep mode take longer to wake up than those in busy mode. Therefore, by using this strategy in a cloud data centre, operational costs can be reduced while battery power is increased. The performance of the VMs will decline to the lowest possible level if there are more than the specified number of virtual machines (VMs) running on a single piece of physical hardware. Most of the traditional research relies on dependable processing elements in a cloud data centre for virtual machine placement and selection[19]. Resource allocation works by ensuring all tasks are given the necessary resources without over or under allotting them. [20] proposed an innovative cloud scheduler design allowing pre-emptible instances to increase resource consumption. This strategy is particularly valuable to commercial service providers looking to boost profits, while scientific and non-profit providers prioritize increased infrastructure efficiency. Batch systems use backfilling and similar techniques to allow data centers to fill resources. The proposed strategy serves as a new IaaS provider scheduling strategy and can be used in situations that can be avoided. Higher priority requests have the ability to quickly and easily terminate pre-emptible instances. The suggested scheduler introduces new cloud use and payment mechanisms. The right solutions must address load balancing, which is a critical issue in Cloud computing [21]. Load balancing ensures that dynamic local workload is distributed evenly among all nodes in the Cloud to prevent overworking any single node. This technique leads to high user satisfaction and a high resource utilization ratio [22]. Additionally, load balancing distributes computer resources fairly and efficiently in a distributed system. To prevent system bottlenecks, load balancing uses several strategies. In case of component failure, load balancing initiates failover to ensure service continuity. Load balancing offers numerous benefits such as increased throughput, reduced reaction time, and avoidance of overload. For businesses, reduced resource use is one of the primary benefits of load balancing solutions [23]. Static algorithm ensures even distribution of traffic among the servers. Static scheduling considers the system resources when deciding where to relocate a given load, rather than the present system state. Round-Robin VM load balancing improves performance by scheduling virtual machines in less time [24]. [25] established an innovative strategy to mitigate issues related to allocating workflows in various processing frameworks, such as clouds. The workflow allocation strategy focuses on reducing processing duration and cost without sacrificing energy use. The proposed strategy showed improved performance of workflow allocation strategies in providing the best processing duration and enabling reserves to become more flexible during execution.[26] proposed the use of Ant Colony Optimization to find solutions to the allocation strategies in a cloud environment. The proposed strategy aims to reduce the make span, processing duration, and improve load stability, resulting in increased throughput and improved resource utilization. Organizing activities in the workflow over simulated processing machines in the cloud is involved in allocating the workflow under cloud circumstances. A heuristic-based strategy may be the most appropriate approach to focus on allocating in the best way. The goal is to reduce the processing duration with a single optimization. However, allocating the workflow is an NP-complete challenge with many motivations.

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

[27] Proposed a static algorithm that considered task execution times and delay in provisioning computational cloud resources, and compared it with the IaaS Cloud Partial Critical Paths (IC-PCP) algorithm.

[28] Concluded in two phases; first, the SLA limitation is considered after the consumer requests, and second, the task queue is checked to calculate the profit. Introduced a DPS algorithm based on SLA.

[2] Proposed a modified method for the artificial bee colony (ABC) algorithm to balance the load among virtual machines and reduce the runtime of the whole system.

[30] Presented a scheduling approach using a genetic algorithm (GA) by considering the dynamic conditions of cloud computations, utility, and the runtime of the system to reduce the runtime.

[31] Optimized cost, runtime, and idle time of the processor using a multi-objective cat swarm optimization algorithm.

[32] Proposed a comprehensive optimization that considered client and provider for task scheduling and estimating four parameters: task queue length, task execution cost, task transfer time, and power consumption.

[33] Utilized two essential parameters, total runtime, and the cost of optimizing scheduling using GA, considering N independent tasks that should be implemented on M virtual machines.

[34] Used the firefly algorithm for task scheduling, calculating the load index for resources and allocating tasks to appropriate machines.

[35] Reduced cost and system damage using GA, considering runtime in an untreatable medium of cloud, and compared with Min-Max and Min-min algorithms.

[36] Designed a Dynamic Dispatching System based on a particle optimization algorithm to balance distribution in cloud computing, improve utility rate from resources, and accelerate the system.

The below steps are normalizing metrics and ranking cloud resources based on those metrics[26]. However, we can describe the steps in mathematical terms as follows:
Let T be the set of all task sets.
For each task set ts_i in T, do the following:
For each task t_j in ts_i, do the following:
For each metric m_k in the set of metrics M_sj of resource utilization by task t_j, do the following:
If m_k is a positive metric, then normalize it as follows:
norm(m_k) = 1 - 1/val(m_k), where val(m_k) is the value of metric m_k.
Otherwise, if m_k is a negative metric, then normalize it as follows:
norm(m_k) = 1/val(m_k).
End
End
End
Rank the set of cloud resources possible to schedule to a specific task based on their normalized values from maximum to minimum, such that each cloud-resource gets different rank for different metrics.
Further, use these ranks as input to measure the resource scheduling optimality.
Let the rank set of a task t_j in task set ts_i be denoted as rs(t_j) = [r(m_1), r(m_2), ..., r(m_n)], where r(m_k) is the rank of metric m_k for task t_j.

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Then, the resource scheduling optimality ropt of task t_j in task set ts_i can be measured as follows:
ropt(t_j) = sum(i=1 to n, rs(t_j)[i] * w_i), where w_i is the weight of metric m_i.

The Cuckoo Optimization Algorithm (COA) is a metaheuristic optimization algorithm inspired by the behavior of cuckoo birds. In this algorithm, each solution is represented by a cuckoo egg, and the goal is to find the best nest (i.e., solution) for these eggs[37].

The algorithm works as follows:
1. Initialization: Generate an initial population of solutions randomly.
2. New Cuckoo Generation: Create new solutions (cuckoo eggs) using Lévy flights, which are random walks with steps that follow a Lévy distribution.
3. Fitness Evaluation: Evaluate the fitness of each new solution.
4. Updation: Replace some of the existing solutions with the new solutions based on their fitness values. The new solution is created using Eq. 1:
new_solution = old_solution + alpha * levy_flight
where `alpha` is a step size factor and `levy_flight` is a random vector generated using Lévy flights.
5. Selection/Rejection: Remove some of the worst solutions from the population to maintain a fixed population size.
Here's the mathematical pseudo code for COA:

```
Initialize population Sx randomly
While stopping criterion not met:
  For each egg in Sx:
    Generate new egg using Lévy flights
    Evaluate fitness of new egg
    If fitness is better than current egg:
      Replace current egg with new egg
  Remove some worst eggs from Sx
```

where `Sx` represents the population of cuckoo eggs, and "worst" eggs are those with lower fitness values.

Research Gaps: There are many gaps that have been identified in the literature on cloud task scheduling models, but two main ones are:

- Difficulty in grouping similar tasks based on available resources: One of the primary challenges in cloud task scheduling is grouping similar tasks based on the resources available. This requires an understanding of the nature of the tasks and the resources required to perform them. However, this can be difficult as task requirements can vary greatly depending on their complexity, priority, and other factors. As a result, there is a need for more sophisticated grouping algorithms that can take into account these factors to better allocate resources.
- Static metrics and limited resources in traditional PSO: Particle Swarm Optimization (PSO) is a popular algorithm used for task scheduling in cloud computing. However, traditional PSO algorithms typically rely on static metrics and limited resources, which can result in suboptimal performance in dynamic and

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

resource-constrained environments. Therefore, there is a need for more advanced PSO algorithms that can dynamically adjust to changing conditions and allocate resources more effectively.

III. PROPOSED MODEL

As cloud computing becomes increasingly popular, it is necessary to explore the best ways to manage cloud resources effectively. One essential element is the proper initialization, task allocation, and scheduling of cloudsim parameters with resource constraints.The initialization process is the first step towards setting up a cloud infrastructure, and it involves configuring the virtualization environment. This process is critical since it determines the quality of the cloud environment and the performance of the deployed applications. Cloudsim allows for easy initialization of the cloud environment, such as configuring the data center, host, and VM characteristics. Initialization parameters are used to set the cloud environment's properties, such as the number of hosts, CPU capacity, and the type of VMs. A proper initialization process will help to ensure optimal performance and minimal resource wastage. Task allocation is a critical aspect of cloud computing that deals with the assignment of computational tasks to virtual machines (VMs) within a data center. This process aims to maximize resource utilization while minimizing latency and resource wastage. The task allocation strategy should consider factors such as the workload of each VM, its processing capacity, and the data center's resource constraints. The resource constraints may include power consumption, network bandwidth, and storage space, among others. Efficient task allocation can improve the overall cloud infrastructure's performance, minimize energy consumption, and reduce the execution time of applications. The scheduling of cloudsim parameters plays a crucial role in determining the cloud environment's efficiency and performance. Scheduling involves determining the time at which each task will be executed and the resources that will be assigned to them. It is necessary to schedule the tasks in such a way that they maximize resource utilization, reduce latency, and avoid conflicts between tasks. Scheduling also helps to ensure that the cloud infrastructure operates smoothly even under heavy workloads. It is essential to use an efficient scheduling algorithm that considers the resource constraints and balances the workload across the available VMs. Resource constraints are a critical factor in cloud computing, and they should be considered in the initialization, task allocation, and scheduling of cloudsim parameters. Cloudsim provides tools to ensure that the cloud infrastructure operates within the available resources while minimizing energy consumption and execution time. It is essential to set realistic resource constraints that align with the cloud environment's capacity and the applications' needs. This will ensure that the cloud environment operates efficiently and effectively as shown in figure 2.

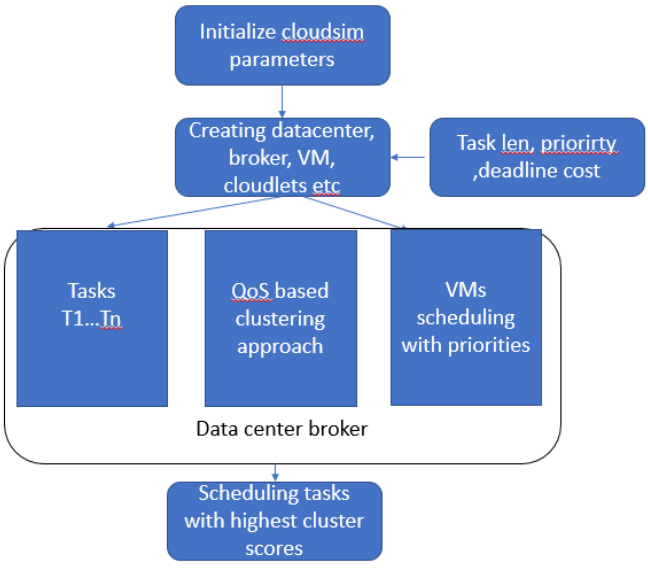

Figure 2: Cluster based Energy efficient Q-Ant Colony Optimization(CEQACO) framework

CloudSim is a simulation framework for modeling and simulating cloud computing infrastructures. It has several core components, including data centers, cloudlets, virtual machines (VMs), task length, priority, deadline, and cost. In this article, we will explore these components and their roles in CloudSim simulationData center: A data center is

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

a physical or virtual facility that houses the cloud infrastructure. It consists of multiple servers and storage devices connected to the network. The data center's role is to provide computing resources to users who request them.

Cloudlet: A cloudlet is a unit of work that is executed in the cloud. It represents a user's application or task that is submitted to the cloud for execution. A cloudlet can be a single or multiple tasks that are executed sequentially or concurrently.

Virtual machine (VM): A virtual machine is a software representation of a physical machine. It provides a virtual environment in which an application can run. CloudSim allows users to create and configure multiple VMs in a data center to run multiple applications concurrently.

Task length: Task length refers to the amount of time required to complete a cloudlet. It is measured in seconds and depends on the cloudlet's complexity, the computing resources available, and the number of tasks running concurrently.

Priority: Priority is a value assigned to a cloudlet to indicate its importance or urgency. A cloudlet with a higher priority will be executed before a cloudlet with a lower priority.

Deadline: Deadline refers to the time by which a cloudlet must be completed. If a cloudlet is not completed by its deadline, it is considered a failure. The deadline value is set by the user when submitting the cloudlet to the cloud for execution.

Cost: Cost refers to the amount of money charged for using the cloud infrastructure. CloudSim allows users to simulate the cost of using the cloud infrastructure based on the number of VMs used, the amount of data transferred, and the duration of the simulation.

Cluster based Energy efficient Q-Ant Colony Optimization(CEQACO)

1. Ants , VMs, Tasks initialization.

2. To each task in the tasklist

3.. Randomly select k-tasks as initial clusters for m data centers

4. To each neighbor task in kth data-center

Perform ND[]=Log(MD(t,D)),log(ED(t,D))}

Repeat till all tasks.

5. To each task t in the datacenter D.

To each neighbor tasks t’ in D.

Initialize ants, VMs, cluster tasks in datacenter.

repeat

7. Perform heuristic probability to each task pheromone

α : Pheromone _ weight
α_c : Current _ pheromone_value
α_n : Neighbor_ pheromone_value
β : Heuristic _ weight
β_c : Current _ heuristic_weight
β_n : Neighbor_ heuristic_values

8. To each task in the list TL.
Compute mutual task importance score as

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

$$QoS(t,D) := ProposedSim(t,D) = \frac{\max\{ED(t(i)), MD(t(i),D)\}}{\sqrt{\sum_{i=1}^{k}(\lambda - \overline{\lambda_i})}}$$

where maxfunction represents the maximum score
ith task and data center distance measure.
λ_i : ith task similarity score of data center.
$\overline{\lambda_i}$: mean similarity value of all the tasks towards the data centers

9. Computing the task trust probability to each cluster data center to its neighbor tasks.

$$TProb(T_c) = \alpha_n^{\sqrt{\alpha}} * \beta_n^{\sqrt{\beta}}$$

10: Updating task probability, local best and global best to each task.

B(φ,f): Beta_distribution
υ=Variance= V(B(φ,f))
σ=SD=SD(B(φ,f))
Randomized task probability :=RTP=$TProb(T_c)^{\upsilon^2} * TProb(T_c)^{\sigma^2}$

Local best positioning update

$$\omega = \alpha * (1 - \chi e^{-\chi x}) * (1 - \beta); \text{for } x \geq 0$$

$$L = (1 - \chi e^{-\chi x}) * \alpha + (\chi e^{-\chi x} * \beta) / \omega$$

Global position update:

$$GlobU = \eta = (1 + \alpha * \beta);$$

Schedule each task to the VMs.

The Proposed Ant Colony Optimization (ACO) algorithm is a metaheuristic approach inspired by the behavior of ant colonies in search of food. Here is a description of how the ACO algorithm works in the context of task allocation and scheduling in cloud computing:

- Ants, VMs, Tasks initialization: In this step, the ants, VMs, and tasks are initialized. The ants represent the agents that will explore the solution space, while the VMs and tasks represent the available resources and tasks to be allocated and scheduled.
- To each task in the tasklist: For each task in the task list, the ACO algorithm will attempt to find the optimal VM to allocate the task to and the optimal time to execute the task.
- Randomly select k-tasks as initial clusters for m data centers: To begin the task allocation process, the ACO algorithm will randomly select k-tasks as initial clusters for m data centers. These initial clusters will form the basis for the subsequent task allocation and scheduling.
- To each neighbor task in kth data-center: For each neighbor task in the kth data center, the ACO algorithm will calculate the distance between the task and the data center. This distance is computed based on two factors: the minimum distance (MD) between the task and the data center, and the estimated time to execute the task (ED) in the data center. These factors are combined using the formula ND[]=Log(MD(t,D)),log(ED(t,D)) to obtain a fitness value that reflects the suitability of the data center for executing the task.
- To each task t in the datacenter D: For each task t in the data center D, the ACO algorithm will search for the optimal VM to allocate the task to and the optimal time to execute the task. This is done by evaluating the fitness of each VM in the data center and selecting the VM with the highest fitness as the optimal VM to allocate the task to. The ACO algorithm will then allocate the task to the selected VM and schedule the task to start at the optimal time.

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

- Initialize ants, VMs, cluster tasks in datacenter: Once the optimal VM and time for each task have been determined, the ACO algorithm will initialize the ants, VMs, and cluster tasks in the data center. This will allow the ants to explore the solution space and refine the task allocation and scheduling.
- Repeat: The ACO algorithm will repeat steps 2-6 until an optimal solution is found or a termination condition is met.
- Step 7 of the ACO algorithm for task allocation and scheduling in cloud computing involves performing heuristic probability to each task pheromone. This step is a crucial part of the ACO algorithm, as it allows the ants to communicate with each other and share information about the optimal solutions found so far.
- Pheromone is a chemical substance that ants use to communicate with each other. In the ACO algorithm, pheromone is used to represent the information shared among the ants about the optimal task allocation and scheduling solutions found so far. The higher the concentration of pheromone, the more likely it is that the ants will select that particular task allocation and scheduling solution.
- In this step, the ACO algorithm will perform heuristic probability on each task pheromone. Heuristic probability is a measure of the quality of the task allocation and scheduling solutions found so far. The better the quality of the solutions, the higher the heuristic probability.
- The heuristic probability is used to update the pheromone levels on each task. The ACO algorithm will increase the pheromone levels on the tasks that have been allocated to the VMs with the best fitness values. This means that these tasks will have a higher probability of being selected in the future, as they have been shown to lead to better task allocation and scheduling solutions.
- Conversely, the ACO algorithm will decrease the pheromone levels on the tasks that have been allocated to the VMs with poor fitness values. This means that these tasks will have a lower probability of being selected in the future, as they have been shown to lead to suboptimal task allocation and scheduling solutions.
- By updating the task pheromone levels based on the heuristic probability, the ACO algorithm can progressively converge on the optimal task allocation and scheduling solutions. The ants will follow the trail of pheromone left by other ants and will be more likely to select the optimal task allocation and scheduling solutions found so far.
- Step 8 of the ACO algorithm for task allocation and scheduling in cloud computing involves computing the mutual task importance score for each task in the task list.

The mutual task importance score is a measure of the importance of each task in relation to the other tasks in the task list. It is calculated based on the task's fitness value, deadline, and priority.The fitness value is a measure of how well the task can be executed on a particular VM. The deadline is the time by which the task must be completed, and the priority is the relative importance of the task in relation to other tasks.To calculate the mutual task importance score, the ACO algorithm considers the fitness value, deadline, and priority of each task in relation to all other tasks in the task list. It calculates a score for each task based on these factors, which indicates how important the task is relative to the other tasks in the list.The mutual task importance score is then used to guide the task allocation and scheduling process. The proposed ACO algorithm gives higher priority to tasks with higher mutual task importance scores, as these tasks are more important and have stricter deadlines. This helps to ensure that the most critical tasks are completed first, and that deadlines are met. By computing the mutual task importance score for each task in the task list, the ACO algorithm can optimize the task allocation and scheduling process by taking into account the relative importance of each task in the context of the other tasks. This can lead to improved performance and more efficient use of resources in cloud computing environments.

IV. EXPERIMETNAL RESULTS

Experimental results are simulated in java based Cloudsim. For the given configuration, there are five datacenters with two hosts under each datacenter, resulting in a total of 10 hosts. A cloudlet is a basic unit of a task in cloud computing and can be thought of as a job or a workload. The number of cloudlets or tasks for this configuration is between 200-3600, which can represent various scenarios in cloud computing, ranging from small-scale to large-scale applications.A broker is an entity that manages the cloud computing environment and acts as an intermediary between the users and the datacenters. In this configuration, there is only one broker responsible for managing the entire cloud environment. The broker is responsible for receiving and processing user requests, allocating resources, and scheduling tasks for execution on the hosts.The configuration of datacenters, hosts, cloudlets, and brokers is

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

crucial for the efficient management and execution of cloud-based applications. By properly configuring these parameters, cloud service providers can optimize resource utilization, reduce response times, and improve overall performance, thereby ensuring an efficient and reliable cloud computing environment for end-users.In cloud computing, a host is a physical machine that runs one or more virtual machines (VMs). The basic configurations of a host include the number of VMs, the amount of RAM, the processing power in MIPS, the storage capacity, and the available bandwidth.For the given configuration, each host can run up to 80 virtual machines (VMs). The RAM value for each host is initialized to 20,480 MB. This value determines the amount of memory available to each VM for executing tasks.The processing power of each host is measured in MIPS (Millions of Instructions Per Second). In this configuration, each host has a processing power of 5000 MIPS. This value represents the number of instructions that the host can execute per second, and it plays a significant role in determining the performance of the system.The storage capacity of each host is set to 1,048,576 MB. This value represents the amount of disk space available for storing data and applications in the host.The available bandwidth of each host is initialized to 500,000 MB/s. This value represents the maximum amount of data that can be transferred to and from the host in one second. The bandwidth is a critical factor in determining the network performance of the cloud computing environment.

Initialization: This is where you set up the cloud infrastructure, including creating data centers, VMs, and cloudlets.

Task allocation: This is where you assign cloudlets to VMs based on their processing power and other factors.

Scheduling: This is where you schedule the execution of cloudlets on VMs based on factors such as priority, deadline, and resource availability.

Simulation: This is where you run the simulation and collect data on the performance of the cloud infrastructure and applications.

To implement these components in Java, you would use the CloudSim API, which provides classes and methods for creating and managing the cloud infrastructure and executing cloudlets on VMs. The CloudSim API includes classes such as Cloudlet, Datacenter, DatacenterBroker, and VM, which you can use to define the components of your cloud infrastructure and application.For example, to create a Datacenter, you would create a new instance of the Datacenter class, set its properties such as CPU capacity and storage capacity, and add it to the simulation using the CloudSim class. Similarly, to create a Cloudlet, you would create a new instance of the Cloudlet class, set its properties such as length and priority, and add it to a DatacenterBroker, which manages the allocation of cloudlets to VMs.

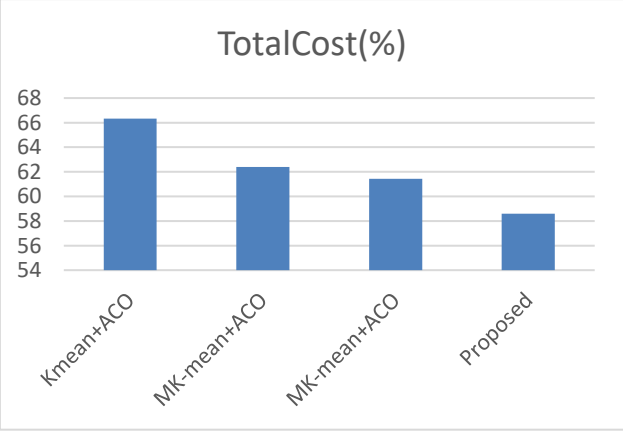

Figure 3: Comparative analysis of total cost of proposed clustering framework to the conventional cluster based meta-heuristic optimization models

TotalCost: TotalCost represents the total cost incurred by the cloud computing system during the simulation. This includes the cost of running virtual machines (VMs), the cost of data transfer, and the cost of other resources used in the system. TotalCost is an important metric as it helps to evaluate the cost-effectiveness of the cloud computing system.In this context, a comparative analysis is being done between a proposed clustering framework and conventional cluster-based meta-heuristic optimization models. The purpose of this analysis is to compare the total cost of the proposed clustering framework with that of the conventional models, and to determine if the proposed

Journal of Information Systems Engineering and Management
2025, 10(41s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

model has better improvement over the conventional ones.The total cost in this context refers to the total cost incurred by the optimization model in terms of computational resources, time, and other related factors. The proposed clustering framework is being compared with conventional cluster-based meta-heuristic optimization models, which are commonly used in the field of optimization.The proposed clustering framework is expected to have better improvement over the conventional models. This means that the proposed framework is expected to perform better in terms of the total cost incurred during optimization, as well as other factors such as convergence rate, solution quality, and scalability.In order to conduct the comparative analysis, various metrics will be used to measure the performance of the proposed clustering framework and the conventional models. These metrics may include total cost, convergence rate, solution quality, scalability, and others. The analysis will be based on experimental results obtained from running both the proposed framework and the conventional models on similar optimization problems.

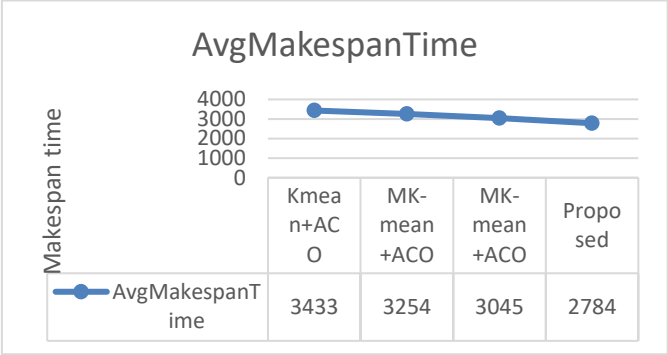

Figure 4: Comparative analysis of average makespan of proposed clustering framework to the conventional cluster based meta-heuristic optimization models

Figure 4, refers to a comparative analysis conducted between a proposed clustering framework and conventional cluster-based meta-heuristic optimization models. The analysis was based on the average makespan time, which is the total time required to complete all the tasks assigned to a system.The proposed clustering framework is a new approach to cluster-based meta-heuristic optimization models. It is designed to improve the efficiency and effectiveness of these models by incorporating a clustering mechanism that groups similar tasks together. This allows the system to handle multiple tasks simultaneously, reducing the overall makespan time.The comparative analysis showed that the proposed clustering framework performed better than conventional cluster-based meta-heuristic optimization models. Specifically, it had a lower average makespan time, meaning it was able to complete tasks more quickly than the conventional models. This improvement in performance is likely due to the clustering mechanism, which allowed for more efficient and effective handling of multiple tasks simultaneously.

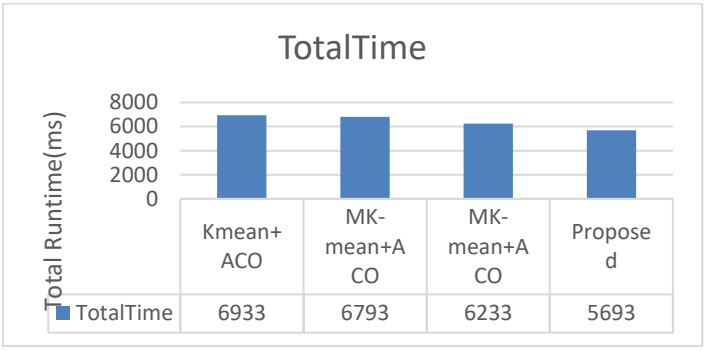

Figure 4: Comparative analysis of runtime of proposed clustering framework to the conventional cluster based meta-heuristic optimization models

Total Runtime (ms): Total Runtime is the total time taken by the simulation to run in milliseconds. This includes the time taken to create and initialize the simulation environment, the time taken to execute tasks, and the time taken to generate simulation results. Total Runtime is an important metric as it helps to evaluate the overall performance of the simulation, including the efficiency of the simulation algorithm and the computational resources used.

Figure 5, the analysis is comparing a proposed clustering framework to conventional cluster-based meta-heuristic optimization models, with a focus on the total runtime metric in milliseconds. The proposed model is found to have a better improvement in this metric compared to the conventional models.

V. CONCLUSION

Cloud based workload scheduling plays an essential role in the real-time cloud based applications. Since, most of the conventional approaches are difficult to schedule the tasks using different cloud instance types, it is necessary to find and allocate the key resources to each virtual machine using the improve meta-heuristic approach. Traditional approaches use static metrics to define the task allocation problem to the available resources. In this work, a Cluster-based Energy Efficient Q-Ant Colony Optimization (CEQACO) framework is proposed for cloud computing environments. The proposed framework utilizes clustering techniques to group cloud virtual machines (VMs) based on their workload characteristics, and then applies a Q-Ant Colony Optimization algorithm to optimize the allocation of VMs to physical servers. The results show that the CEQACO framework is able to reduce time computation and energy by up to 6%, while still meeting the quality of service requirements of cloud users.

Future work: With the increasing focus on sustainability and energy efficiency, future research can focus on developing cloud task scheduling models that optimize resource allocation based on energy consumption. This can include developing techniques for dynamically adjusting resource allocation based on energy consumption levels, or incorporating energy consumption metrics into the task scheduling decisions.

REFERENCES

[1] M. Kumar, S. C. Sharma, A. Goel, and S. P. Singh, “A comprehensive survey for scheduling techniques in cloud computing,” Journal of Network and Computer Applications, vol. 143, pp. 1–33, Oct. 2019, doi: 10.1016/j.jnca.2019.06.006.

[2] M. Hussain, L.-F. Wei, F. Abbas, A. Rehman, M. Ali, and A. Lakhan, “A multi-objective quantum-inspired genetic algorithm for workflow healthcare application scheduling with hard and soft deadline constraints in hybrid clouds,” Applied Soft Computing, vol. 128, p. 109440, Oct. 2022, doi: 10.1016/j.asoc.2022.109440.

[3] K. Dubey and S. C. Sharma, “A novel multi-objective CR-PSO task scheduling algorithm with deadline constraint in cloud computing,” Sustainable Computing: Informatics and Systems, vol. 32, p. 100605, Dec. 2021, doi: 10.1016/j.suscom.2021.100605.

[4] M. Kalra and S. Singh, “A review of metaheuristic scheduling techniques in cloud computing,” Egyptian Informatics Journal, vol. 16, no. 3, pp. 275–295, Nov. 2015, doi: 10.1016/j.eij.2015.07.001.

[5] S. A. Murad, A. J. M. Muzahid, Z. R. M. Azmi, M. I. Hoque, and M. Kowsher, “A review on job scheduling technique in cloud computing and priority rule based intelligent framework,” Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 6, Part A, pp. 2309–2331, Jun. 2022, doi: 10.1016/j.jksuci.2022.03.027.

[6] M. Gill and D. Singh, “ACO Based Container Placement for CaaS in Fog Computing,” Procedia Computer Science, vol. 167, pp. 760–768, Jan. 2020, doi: 10.1016/j.procs.2020.03.406.

[7] H. Xing, J. Zhu, R. Qu, P. Dai, S. Luo, and M. A. Iqbal, “An ACO for energy-efficient and traffic-aware virtual machine placement in cloud computing,” Swarm and Evolutionary Computation, vol. 68, p. 101012, Feb. 2022, doi: 10.1016/j.swevo.2021.101012.

[8] M. Alaei, R. Khorsand, and M. Ramezanpour, “An adaptive fault detector strategy for scientific workflow scheduling based on improved differential evolution algorithm in cloud,” Applied Soft Computing, vol. 99, p. 106895, Feb. 2021, doi: 10.1016/j.asoc.2020.106895.

[9] S. G. Domanal and G. R. M. Reddy, “An efficient cost optimized scheduling for spot instances in heterogeneous cloud environment,” Future Generation Computer Systems, vol. 84, pp. 11–21, Jul. 2018, doi: 10.1016/j.future.2018.02.003.

[10] M. Abdullahi, M. A. Ngadi, S. I. Dishing, S. M. Abdulhamid, and B. I. Ahmad, “An efficient symbiotic organisms search algorithm with chaotic optimization strategy for multi-objective task scheduling problems in cloud computing environment,” Journal of Network and Computer Applications, vol. 133, pp. 60–74, May 2019, doi: 10.1016/j.jnca.2019.02.005.

[11] A. Talha, A. Bouayad, and M. O. C. Malki, "An improved pathfinder algorithm using opposition-based learning for tasks scheduling in cloud environment," Journal of Computational Science, vol. 64, p. 101873, Oct. 2022, doi: 10.1016/j.jocs.2022.101873.
[12] N. Sharma, Sonal, and P. Garg, "Ant colony based optimization model for QoS-based task scheduling in cloud computing environment," Measurement: Sensors, vol. 24, p. 100531, Dec. 2022, doi: 10.1016/j.measen.2022.100531.
[13] G. J. Samuel Babu and M. Baskar, "Application of blockchain methodology in secure task scheduling in cloud environment," Advances in Engineering Software, vol. 172, p. 103175, Oct. 2022, doi: 10.1016/j.advengsoft.2022.103175.
[14] G. Rjoub, J. Bentahar, and O. A. Wahab, "BigTrustScheduling: Trust-aware big data task scheduling approach in cloud computing environments," Future Generation Computer Systems, vol. 110, pp. 1079–1097, Sep. 2020, doi: 10.1016/j.future.2019.11.019.
[15] S. Vishwakarma et al., "Cloud data storage with improved resource scheduling in healthcare application based on security system," Optik, vol. 272, p. 170225, Feb. 2023, doi: 10.1016/j.ijleo.2022.170225.
[16] A. M. Yadav and S. C. Sharma, "Cooperative task scheduling secured with blockchain in sustainable mobile edge computing," Sustainable Computing: Informatics and Systems, vol. 37, p. 100843, Jan. 2023, doi: 10.1016/j.suscom.2022.100843.
[17] X. Wang et al., "Dynamic scheduling of tasks in cloud manufacturing with multi-agent reinforcement learning," Journal of Manufacturing Systems, vol. 65, pp. 130–145, Oct. 2022, doi: 10.1016/j.jmsy.2022.08.004.
[18] S. Mangalampalli, G. R. Karri, and G. N. Satish, "Efficient Workflow Scheduling algorithm in cloud computing using Whale Optimization," Procedia Computer Science, vol. 218, pp. 1936–1945, Jan. 2023, doi: 10.1016/j.procs.2023.01.170.
[19] R. Medara, R. S. Singh, and Amit, "Energy-aware workflow task scheduling in clouds with virtual machine consolidation using discrete water wave optimization," Simulation Modelling Practice and Theory, vol. 110, p. 102323, Jul. 2021, doi: 10.1016/j.simpat.2021.102323.
[20] J. Wen, J. Yang, T. Wang, Y. Li, and Z. Lv, "Energy-efficient task allocation for reliable parallel computation of cluster-based wireless sensor network in edge computing," Digital Communications and Networks, Jun. 2022, doi: 10.1016/j.dcan.2022.06.014.
[21] J. K. Konjaang, J. Murphy, and L. Murphy, "Energy-efficient virtual-machine mapping algorithm (EViMA) for workflow tasks with deadlines in a cloud environment," Journal of Network and Computer Applications, vol. 203, p. 103400, Jul. 2022, doi: 10.1016/j.jnca.2022.103400.
[22] T. Kaur and I. Chana, "GreenSched: An intelligent energy aware scheduling for deadline-and-budget constrained cloud tasks," Simulation Modelling Practice and Theory, vol. 82, pp. 55–83, Mar. 2018, doi: 10.1016/j.simpat.2017.11.008.
[23] S. A. Alsaidy, A. D. Abbood, and M. A. Sahib, "Heuristic initialization of PSO task scheduling algorithm in cloud computing," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 6, Part A, pp. 2370–2382, Jun. 2022, doi: 10.1016/j.jksuci.2020.11.002.
[24] S. Iftikhar et al., "HunterPlus: AI based energy-efficient task scheduling for cloud–fog computing environments," Internet of Things, vol. 21, p. 100667, Apr. 2023, doi: 10.1016/j.iot.2022.100667.
[25] D. A. Shafiq, N. Z. Jhanjhi, and A. Abdullah, "Load balancing techniques in cloud computing environment: A review," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 7, pp. 3910–3933, Jul. 2022, doi: 10.1016/j.jksuci.2021.02.007.
[26] Y. Liu et al., "Logistics-involved service composition in a dynamic cloud manufacturing environment: A DDPG-based approach," Robotics and Computer-Integrated Manufacturing, vol. 76, p. 102323, Aug. 2022, doi: 10.1016/j.rcim.2022.102323.
[27] X. Zuo, G. Zhang, and W. Tan, "Self-Adaptive Learning PSO-Based Deadline Constrained Task Scheduling for Hybrid IaaS Cloud," IEEE Transactions on Automation Science and Engineering, vol. 11, no. 2, pp. 564–573, Apr. 2014, doi: 10.1109/TASE.2013.2272758.
[28] Z. Zhou, F. Li, J. H. Abawajy, and C. Gao, "Improved PSO Algorithm Integrated With Opposition-Based Learning and Tentative Perception in Networked Data Centres," IEEE Access, vol. 8, pp. 55872–55880, 2020, doi: 10.1109/ACCESS.2020.2981972.

[29] Z. Zhong, K. Chen, X. Zhai, and S. Zhou, "Virtual machine-based task scheduling algorithm in a cloud computing environment," Tsinghua Science and Technology, vol. 21, no. 6, pp. 660–667, Dec. 2016, doi: 10.1109/TST.2016.7787008.

[30] Z.-J. Wang et al., "Dynamic Group Learning Distributed Particle Swarm Optimization for Large-Scale Optimization and Its Application in Cloud Workflow Scheduling," IEEE Transactions on Cybernetics, vol. 50, no. 6, pp. 2715–2729, Jun. 2020, doi: 10.1109/TCYB.2019.2933499.

[31] Y. Wang and X. Zuo, "An Effective Cloud Workflow Scheduling Approach Combining PSO and Idle Time Slot-Aware Rules," IEEE/CAA Journal of Automatica Sinica, vol. 8, no. 5, pp. 1079–1094, May 2021, doi: 10.1109/JAS.2021.1003982.

[32] P. Wang, Y. Lei, P. R. Agbedanu, and Z. Zhang, "Makespan-Driven Workflow Scheduling in Clouds Using Immune-Based PSO Algorithm," IEEE Access, vol. 8, pp. 29281–29290, 2020, doi: 10.1109/ACCESS.2020.2972963.

[33] H. Sun, S. Wang, F. Zhou, L. Yin, and M. Liu, "Dynamic Deployment and Scheduling Strategy for Dual-Service Pooling-Based Hierarchical Cloud Service System in Intelligent Buildings," IEEE Transactions on Cloud Computing, vol. 11, no. 1, pp. 139–155, Jan. 2023, doi: 10.1109/TCC.2021.3078795.

[34] D. Subramoney and C. N. Nyirenda, "Multi-Swarm PSO Algorithm for Static Workflow Scheduling in Cloud-Fog Environments," IEEE Access, vol. 10, pp. 117199–117214, 2022, doi: 10.1109/ACCESS.2022.3220239.

[35] A. Song, W.-N. Chen, X. Luo, Z.-H. Zhan, and J. Zhang, "Scheduling Workflows With Composite Tasks: A Nested Particle Swarm Optimization Approach," IEEE Transactions on Services Computing, vol. 15, no. 2, pp. 1074–1088, Mar. 2022, doi: 10.1109/TSC.2020.2975774.

[36] M. Sardaraz and M. Tahir, "A Hybrid Algorithm for Scheduling Scientific Workflows in Cloud Computing," IEEE Access, vol. 7, pp. 186137–186146, 2019, doi: 10.1109/ACCESS.2019.2961106.

[37] H. Saleh, H. Nashaat, W. Saber, and H. M. Harb, "IPSO Task Scheduling Algorithm for Large Scale Data in Cloud Computing Environment," IEEE Access, vol. 7, pp. 5412–5420, 2019, doi: 10.1109/ACCESS.2018.2890067.