Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
 Volume 26, Issues 1, pp. 490–502, DOI 10.12694/scpe.v26i1.3955

ENERGY AND DEADLINE AWARE WORKFLOW SCHEDULING USING ADAPTIVE REMORA OPTIMIZATION IN CLOUD COMPUTING

VIDYA SRIVASTAVA, RAKESH KUMAR *

Abstract. Cloud computing has become a more popular and well-known computing paradigm for delivering services to different organizations. The main benefits of the cloud computing paradigm, including on-demand services, pay-as-per-use policy, rapid elasticity, and so on, make cloud computing a more emerging technology to lead with new methods. Cloud systems have become more challenging than other systems because of their wide range of clients and the variety of services in the system. The cloud data center consists of many physical machines (PM) with virtual machines (VM), load balancers, switches, storage etc. Because of the inappropriate use of resources and inefficient scheduling, these data centers consume a lot of energy. In this paper, a multi-objective optimization model called Adaptive Remora Optimization (AROA) is proposed, which comprises sub-models viz; priority calculation, task clustering, probability definition and task-VM mapping using search mode based on Remora optimization to optimize energy consumption and execution time. CloudSim is used for the implementation of the proposed optimization technique. Through simulation the energy consumption is 0.695kWh and the execution time is 179.14sec. The result obtained by AROA is compared with the existing approaches to prove the efficacy of the proposed approach.Experimental results show that the proposed AROA algorithm outperforms the existing approaches.

Key words: Cloud computing, Execution time, Energy consumption, Remora optimization, Task scheduling

1. Introduction. Cloud computing has become vital to IT-based organizations and individual users over the last few years [15]. It gives desired computing assets for multiple programs in virtual machines, software, and hardware delivered by cloud service via cloud data centers [14]. Growing demands on cloud resources are completed with more powerful servers and other related hardware assets. Cloud computing enables the delivery of on-demand services such as networking, software, and intelligence via the Internet [39]. SaaS (Software as a service), PaaS (Platform as a service), and IaaS (Infrastructure as a service) are the three primary services used in the cloud computing environment, which provide extensible payment services, public-private, hybrid, and community cloud are four deployment models . Properties having multilevel abstraction and virtualization make cloud more resourceful computing [16]. The user only focuses on the utilization of resources and does not bother about their physical location. To improve the efficacy of the task scheduling algorithm user performed the proper allocation of a task to an appropriate resource. Scheduling plays a significant role in cloud computing [11]. Scheduling can be referred as an essential operating system function. All the resources that are available for operation should be scheduled before operation. Task scheduling is responsible for giving access to the system resources with the help of thread and process. So, task scheduling problems can be considered the appropriate obtaining mapping between the set of tasks over the available resources (CPU, bandwidth, virtual machine) [9, 38] . There are three types of task scheduling algorithms: pre-emptive, non-pre-emptive, and round-robin scheduling. When a high-priority task arises, a pre-emptive task scheduling algorithm interrupts the operation [32]. In non-pre-emptive task scheduling running task is executed till its completion. The third one is the round-robin scheduling algorithm that follows the FCFS policy in which pre-emption occurs in the middle of the operation [28]. In a scheduling-based scheme, all the tasks and jobs are appropriately arranged without interruption. Work must be completed within the deadline and executed individually to maximize reliability and minimize the execution time to optimize the system's overall time for allocating resources [30, 16].

Task scheduling plays a significant role in the cloud computing system. It cannot be done based on one criterion but on the various rules and regulations that can be termed as an accord between the user and cloud provider. The most crucial method to tackle the challenge of reducing energy consumption in the cloud environment can be done through effective task scheduling.[19]. According to the agreement, high-quality

*Madan Mohan Malaviya University of Technology, Gorakhpur, India (2020028010@mmmut.ac.in, rkiitr@gmail.com)

services to users and clients are the determining task for providers. At the same time, numerous jobs are running at the provider. Centralized and distributed are two main types of cloud task scheduling[31]. In a centralized task scheduling scenario, a single scheduler assigns tasks to resources, but if the scheduler fails, all systems are shut down. Additionally, in a centralized task scheduling system, scalability problems and fault tolerance issues occur[18]. In distributed task scheduling algorithms, schedulers are connected, meaning that all schedulers are collectively assigned to the tasks [6]. The single point of failure problem has been solved in distributed scheduling, but another complication, i.e., congestion problem, occurred between schedulers [1].

1.1. Motivation and Contribution. One of the significant awareness gaps associated with the previous survey is mapping tasks with appropriate VM placement in cloud computing. Several challenges arise in delivering services to the cloud user. There is a need for an efficient scheduling process provided by the cloud scheduler, such as SLA parameter, time, and cost must satisfy the user constraint[22]. In the case of a task, scheduling is concerned with the optimal mapping of resources on the specified virtual machine so that better system performance must be achieved. Variations in task characteristics and resource heterogeneity scheduling lead to NP-complete problems. No method is specified for finding the polynomial solution for such a problem. The scheduling algorithm reduced the makespan of the application[2]. Workflow scheduling is considered more with the emerging growth of cloud computing technology. A scheduling algorithm minimizes the makespan of the data center. However, sometimes selected energy does not give the optimal solution, so task scheduling can be treated as the more challenging way to increase the reliability of a system and reduce the makespan simultaneously [4]. To overcome this problem, an optimization function/ fitness function is required for finding the suitable scheduling. In the cloud environment, scheduling issues become widely explored, whether it is to be single or multi objectives challenges [37]. A large no. of applications with their higher load makes the cloud more complex due to the inefficient use of resources. It is a challenge for researchers to present a new optimization approach in a dynamic environment that overcomes the problem in previous studies. The user aims to find suitable physical hosts for their users, and cloud providers aim to utilize their infrastructure.

2. Related Works. The author of [26] proposed a workflow scheduling system that is both an energy-efficient and reliable(EERS) mechanism that jointly optimizes the system's dependability and minimizes energy consumption. EERS consists of five sub-algorithms; first, a rank calculation algorithm is used to preserve the dependency of a task. Second is the clustering algorithm for conserving energy. The third discusses a distribution mechanism for defining the makespan for each task, then applies the fourth method i.e., cluster-VM mapping aiming to maximize the system's reliability and minimize energy consumption. The last algorithm is a slack algorithm associated with non-critical tasks. The simulation result indicates that the proposed algorithm optimizes energy and reliability in polynomial time. It also examined the genuine electricity cost for the task scheduling algorithm as a subsequent work.

The paper [12] developed a new scheduling approach known as reliability and energy-efficient workflow scheduling (REEWS) to maximize a system's reliability and minimize energy consumption. REEWS Scheme is divided into four sub-algorithms. The first is priority calculation, the second is the clustering of the task, the third is distribution, and the fourth is assigning clusters with a proper frequency level. Gaussian elimination and a randomly generated graph are used to enhance an algorithm's performance. The performance of the REEWS algorithm is compared with other well-known such as heterogeneous Earliest-finish-time(HEFT), reliable-HEFT(RHEFT), algorithms. It will have observed that the REEWS algorithm has outstanding performance compared to the different algorithm simulated result will be enhanced by combining the clustering algorithm with the load balancing module.

The survey [24] proposed an in-depth analysis of PSO scheduling, main objectives such as load balancing, makespan, and execution time. Particle swarm optimization is a population-based meta-heuristic technique covering a wide range of applications because of its effectiveness and low computational cost. In addition to that, different levels of trust and reliability must be investigated and evaluated to solve more scheduling problems.

The author of [25] developed a task scheduling optimization scheme based on improved ant colony optimization to improve scheduling methods that fall under local optimization. The fitness function determines the optimal solution for the scheduling algorithm. The feasibility analysis demonstrates how the algorithm performs well with the fastest convergence speed and shortest computation time.

The paper [5] introduced a technique based on the co-optimization process aiming to map the task into a virtual machine within the deadline time and then assign a suitable placing virtual machine to the correct physical host within the capacity constraint. Experimental results show how this technique performs better than all other optimization problems.

Another paper [29] proposed algorithm works in two stages: Virtual machine(VM) scheduling and consolidation. The maximum runtime job is assigned to the virtual machine during the VM scheduling phase, which is expected to reduce energy usage. A double threshold technique is used in the consolidation phase to find overload and underload hosts. Experimental results simulate the Combination of scheduling and consolidation phase successfully increases resource utilization and decreases energy consumption.

The study done by [35] discussed a comprehensive review of meta-heuristic optimization algorithms in the cloud computing system. Approaches discussed in the metaheuristic mechanism will be enough for the reader to select a new mechanism for solving task scheduling problems. A brief review of future research work was also discussed in the research work.

The author of [36] presents two heuristic algorithms, including budget-deadline constraints. Resources that support DVFS technology use Budget-deadline DVFS enable energy (BDD); resources that do not work well with DVFS use Budget-deadline constraint energy aware (BDCE). Several metrics like cost, utilization rate, energy consumption, and success rate are utilized to estimate the fulfillment of the proposed algorithm.

The paper [13] presents a comprehensive review of energy optimization in a cloud computing environment that compares 67 scheduling algorithms that reduce energy consumption throughout the scheduling process. This work is appropriate for the reader to select a relevant approach to minimizing energy consumption.

The study done in [3] developed a multi-workflow scheduling algorithm with dynamic reusability aiming to minimize the time taken to complete individual tasks and then computing the overall time taken to complete tasks within the deadline; they dynamically reuse the available virtual machine while it is needed Because of the maximum utilization of resources such as CPU and virtual machines within their deadline, simulation results show that the algorithm is moving towards a new generation of multi-objective scheduling. Table 2.1 shows the Silent features of a few existing task-scheduling algorithms.

3. System Model. This section introduced various types of system models, including the cloud model, task model, workflow model, and energy model detailed illustration has been done in the proposed work. The Cloud sim toolkit is used for simulation to implement the proposed work. For easy insight, Table 3.1 outlines the primary notations with their meaning, which are used in this research paper.

3.1. Cloud Model. In general cloud system can be represented by infrastructure as a service (IaaS) which is responsible for managing resources , such as physical and virtual, to fulfilling the requirement of cloud users. Researchers were aiming to build a system model to enhance Various parameters available to measure the capacity of a system, such as CPU, RAM, Bandwidth, and storage [34, 23]. For scheduling, the execution of workflow virtual machine is assigned to each physical host[32]. A physical host can carry more than one virtual machine based on requirements. We consider M number of virtual machines available that can be represented as VM= VM_1, VM_2, VM_3.... VM_v, which are fully connected with each other. Fig. 3.1 shows the cloud architecture.

3.1.1. Task Model. Task scheduling is defined as assigning a task to the appropriate resources to minimize energy consumption and execution time. Several models are available depending on the scheduling criteria available on the specific cloud system. Consider m number of resources present in cloud represented as R= r_1,r_2,r_3....r_c, and mapping of task is denoted by M f : $T \to R$ represent the mapping function such as Mf (j) ,represent resources corresponding to the task t_k is assigned[10]. Fig. 3.2 shows the task model The virtual machine is the collection of computing resources that helps to virtualize the physical machine.

3.1.2. Energy Model. In the data center model, resources such as CPU and other networking devices can cause energy consumption. From the net amount of energy, only processors have depleted 37-43% In the data center model, resources such as CPU and other networking devices can cause energy consumption. From the net amount of energy, only processors have depleted 37-43% Networking devices consumed 33% of total energy[33]. As we all know, all networking devices are nearly fixed and cannot be modified in the event of any execution workflow. When the processor begins to function, the energy consumption is directly proportional

Table 2.1: Silent features of existing task scheduling algorithms

Authors and year of Publication	Key objective	Application area	Issues
Ref.[[26]](2021)	Minimizing energy consumption maximizes system reliability.	EERS algorithms have been proposed for workflow scheduling and consolidation.	Cost,frequency independent energy consumption
Ref.[[12]](2019)	Perform a combinatorial optimization of the reliability of the application and guarantees the QoS.	Proposed a REEWS Algorithm executes in four stages: calculation, clustering, time distribution associated with assigning appropriate voltage	Only consider essential features of scheduling algorithms.
Ref.[[5]] (2021)	Minimizing execution cost, degree of Imbalance, makespan, and cost.	Aims to assign scheduled tasks into appropriate VM with the least execution cost within deadline constraints.	More memory required to increase complexity.
Ref.[[24]](2017)	Minimize energy consumption and achieve green cloud computing .	Presents a deep analysis of the particle swarm optimization algorithm.	Not consider Load balancing and VM parameters.
Ref.[[37]](2020)	Increase convergence speed, and minimize completion time,	Propose an improved ant colony algorithm, including pheromone update and volatilization	Resource utilization, ensuring QoS and energy consumption.
Ref.[[29]](2018),	Maximize resource utilization, minimize energy consumption.	The migration technique is used for the detection of overload and underload hosts.	Unable to find a better resource selection.
Ref.[[35](2017)]	Suggest readers decide suitable approach for scheduling	Present a competitive taxonomy among existing algorithms.	Lack of scalability and standardization in new computing model.
Ref. [[36]](2021)	Affordable price increasess-cheduling length and higher energy-saving ratio.	Develop two heuristic algorithms suitable for cloud environments, including budget and deadline constraints.	High-fault occurrence, high resource wastage.
Ref.[[13]](2022)	Comparative analysis of 67 algorithms to minimize energy consumption.	Discuss important aspects of heuristics, metaheuristics, and task scheduling algorithms.	Need for a simulator that supports multi-objective scheduling.
Ref. [[3](2018)	Reduce the makespan of every task and wrap up the overall execution of the entire workflow within the deadline.	Present competent and dynamic multi-workflow scheduling (CDMWS) algorithm that dynamically produces VM.	Increases network load, rescheduling task

to the execution of the workflow. When the resources are equipped, static and dynamic energy are responsible for energy consumption; static energy remains constant, whereas dynamic energy is entirely dependent upon the frequency/voltage of the corresponding processor. In the case of frequency transition, we consider zero overhead because each transition takes a minute amount of time (about the microsecond range). It is more precise to say that dynamic energy is blamed for energy consumption.

3.1.3. Workflow Model. Workflow can be defined as the group of computational tasks with their reliance constraint between them[20]. Data passing from one workflow to another is considered a dependency constraint. So, DAG D = (T, E) where T= collection of task (t_1, t_2,t_3.....t_k) and E = collection of edge(e_1,e_2,e_3,.....e_g,) which indicate the correlation between the task. The graph is used to dictate the workflow task. It is an automation of a repeatable pattern of processes where the data and information are passed from one cloud user to another for specific action. To improve efficiency and profitability, there is a need for coordination between

Table 3.1: Primary Notations

Notation	Meaning
EnC	Energy Consumption
ExT	Execution time
$EnC_dynamic$	Dynamic energy consumption
$EnC_Sstatic$	Static energy consumption
t_k	The kth task in workflow
r_c	The Cth resource in workflow
Cc	Computing capacity
Vol	Voltage
f	Frequency
M	Total number of virtual machines
Cp	Capacitance
P	Remora Position in search space
Dist	Current optimal solution
n	Represents the number of remora
i	Current iteration
I	Maximum iteration
Pbest	The optimal solution in the algorithm
f(Pbest)	The fitness function of the best position
P_n	Current position
P_{pre}	Position of the previous iteration
P_{tnt}	Tentative step
P_{random}	Random location
randn	Small global movement
A	Volume space

the user and synchronized data. In workflow execution criteria, workflow scheduling plays a major key issue; however, workflow defines the execution of workflows on which tasks are assigned to well-suited resources for satisfying constraints such as energy consumption and execution time[27].

3.2. Problem Formulation. We consider independent scheduling tasks that comprise heterogeneous virtual machines (VM) and physical machines (PM). This section introduces an optimization function that consists of reducing energy consumption as well as execution time and also represents constraints that are specified in this problem. The main aim is to find a suitable algorithm for workflow scheduling and virtual machine placement to reduce energy consumption and execution time.

3.2.1. Energy Consumption. In this study, we opted classic energy consumption model to analyze the energy consumption.

$$EnC = EnC_{static} + EnC_{dynamic} \tag{3.1}$$

where EnCstatic is energy consumption when the system does not carry out any workload, i.e., the system turned off. We consider the dynamic consumption of the cloud model to be discussed in the model. Total Energy consumption is defined in Eq. (3.1). Dynamic energy consumption occurs when the system turns on. Formulations Cp, Vol and f, represent constants belonging to processor capacity, voltage, and frequency, respectively, defined in Eq. (3.2).

$$EnC_{dynamic} = Cp.Vol^2.f \tag{3.2}$$

where C, V and f represent capacity voltage and frequency, respectively. Since $f \propto vol^{\frac{1}{\eta}} for(0 < \eta < 1)$ or i.e frequency-dependent energy consumption defined in Eq.(3.3)

$$EnC_{dynamic} \propto f^{\lambda} \quad \lambda = 1 + 2/\eta \geqslant 3 \tag{3.3}$$

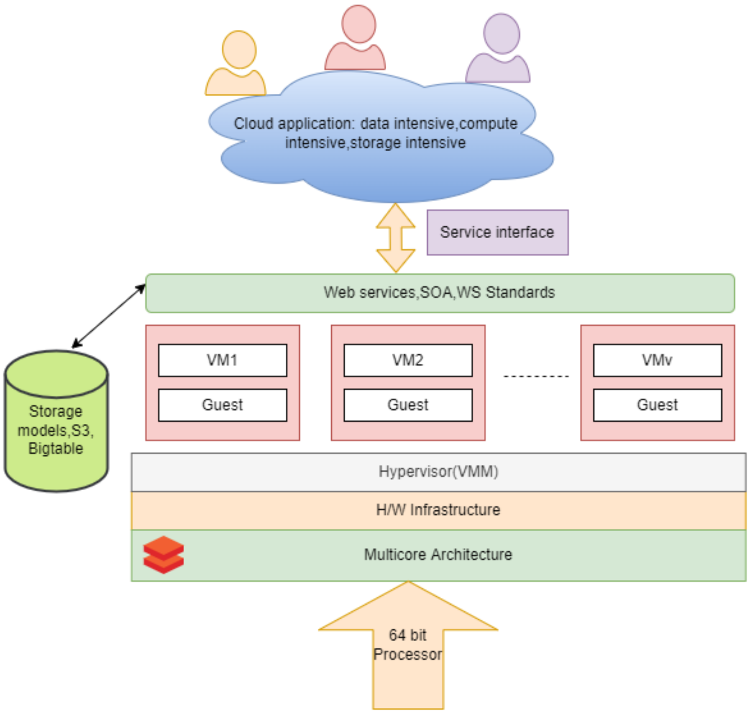

Fig. 3.1: Cloud Architecture

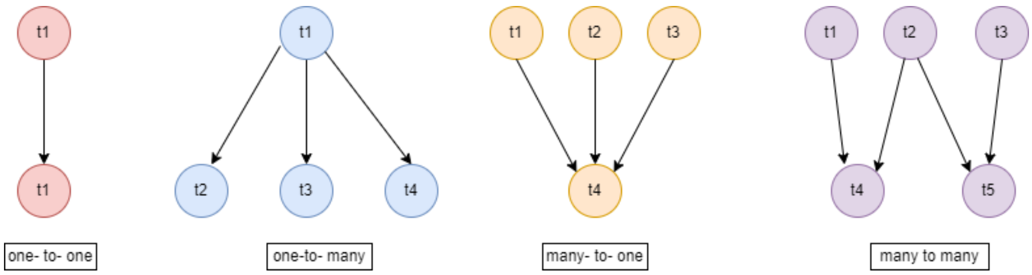

Fig. 3.2: Task Model

So, in this study, energy consumption is represented in Eq. (3.4)

$$EnC = EnC_{static} + Cp.f^{\lambda} \tag{3.4}$$

Further, Energy consumption can be divided into two parts-resource is busy/idle. Processor frequency will be at min level when the resources are idle. Processor frequency will be highest in case of execution starts, defined in Eq. (3.5)

$$EnC_{r_c}[total] = EnC_{r_c}[idle] + EnC_{r_c}[busy] \tag{3.5}$$

when resource rc is busy (i.e, the task is executing), energy consumption is calculated by Eq. (3.6), where Vol and f represent the voltage and frequency of resources rc upon the task tk is being finished their execution.

$$EnC_{r_c}[busy] = \sum_{t_k \in r_c} C_p.Vol^2_{r_c,t_k}.f_{r_c,t_k}.E_xT \tag{3.6}$$

When a resource is idle (i.e., the task is not executing), the processor works at a minimum frequency and voltage. During an idle time of resource, energy consumption is defined in equation (3.7), where idlerm defined idle time of rm.

$$EnC_{r_c}[idle] = C_p.Vol^2_{min}.f_{min}.idle_{r_c} \tag{3.7}$$

Therefore total energy consumption can be calculated in equation(3.8).

$$EnC_{r_c}[total] = \sum_{t_k \in r_c} Cp.Vol^2_{r_c,t_k}.f_{r_c,t_k}.E_xT + Cp.Vol^2_{min}.f_{min}.idle_{r_c} \tag{3.8}$$

3.2.2. Execution Time. The Execution time ExT denotes the time interval that is taken for executing task tk on resources VMCp based on the computing capacity of available resources that is defined as in Eq.(3.9):

$$ExT(t_k, VM_{Cc}(t_k)) = \frac{\sum_{i=0}^{k} length(t_k)}{\delta(VM_{Cc})} \tag{3.9}$$

3.3. Proposed Mechanism. We proposed a new scheduling algorithm called Adaptive Remora Optimization Algorithm(AROA) to minimize energy consumption and execution time (Figure 3.3). The mechanism is explained in four phases:

1. Priority calculation to provide a reliable topological workflow ordering that satisfies precedence constraint.
2. Tasks clustering for reducing the communication cost among tasks aiming to minimize energy consumption of the given system.
3. Define a probability definition.
4. Assigning the tasks to the appropriate processor at the proper voltage/frequency level to minimize energy consumption and execution time. Fig.3.3 shows proposed mechanism.

3.3.1. Priority Calculation. The task priority is calculated to ensure that the most time-consuming tasks are finished first. The task priority order is calculated to ensure that the most tedious tasks are executed first. In addition to that, it saves tasks that are waiting for input from higher-priority tasks. The tasks are prioritized in a hierarchical order for scheduling. The tasks are scheduled so that the precedence constraints are met.

Algorithm 1 Priority Calculation for Task (t_k)

```
Initialize the number of tasks (t_k)
for each task do
    Evaluate execution time (ExT)
    Sort the execution time in ascending order
    Assign the priority such that the lowest execution time task gets the highest priority
end for
```

3.3.2. Clustering of Task. After priority calculation of all tasks, clustering will be the next step in which a cluster formation occurs, which will be used in remora scheduling.

3.3.3. Probability Definition. The proposed algorithm reduces the execution time by optimizing the available number of processors and energy consumed by the processor. The scheduling method divided into the no. of equal time steps (Δt_s). The probability-based scheduling algorithm determines the probability of execution time P(ExT) of the task into equal time steps (Δt_s). The task is scheduled at which time when the probability of execution time is minimum. The probability of execution time is the sum of its current task and its successor's tasks. It is assumed that all tasks are operated at a higher frequency[21]. The probability of execution time is dynamically updated based on time steps assigned to the selected voltage-scaled job. Finally, the no. of processors required to execute the last schedule is calculated. The steps are explained in the following:

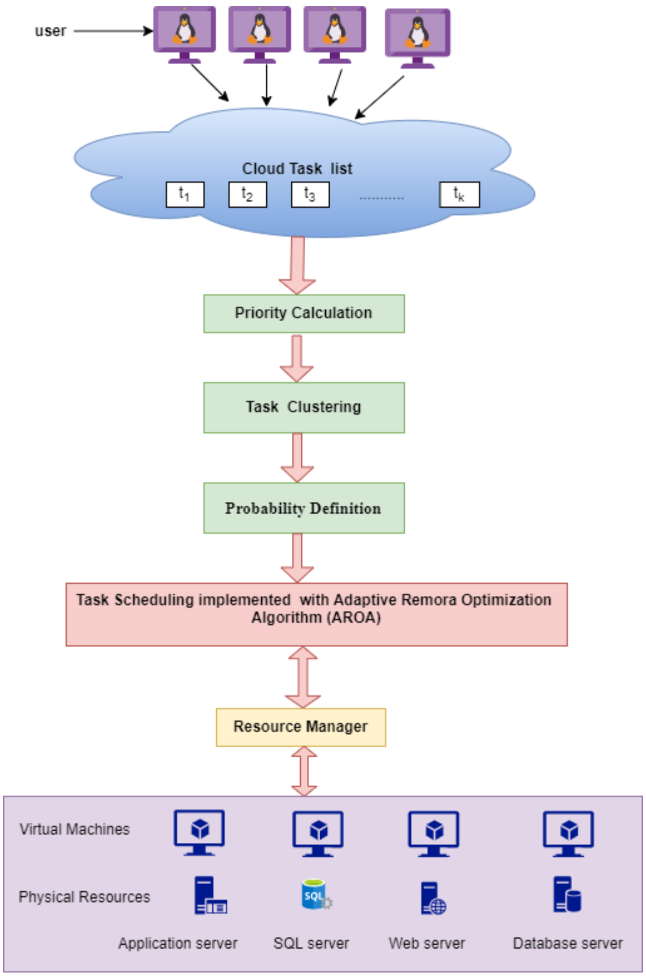

Fig. 3.3: Proposed Mechanism

Algorithm 2 K-Means Clustering Algorithm

1: Select the number of clusters, k
2: Select random tasks as initial centroids (other than the input dataset)
3: **repeat**
4: Allocate each task to its closest centroid, forming k clusters
5: Evaluate variance and update centroids for each cluster
6: Repeat the allocation and centroid update for more cluster formation
7: **until** No reassignment occurs
8: Cluster formation is complete

Step 1 (Time step calculation) Available tasks in cluster (t_1, t_2, t_3, t_4 ... t_k) is to be considered they are operated at the BCET scheduling algorithm. All the tasks are operated at a higher frequency. Tasks are divided into the time step taking as a unit with the smallest time step execution.

Step 2 (Best-case Performance) Using BFS (breadth-first search), the best-case algorithm begins with the source task in the task graph before it. In case of scheduling the predecessor, each task is scheduled with the earliest available time step BCP (t_k). The task is categorized by its start time BCPstrt(t_k).

and finish time BCPfn(t_k). The earliest possible execution time of the task graph is described in this algorithm.

Step 3 (Worst-case Performance) Worst-case scheduling same as the best-case scheduling algorithm. The difference is only its start with the sink of the task graph with upward preceding. Algorithms describe the longest possible execution of the task graph. Task(t_k) is categorized by its start time WCPstrt(t_k). and finish time WCPfn(t_k). This algorithm describes the latest possible execution time of the task graph.

Step 4 (Bound limit demonstration) Algorithm moves towards determining the bound limit (Δtk) of each task. The bound limit can be calculated in Eq. (3.10).

$$\Delta t_k = worstcase_{fn}(t_k) - bestcase_{strt}(t_k) \tag{3.10}$$

where Δt_k is the time step within which the current task is scheduled.

Step 5 (Finding the probability of execution time of each task in each time step) Next step is to find the probability of execution time in each time

3.4. Adaptive Remora Optimization Algorithm. ROA is a new natural bio-inspired and meta-heuristic algorithm aiming to minimise energy consumption and execution time. Parasitic behavior is the main inspiration for the remora algorithm. Host locations are updated[17]. In the case of a large host (giant whales), remora feed on the host's extermination and natural enemies. In the case of a small host, the remora chase the host to move fast(swordfish) to the bait-rich area to prey; from the above two cases, the remora makes a judgment based on experience. When it starts initiation to prey, continuously update the host and make a global decision. If it eats encircling the host remora, continue to local update without changing their host. Fitness function f(x) is defined in Eq.(11)

A population-based search algorithm's main attainment is exploration and exploitation trade-off. AROA has two parts: exploration and exploitation like ROA. These two parts have a great impact on the algorithm that how long these two algorithms perform. The main difference between remora and adaptive remora optimization is utilizing a new parameter, "search mode(SM)." As mentioned in the AROA algorithm, the value decreases over time with a small value. In the initial search, space exploration changes the solution, activating the exploitation part on time increment. However, in AROA, the active part, including viz, exploitation and exploration, is altered adaptively by introducing the search mode parameter for tracking the behaviour of the solution in the given population. Assuming the search mode value is set to 1 for promoting the exploitation stage, the calculated probability from Probability Definition Step 5, In such a situation exploitation stage become more active than exploration stage. If the solution quality does not change over the iteration search mode is updated 2. in such a scenario exploration stage becomes more active compared to exploitation. According to algorithm 3, search mode SM not only determines the active part adaptively at a time, also justifies the duration when the exploration/exploitation parts become active and can change the position of solution.

4. Simulation Results and Experiments. CloudSim tool is used for evaluating the proposed AROA optimization algorithm. It is broadly used to simulate the cloud systems methods such as virtual machines and data centers to support task scheduling policies such as task selection and virtual machine placement [8]. It is to be very clear that the proposed AROA works on the large-scale data center. At the host level total no. of a heterogeneous host is taken to be 800 capacity of RAM is 512MB, and the corresponding bandwidth is to be 1000 Mbps, CPU capacity is 50 MIPS, and storage is 100GB. At VM level total no. of XEN virtual machine is 1175, no, of tasks is set to be 3000, the memory is 1536 MB, the CPU capacity is 1000 MIPS, no. of CPU is 1, the bandwidth is 1000 Mbps, and the storage is 1000 GB. Various varieties of performance are evaluated to calculate the performance of energy consumption and execution time. Based on metrics proficiency, the proposed algorithm can be calculated. A comparison between the existing and proposed approaches is performed[7].

4.1. Evaluation of Energy Consumption. The experimental results of the proposed AROA are to be discussed in this section. Adaptive remora optimization's performance is comparable with the well-known existing approach to find the efficiency of the proposed method. Existing mechanisms are the Combination of metaheuristic approaches such as Genetic algorithm (GA), Particle-swarm optimization (PSO) algorithm, Minimum-Migration Time (MMT) policy, Random selection (RS) policy, Median Absolute Deviation (MAD)

Algorithm 3 Adaptive Remora Optimization Algorithm

Input: Application graph $G(T, E)$, number of physical machines/processors
Output: Energy-efficient workflow scheduling
Step 1 Input the task t_k in T
Step 2 Calculate priority from Algorithm 1
Step 3 Assign priority such that the lowest execution time task gets the highest priority
Step 4 Make clusters using Algorithm 2
Step 5 Calculate the number of clusters
Step 6 For each cluster
Step 7 Select Remora population size (N) and Max number of iterations (I)
Step 8 Set the position of entire search agents P_n $(n = 1, 2, 3, \ldots, N)$
Step 9 While $(i \leq I)$
Step 10 Evaluate the fitness of each remora
Step 11 Evaluate the best fitness and best position, P_{best}
Step 12 $Prob_s =$ calculated using Probability Definition in Step 5; $SM = 1$; $K_{\text{rand}} = \text{rndreal}(0, 1)$
Step 13 For nth remora
Step 14 if $H(n) = 0$
Step 15 if $(SM == 1$ and $K_{\text{rand}} \leq Prob_s)$ or $(SM == 2$ and $K_{\text{rand}} > Prob_s)$ // exploits mode
Step 16 ...
Step 17 End if
Step 18 Else if $H(n) = 1$
Step 19 if $(SM == 2$ and $K_{\text{rand}} \leq Prob_s)$ or $(SM == 1$ and $K_{\text{rand}} > Prob_s)$ // explore mode
Step 20 ...
Step 21 End if
Step 22 End if
Step 23 Generate tentative candidate position
Step 24 ...
Step 25 If $f(P_{\text{tnt}}) < f(P_n)$
Step 26 $P_n = P_{\text{tnt}}$
Step 27 $H(n) = \text{round(random)}$
Step 28 Else
Step 29 Update
Step 30 Endif
Step 31 End for
Step 32 $i = i + 1$
Step 33 End While
Step 34 Return the best fitness value P_{best}

and Interquartile Range (IQR) algorithms. The number of virtual employed for executing workflow is the important factor for calculating that is based on energy consumption. Virtual machines take place inside the physical machine. Thus, energy consumption is closely associated accord to the number of the current physical server. Energy consumption is calculated based on Eq. (3.8).

Fig. 4.1 compares the proposed AROA energy consumption(in kWh) with eight existing approaches. Compared to the existing approaches, the proposed AROA consumes minimum energy throughout the entire execution in the cloud computing system. The proposed approach is efficient for workflow based on the experimental results.

4.2. Evaluation of Execution Time. Fig. 4.2 compares the execution time (in sec) of the proposed AROA with eight existing approaches. The proposed approach consumed less time in the cloud computing system when compared to the existing approaches. From the experimental results, the proposed approach is efficient for workflow scheduling.

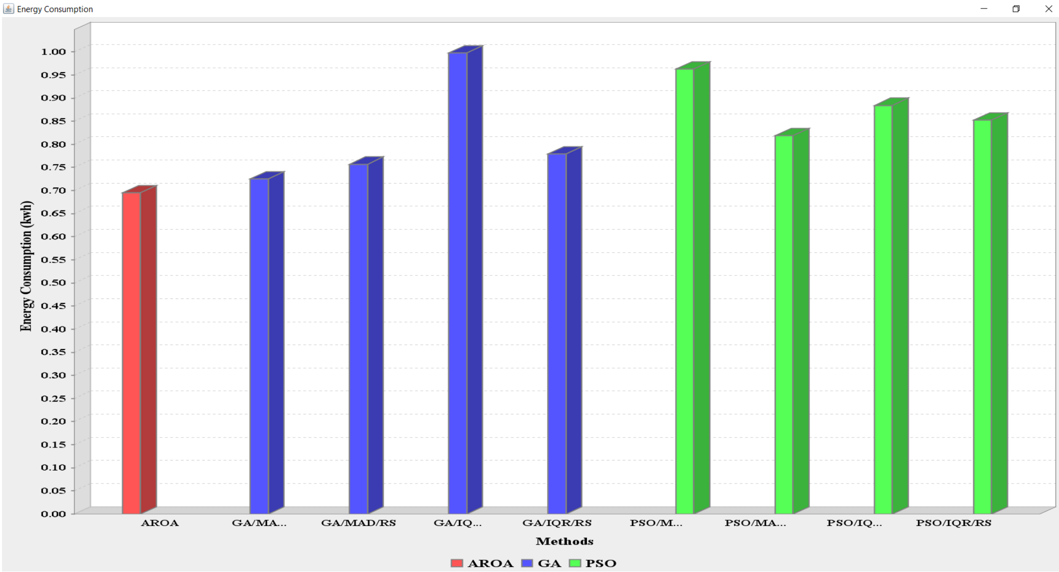

Fig. 4.1: Comparison of Proposed AROA EnC and eight existing approaches

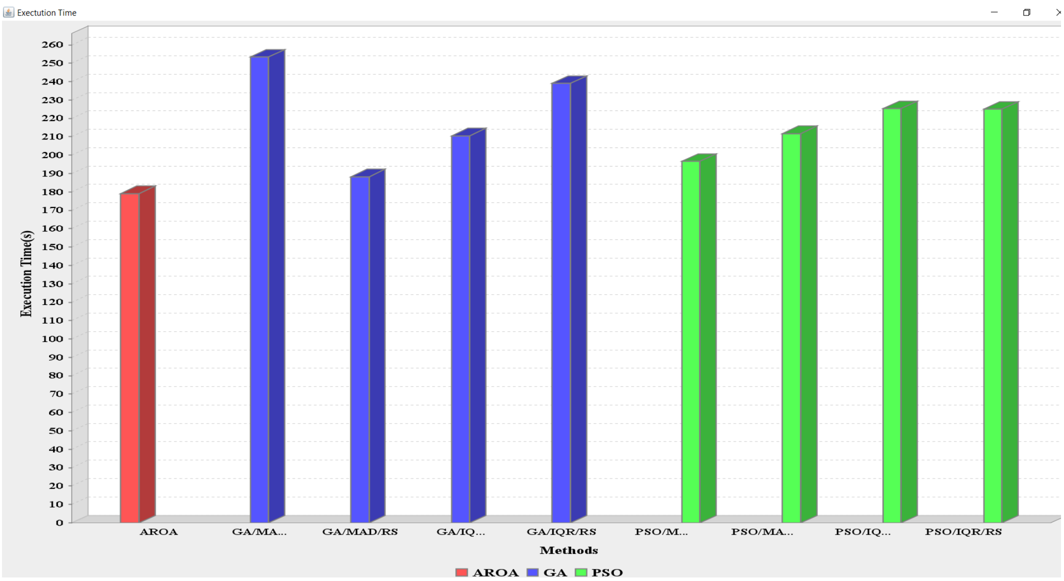

Fig. 4.2: Comparison of proposed AROA ExT and eight existing approaches

4.3. Result Discussion. The performance comparison of the proposed Adaptive remora optimization algorithm (AROA) was done versus the eight existing meta-heuristic algorithms, including GA/MAD/MMT, GA/MAD/RS, GA/IQR/MMT, GA/IQR/RS, PSO/MAD/MMT, PSO/MAD/RS, PSO/IQR/MMT, PSO/IQR/RS. The performance analysis indicates that the proposed AROA provides better results when comparison is performed with the existing mechanism for task scheduling. Proposed AROA energy consumption is .695 kWh and execution time is 179.14sec.

4.4. General Computational Complexity Calculation. The computing complexity influences the initialization, fitness assessment, and position update technique of the optimization algorithm. Initialization has an O(N) computational complexity for the fundamental ROA. In this case, the number of search agents is represented by the parameter N. The computational complexity of applying the SFO or WOA method for the entire iterative process is O (N $\times$ D$\times$ I), where I is the maximum number of iterations and D is the dimensions of the search space.

5. Conclusion. In cloud computing, the workflow schedule is a significant process for assigning tasks to virtual machines. Energy consumption and execution time have recently become more important in research work. We Consider several tasks in a cloud environment. Here we present Adaptive Remora optimization (AROA), comprising sub-models viz; priority calculation, clustering, probability definition and task-VM mapping using search mode for optimizing energy consumption and execution time. The experimental result shows that adaptive remora is more suitable for minimizing energy consumption and execution time. CloudSim is the simulation platform for the use implementation of AROA optimization techniques. Performance of proposed algorithm compared with some existing approaches. The experimental result depicts , energy consumption is 0.695kWh, and the execution time is 179.14 sec. The result exhibits that the proposed technique is much better than the existing approaches. Future scope of improvement consists of

1. Load balance mechanism on VM.
2. Adding more data center on the network.
3. Including metrics such as cost-effectiveness, cloud mobility.
4. Proposed reliability and budget constraint scheduling algorithm.
5. Concepts of hybrid cloud and security factors should be considered more.

REFERENCES

[1] M. Abd Elaziz, S. Xiong, K. Jayasena, and L. Li, *Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution*, Knowledge-Based Systems, 169 (2019), pp. 39–52.

[2] M. N. Abdulredha, A. A. Bara'a, and A. J. Jabir, *Heuristic and meta-heuristic optimization models for task scheduling in cloud-fog systems: A review*, Iraqi Journal for Electrical And Electronic Engineering, 16 (2020), pp. 103–112.

[3] M. Adhikari and S. Koley, *Cloud computing: a multi-workflow scheduling algorithm with dynamic reusability*, Arabian Journal for Science and Engineering, 43 (2018), pp. 645–660.

[4] R. Al-Arasi and A. Saif, *Task scheduling in cloud computing based on metaheuristic techniques: A review paper*, EAI Endorsed Transactions on Cloud Systems, 6 (2020).

[5] D. Alboaneen, H. Tianfield, Y. Zhang, and B. Pranggono, *A metaheuristic method for joint task scheduling and virtual machine placement in cloud data centers*, Future Generation Computer Systems, 115 (2021), pp. 201–212.

[6] N. Anwar and H. Deng, *A hybrid metaheuristic for multi-objective scientific workflow scheduling in a cloud environment*, Applied sciences, 8 (2018), p. 538.

[7] M. Askarizade Haghighi, M. Maeen, and M. Haghparast, *An energy-efficient dynamic resource management approach based on clustering and meta-heuristic algorithms in cloud computing iaas platforms: Energy efficient dynamic cloud resource management*, Wireless Personal Communications, 104 (2019), pp. 1367–1391.

[8] J. Aswini, B. Yamini, R. Jatothu, K. S. Nayaki, and M. Nalini, *An efficient cloud-based healthcare services paradigm for chronic kidney disease prediction application using boosted support vector machine*, Concurrency and Computation: Practice and Experience, 34 (2022), p. e6722.

[9] A. Belgacem and K. Beghdad-Bey, *Multi-objective workflow scheduling in cloud computing: trade-off between makespan and cost*, Cluster Computing, 25 (2022), pp. 579–595.

[10] G. Bindu, K. Ramani, and C. S. Bindu, *Optimized resource scheduling using the meta heuristic algorithm in cloud computing*, IAENG International Journal of Computer Science, 47 (2020), pp. 360–366.

[11] R. Z. Frantz, S. Sawicki, F. Roos-Frantz, F. P. Basso, B. Zucoloto, and R. M. Pillat, *On the analysis of makespan and performance of the task-based execution model for enterprise application integration platforms: An empirical study*, Software: Practice and Experience, 52 (2022), pp. 1717–1735.

[12] R. Garg, M. Mittal, and L. H. Son, *Reliability and energy efficient workflow scheduling in cloud environment*, Cluster Computing, 22 (2019), pp. 1283–1297.

[13] R. Ghafari, F. H. Kabutarkhani, and N. Mansouri, *Task scheduling algorithms for energy optimization in cloud environment: a comprehensive review*, Cluster Computing, 25 (2022), pp. 1035–1093.

[14] T. K. Ghosh, K. G. Dhal, and S. Das, *Comparative study of some nature-inspired meta-heuristics for task scheduling in a computational grid system*, in Novel Research and Development Approaches in Heterogeneous Systems and Algorithms, IGI Global, 2023, pp. 1–15.

[15] I. M. Ibrahim et al., *Task scheduling algorithms in cloud computing: A review*, Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12 (2021), pp. 1041–1053.

[16] B. Jamil, H. Ijaz, M. Shojafar, K. Munir, and R. Buyya, *Resource allocation and task scheduling in fog computing and internet of everything environments: A taxonomy, review, and future directions*, ACM Computing Surveys (CSUR), 54 (2022), pp. 1–38.
[17] H. Jia, X. Peng, and C. Lang, *Remora optimization algorithm*, Expert Systems with Applications, 185 (2021), p. 115665.
[18] J. Kakkottakath Valappil Thekkepuryil, D. P. Suseelan, and P. M. Keerikkattil, *An effective meta-heuristic based multi-objective hybrid optimization method for workflow scheduling in cloud computing environment*, Cluster Computing, 24 (2021), pp. 2367–2384.
[19] S. Kaur, P. Bagga, R. Hans, and H. Kaur, *Quality of service (qos) aware workflow scheduling (wfs) in cloud computing: A systematic review*, Arabian Journal for Science and Engineering, 44 (2019), pp. 2867–2897.
[20] J. K. Konjaang and L. Xu, *Meta-heuristic approaches for effective scheduling in infrastructure as a service cloud: A systematic review*, Journal of Network and Systems Management, 29 (2021), pp. 1–57.
[21] A. A. Laghari, H. He, A. Khan, N. Kumar, and R. Kharel, *Quality of experience framework for cloud computing (qoc)*, IEEE Access, 6 (2018), pp. 64876–64890.
[22] A. A. Laghari, A. K. Jumani, and R. A. Laghari, *Review and state of art of fog computing*, Archives of Computational Methods in Engineering, (2021), pp. 1–13.
[23] N. Manikandan, N. Gobalakrishnan, and K. Pradeep, *Bee optimization based random double adaptive whale optimization model for task scheduling in cloud computing environment*, Computer Communications, 187 (2022), pp. 35–44.
[24] M. Masdari, F. Salehi, M. Jalali, and M. Bidaki, *A survey of pso-based scheduling algorithms in cloud computing*, Journal of Network and Systems Management, 25 (2017), pp. 122–158.
[25] ———, *A survey of pso-based scheduling algorithms in cloud computing*, Journal of Network and Systems Management, 25 (2017), pp. 122–158.
[26] R. Medara and R. S. Singh, *Energy efficient and reliability aware workflow task scheduling in cloud environment*, Wireless Personal Communications, 119 (2021), pp. 1301–1320.
[27] S. K. Mishra and R. Manjula, *A meta-heuristic based multi objective optimization for load distribution in cloud data center under varying workloads*, Cluster Computing, 23 (2020), pp. 3079–3093.
[28] A. Mohammadzadeh, M. Masdari, and F. S. Gharehchopogh, *Energy and cost-aware workflow scheduling in cloud computing data centers using a multi-objective optimization algorithm*, Journal of Network and Systems Management, 29 (2021), pp. 1–34.
[29] N. Mohanapriya, G. Kousalya, P. Balakrishnan, and C. Pethuru Raj, *Energy efficient workflow scheduling with virtual machine consolidation for green cloud computing*, Journal of Intelligent & Fuzzy Systems, 34 (2018), pp. 1561–1572.
[30] A. A. Nasr, N. A. El-Bahnasawy, G. Attiya, and A. El-Sayed, *Cost-effective algorithm for workflow scheduling in cloud computing under deadline constraint*, Arabian Journal for Science and Engineering, 44 (2019), pp. 3765–3780.
[31] M. Nematpour, H. Izadkhah, and F. Mahan, *Enhanced genetic algorithm with some heuristic principles for task graph scheduling*, The Journal of Supercomputing, 79 (2023), pp. 1784–1813.
[32] T. Nguyen, K. Doan, G. Nguyen, and B. M. Nguyen, *Modeling multi-constrained fog-cloud environment for task scheduling problem*, in 2020 IEEE 19th international symposium on network computing and applications (NCA), IEEE, 2020, pp. 1–10.
[33] A. Ramathilagam and K. Vijayalakshmi, *Workflow scheduling in cloud environment using a novel metaheuristic optimization algorithm*, International Journal of Communication Systems, 34 (2021), p. e4746.
[34] M. H. Shirvani, *A hybrid meta-heuristic algorithm for scientific workflow scheduling in heterogeneous distributed computing systems*, Engineering Applications of Artificial Intelligence, 90 (2020), p. 103501.
[35] P. Singh, M. Dutta, and N. Aggarwal, *A review of task scheduling based on meta-heuristics approach in cloud computing*, Knowledge and Information Systems, 52 (2017), pp. 1–51.
[36] A. Taghinezhad-Niar, S. Pashazadeh, and J. Taheri, *Energy-efficient workflow scheduling with budget-deadline constraints for cloud*, Computing, (2022), pp. 1–25.
[37] X. Wei, *Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing*, Journal of Ambient Intelligence and Humanized Computing, (2020), pp. 1–12.
[38] A. M. Yadav, K. N. Tripathi, and S. Sharma, *An enhanced multi-objective fireworks algorithm for task scheduling in fog computing environment*, Cluster Computing, (2022), pp. 1–16.
[39] X. Zhang, T. Wu, M. Chen, T. Wei, J. Zhou, S. Hu, and R. Buyya, *Energy-aware virtual machine allocation for cloud with resource reservation*, Journal of Systems and Software, 147 (2019), pp. 147–161.

Edited by: SCPE Editor-in-chief
Regular paper
Received: May 17, 2024
Accepted: Dec 13, 2024