An AHP based Task Scheduling and Optimal Resource Allocation in Cloud Computing

Syed.Karimunnisa[1], Yellamma Pachipala[2]
Research Scholar[1], Associate Professor[2]
Department of Computer Science and Engineering Koneru Lakshmaiah Education Foundation Vaddesvaram,
AP, India-522502

***Abstract*—Cloud systems by virtue characterize ultimate resource utilization with ever evolving user requirements facilitating adaptivity. With a scope of enhancing the QoS needs of user applications, numerous factors are considered fortunning among which Task scheduling promises to grab focus. The Task Scheduling mechanism ascertains improvement by distributing the subtasks to specific set of resources pertaining to prevailing Quality models. The work emphasizes the need for effective task scheduling and optimizing resource allocation by modelling a modified AHP (Analytical Hierarchy Process) driven approach. The proposed method guarantees the functionality in two phases pertaining to Task ranking and pipelined with Optimized scheduling algorithms resulting in maximization of resource utilization. The former phase of task ranking is aided by improved AHP with substantial usage of fuzzy clustering followed by an enhanced CUCMCA (Chimp Updated and Cauchy Mutated Coot Algorithm) algorithm for optimal resource allocation of cloud applications. The contributed model promises leveraged performance of 32% for memory usage, 33.5% for execution time, 29% for makespan and 18% for communication cost over pre-existing conventional models considered.**

***Keywords*—Task scheduling; AHP; TS; QoS; optimization; CUCMCA**

NOMENCLATURE

Abbreviation	Description
QoS	Quality of Service.
RM	Resource management.
CESS	Cross-Entropy based stochastic scheduling.
OP-MLB	Online VM Prediction based Multi-objective Load Balancing.
MESA	Migration enabled scheduling algorithm.
MINLP	Mixed integer non-linear programming.
LCS	Learning classifier systems.
SLA	Service level agreement.
VM	Virtual machine.
HASRA	Hotspot aware server relocation algorithm.
PM	Physical machine.
HAWDA	Hotspot adaptive workload deployment algorithm.
CSO	Cat swarm optimization.
RMFW	Resource Management Framework for multiple online Scientific workflows.
HGCSBAT	Hybrid cat swarm bat algorithm.
TS	Task scheduling.

I. INTRODUCTION

Cloud computing is a flexible strategy for exchanging distributed services and resources with the user wherever and anytime they require [1]. Because of its scalable user pool of services and resources, cloud computing has become more popular in recent years. This is because it allows users to freely control their consumption and only pay for the cloud resources they actually utilize [10] [28]. Google Compute Engine, Rackspace Cloud, and Amazon EC2 are just a few of the commercial cloud computing systems that have recently entered the market [2] [15]. Moreover, for hosting and delivering software solutions for many industrial applications, cloud computing has established itself as a dependable, affordable, and scalable service option [14] [12]. Still, a cloud must have sufficient capacity to meet peak user demand in order to uphold user expectations for QoS [19] [16].

Resource usage, server power consumption, and storage all play significant roles in the cloud environment, making resource management essential. While provisioning and allocating cloud resources, availability was frequently used as the determining factor without considering the other essential factors like resource utilization or the server's thermal properties [9]. A cloud system's ability to manage its resources autonomously and adaptively based on the workload changes is known as autonomous RM [20]. Elastic resource management encompasses a wide range of processes, including balanced virtual machine and application scheduling, server over/under-load control utilizing VM migration, etc. [3] [17]. One or more physical machine resources, including CPU, memory, I/O, and network bandwidth, may be overloaded due to the increased load of virtual machine operations [5] [8].

The scheduling solution was optimized in terms of each metric specified in the QoS model using a QoS-driven CESS method [4]. It has been suggested to use an OP-MLB system, which combines a number of algorithms that cooperate to provide effective resource management for cloud environments [6] [7]. With more effective storage and V/F scaling improvement, the EARU model significantly reduces LLC disappointments and thus more effectively utilizes asset. In terms of CPU utilization, preparation time, and energy output, it also achieves preferred execution to the board's current asset management plan [11]. Greedy method named MESA is recommended due to the high computing complexity of addressing the MINLP problem [18] in order to arrive at the best solution [13]. Also, advanced metaheuristic models are in need to proceed with optimal scheduling process.

This paper introduces a new optimization assisted task scheduling, and the main contributions are as follows:

- Initially, modified AHP process is introduced for ranking the task.
- A hybrid optimization model, namely, CUCMCA method for optimal scheduling of task with appropriate allocation of resources to execute the task.
- The proposed method is implemented using CloudSim simulator.

The work progresses initiating with intense literature review in Section II paving path for improvisation issues addressed with modified AHP based task ranking and hybrid optimal resource allocation provided in Section III followed by proposed CUCMCA Algorithm in Section IV formulates results and discussions in Section V Section VI contributes to conclusion of work with a wide overview of work.

II. Related Work

Several Researchers have contributed innumerable solutions addressing issues of scheduling and resource allocation. Despite, leaving few coins unturned that are addressed in our work with enhancements improvising performance.

In 2020, Mahdi Abbasi et al. [1] presented two approaches, XCS and BCM-XCS, depending on XCS - LCS, to manage the network's edge power consumption and lessen workload delay. The outcomes of this tests show that BCM-XCS is superior to the standard XCS-based approach. The workloads were distributed using the suggested approaches in a way that both the communication and processing delay among cloud and fog nodes were kept to a minimum. Additionally, the suggested approaches can recharge the reusable batteries utilized at the network's edge 18% faster than the existing technique.

In 2020, Yunliang Chen et al. [2] suggested a detailed QoS model to evaluate the performance level of data center clouds. To improve the cumulative QoS and sojourn time of all activities, an enhanced CESS algorithm was created. According to experimental findings, this approach outperforms the baseline algorithm in terms of accumulative QoS as well as sojourn duration by up to 56.1% and 25.4%, correspondingly. The algorithm's duration only increases linearly as more Cloud data centres and workloads are added. This technique constantly develops scheduling solutions with acceptable QoS without compromising sojourn time when the arrival rate as well as service rate ratio are kept constant.

In 2022, M. Hasan Jamal et al. [6] suggested a HAWDA and HASRA depending on thermal profiling considering outlet temperature detection. In order to reduce the peak output temperatures, HAWDA distributed workload on servers in a thermally efficient manner, while HASRA optimized server positioning in thermal hotspot areas. To evaluate the effectiveness of HAWDA against the TASA and GRANITE methods, performance comparison is done. Results showed that HAWDA, which reduces peak outlet temperature, achieved average peak server utilization comparable to GRANITE as well as TASA without adding additional load to the cooling system, with or without server relocation.

In 2022, M. Hasan Jamal et al. [6] suggested a HAWDA and HASRA depending on thermal profiling considering outlet temperature detection. In order to reduce the peak output temperatures, HAWDA distributed workload on servers in a thermally efficient manner, while HASRA optimized server positioning in thermal hotspot areas. To evaluate the effectiveness of HAWDA against the TASA and GRANITE methods, performance comparison is done. Results showed that HAWDA, which reduces peak outlet temperature, achieved average peak server utilization comparable to GRANITE as well as TASA without adding additional load to the cooling system, with or without server relocation.

In 2020, Ali Asghari et al. [7] developed a new architecture made up of many cooperating agents that took into account all aspects of TS and resource provisioning and managed the QoS offered to users. The integrated model that was suggested included all processes for TS and resource provisioning, and its many components help with managing user applications and making better use of cloud resources. This framework performs effectively with concurrent dependent activities, which have a challenging scheduling procedure due to the dependency of their subtasks.

In 2021, A. M. Senthil Kumar et al. [8] suggested a new task allocation method employing BAT and CSO method. The BAT algorithm aids the CSO algorithm in overcoming a pre convergence problem. The suggested HGCSBAT algorithm's performance was assessed and contrasted with that of the well-known CSO & BAT methods. In regards to availability & throughput, HGCSBAT performs better than that of the BAT, Cat Swarm Optimization, & Genetic algorithms. Traditional work scheduling algorithms features and limitations are given in Table I.

TABLE I. FEATURES AND LIMITATIONS OF TRADITIONAL TASK SCHEDULING ALGORITHMS

Author [citation]	Methodology	Features	Limitations
Mahdi Abbasi	XCS and BCM-XCS	Processing delay gets minimized	Workload latency is increased
Yunliang Chen	CESS method	Minimize waiting time	QoS is need to be upgraded for responsibilityexecution
Deepika Saxena	OP-MLB Method	Less power Consumption	Need for cloud data centre's performanceimprovement
Uma Tadakamalla	FogQN-AC	optimized cost average response time	Unplanned resources demand
Lei Yu	Stochastic Load Balancing approach	Migration cost is minimized	It is crucial to evaluate how different workload distributions affect load balancing performance.
M. Hasan Jamal	HAWDA and HASRA	Low memory usage	Fairer resource allocation needs a moreaccurate approach.
Ali Asghari	RMFW method	Reduced resource utilization	Higher computation cost
A. M. Senthil Kumar	HGCSBAT	Increased throughputand availability	Needs to consider balancing problem

In 2023 K.Pradeep, Sharma, and Jishnu[29] proposed an intense review on Task scheduling parameters which shed light on various strategies that make way for efficient scheduling with fault tolerant approaches in Fog computing. Their work emphasizes on tuning QoS parameters for improved results.

III. MODIFIED AHP BASED TASK RANKING AND HYBRID OPTIMAL SCHEDULING WITH APPROPRIATE RESOURCE ALLOCATION

Fig. 1 depicts the scenarios of Task Scheduling where user requests assemble at various virtual machines pertaining to a physical machine. These VMs promise improved performance by scheduling the tasks using our proposed approach. Scheduling of Tasks is performed in two phases i.e., Task ranking using FCM clustering followed by CUCMCA optimization for enhanced results.

A. System Model

Considering a data center with M servers include servers $A \in \{A_1, A_2, ..., A_M\}$, here, various VM types are purchased by Q users for executing the applications on U VM's includes $vm \in \{vm_1, vm_2, ..., vm_U\}$. Assume application R_Q pertaining to Q^{th} number of users depicted as $\{Tsk_1, Tsk_2, ..., Tsk_Z\} \in R_Q$, here Tsk_z represents the application task. The tasks are scheduled according to their resource requirements, which chooses the best VM for i^{th} task (Tsk_i^{res}) execution, where res defines resources such as memory, CPU, etc. as well as $vm_A^{res}, vm_Q^{res}, vm_L^{res}, vm_{XL}^{res}$ were small, medium, large as well as extra-large VM sets. The number of VM types available at a specific data center can be expanded. If need for resources of i^{th} task (Tsk_i^{res}) is equal or lesser to capacity of resource of vm_A, then smaller VM types were given to it. The proposed workload scheduling and resource management is progressed with two different steps:

1) Task ranking: This phase handles the process for ranking the tasks as per their priorities by first identifying priorities and generating Task queues pertaining to priority groups.

2) Optimal scheduling: Optimal scheduling phase deals with the assignment of corresponding resources according to the constraints resulting in better performance.

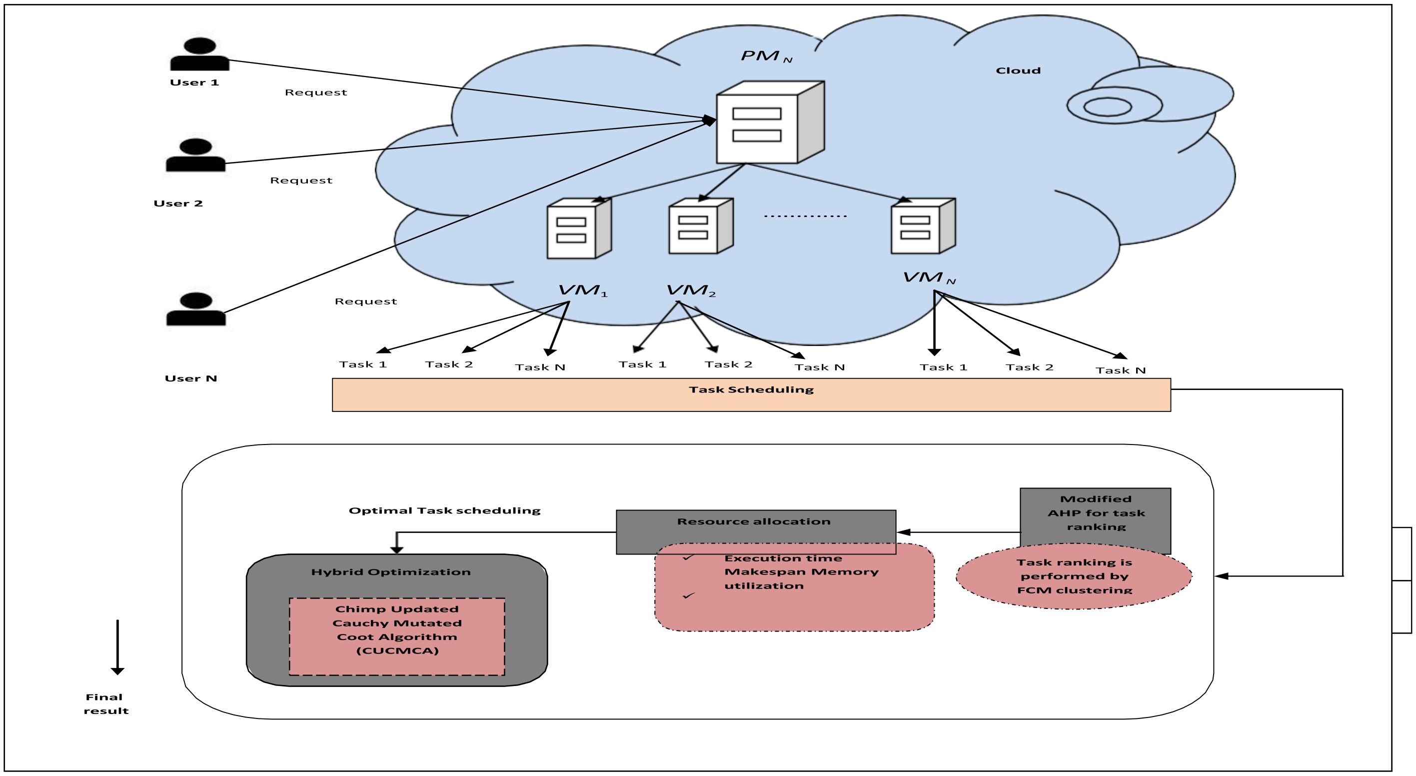

Fig. 1. Task scheduling in proposed approach.

B. *Task Ranking*

This is the first stage, where the ranking of task was done by using modified AHP (Analytical Hierarchy Process). Conventional AHP steps for ranking purpose are as follows:

- Implement the Saaty preference table [23]. Given below is Saaty preference table which offers point scale including descriptors (Table II).

TABLE II. SAATY TABLE

Points	Descriptor
1	Conditions were equally important
3	First condition is slightly more significant than second condition
5	First condition is rather more significant than second condition.
7	Obviously, the first condition is more significant than the second one.
9	The first condition is unquestionably more significant than the

- Afterwards each column summation and each column normalization and weighted sum is computed [24].
- Based on the Saaty preference table, AHP ranking is performed.

According to recommended concept, modified AHP process is followed for ranking purpose. Modified AHP steps are as follows:

- Instead of using the Saaty preference table, in modified AHP, FCM (Fuzzy c-Means Clustering) is followed on the basis of 1-9 clustering for implementing the table. A data collection is divided into N clusters using the FCM data clustering method, with each cluster having some of the data points in the dataset.
- Based on the FCM, AHP ranking is performed for task scheduling. Modified AHP based ranking is shown in Fig. 2.

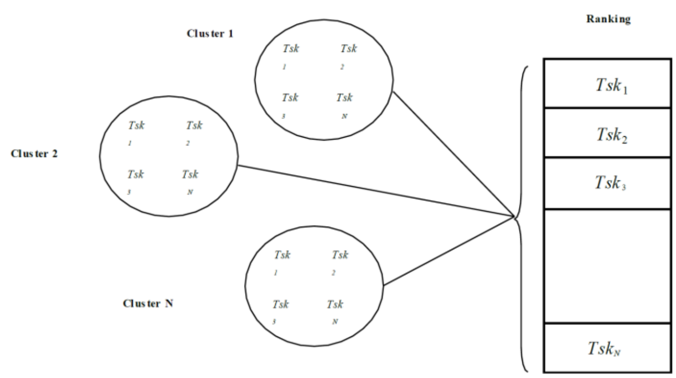

Fig. 2. Modified AHP based ranking paradigm.

C. *Optimal Task Scheduling and Resource Management*

In this step, the process of optimal scheduling is done by selecting the optimal PM and VM to execute the tasks. During this process, certain constraints are evaluated to ensure the scheduling in a precise manner. Also, the constrains involve in the assignment of corresponding resources ∇ to execute the tasks involved.

Constraints involved in this optimal scheduling process are as follows:

- Execution time
- Make span
- Memory utilization
- Communication cost

Optimal assignment of PM and VM for task scheduling: In this stage, optimal assignment of PM and VM for task scheduling is done by hybrid optimization combines Coot and Chimp optimization algorithms which is detailed in upcoming section. The factors used for optimal TS were:

- **Make span** ($F1$): Make span is described as the whole amount of time needed to complete the task. Make span is defined in eq. (1), where m denotes VM count, n denotes task count, Tsk denotes task, len denotes the Tsk size in MI (Million Instructions), and $pesnum$ denotes PE (processing Element) in VM.

$$Makespan = Maxi_{1 \le i \le m}\{fTsk_i\} \quad (1)$$

- **Communication Cost** ($F2$): Here, communication cost among task i and task j are given to different VM's.

  Communication cost Com_Cost is calculated in eq. (2).

$$Com\_Cost = \sum_{Tsk_i=1}^{n} Com\_Cost(Tsk_i, i = 1,2, \ldots n) \quad (2)$$

- **Execution time ($F3$):** The time duration needed by VM to finish each task is known as execution time.

- **Memory Utilization ($F4$):** Memory Utilization is a measure of average memory usage that is calculated by averaging the percentage of memory space that is being used at any given time across the reporting interval. Memory Utilization is defined in eq. (3).

$$Mem - Util = \left(100^*(TotalMBytes - AvailMBytes/TotalMBytes)\right) \quad (3)$$

$$TotalMBytes \rightarrow vm(Mem)$$

Here,

$$AvailMem \rightarrow (VirtualMem - TaskMem)$$

Solution encoding: The selection process will be decided by the proposed optimization algorithm where the solution including both PM and VM set from which the model selected corresponding PM and VM to execute the respective task. Fig. 3 gives solution encoding ofproposed CUCMCA method.

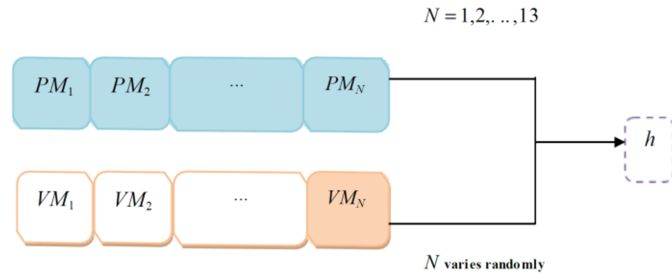

Fig. 3. Solution encoding of proposed CUCMCA method.

Weighted Objective Function: Minimization: The objective function obj defined in the model is given in eq. (4), where $w1$, $w2$, $w3$, $w4$ are the weights assigned to each parameter.

These weights are calculated using chaotic cubic map function.

$$obj = w1^*F1 + w2^*F2 + w3^*F3 + w4^*F4 \quad (4)$$

Cubic map: The cubic map is one of the maps that are most frequently used to generate chaotic sequences in many applications. In eq. (5), cubic map is defined.

$$L_{y+1} = 259 \times L_y\left(1 - L_y^2\right) \quad (5)$$

IV. Chimp Updated Cauchy Mutated Coot Algorithm (CUCMCA) Optimization

Hybrid optimizations are a new class of optimization methods that we develop to solve the optimization issue with more convergence efficiency. For hybrid optimizations, two or more algorithms must have been used for the same optimization. In this paper, we hybridized two algorithms named coot and chimp optimization. The solution update is done by this hybridized algorithm. Here the random number K^2 is estimated by using Tent map function. Also, Cauchy mutation is introduced in our proposed concept. Our proposed hybrid algorithm concept is given below:

Small water birds called Coots [21] were the rail family members. In coot optimization, with the formula (6), a small area is used to produce the population at random, where $Coot^{pos(i)}$ represents the coot position, g represents the variables count, and ub,lb represents the upper as well as lower bound of search space.

$$C^{pos(i)} = rnd(1, O)^*(ub - lb) + lb \quad (6)$$

Random motion to this direction and that direction: Different areas of the search space are explored by coot migration. This movement will let the algorithm escape the local optimal if it becomes stuck in the local optimal. Coot's new position is determined in eq. (7).

$$C^{pos(i)} = C^{pos(i)} + G \times K^2 \times \left(Z - C^{pos(i)}\right)$$

$$Z = rnd(1, O)^*(ub - lb) + lb \quad (7)$$

Where, K^2 represents the random number which is calculated using tent map function according to proposed model, $G = 1 - Y \times \left(\frac{I}{iter}\right)$ where Y represents the current iteration, $iter$ represents the maximum iteration.

Tent map: Tent chaotic map is also refereed as the logistic map represents particular chaotic effects. The following equation (8) gives the definition of this map:

$$q_{r+1} = \begin{cases} 2qr, qr < 0.5 \\ 2(1-q_r), q_r \geq 0.5 \end{cases} \tag{8}$$

Chain movement: It is possible to construct chain movement by using the average position of 2 coots. The formula (9) is used to calculate the coot's new position, where $C^{Pos}(i-1)$ represents the second coot.

$$C^{Pos(i)} = 0.5 \times \left(C^{Pos}(i-1) + C^{Pos}(i)\right) \tag{9}$$

Adjusting position in accordance with the group leaders: To carry out this movement, a system is deployed based on the formula (10) to choose the leader, where i represents current coot index, D represents leader index, and lc represents the leader count.

$$D = 1 + (iMODlc) \tag{10}$$

Depending on leader O , $C^{Pos}(i)$ update its position. Formula (11) uses the chosen leader to determine the coot's subsequent position, where $C^{Pos}(i)$ represents new coot position, $L^{pos}(i)$ represents selected leader position, and *I1*, *I* represents the random number.

$$C^{Pos}(i) = L^{Pos}(O) + 2 \times I1 \times cos(2I\pi) \times \left(L^{Pos}(O) - C^{Pos}(i)\right) \tag{11}$$

According to proposed model, position update is done by hybridizing Coot and Chimp position [22] which is specified in eq. (12) to eq. (16).

Proposed update equation: $h_{chimp} = h_{prey}(t) - u.v$

Substitute $L^{Pos} = LP, h_{prey} = h, C^{Pos} = h$

$$h - u.v = LP(O) + 2 \times I1 \times \cos(2I\pi) \times LP(O) - C^{Pos}(i) \tag{12}$$

$$h - u.v = LP + 2I1 \times \cos(2I\pi) \times (LP - h) \tag{13}$$

$$h - u.v = LP + 2I1 \times \cos(2I\pi) \times LP - 2I1 \times \cos(2I\pi) \times h \tag{14}$$

$$h - u.v = LP(1 + 2I1 \times \cos(2I\pi) - 2I1 \times \cos(2I\pi) \times h \tag{15}$$

$$h = LP(1 + 2I1 \times \cos(2I\pi) + u.v - 2I1 \times \cos(2I\pi) \times h \tag{16}$$

Leader movement: Leaders need to change their location with respect to the objective in order to move the group toward an objective (the ideal region). It is advised to update leader position using formula (17), where $OBest$ represents the best position, $J3$, $J4$ represents the random number.

$$L^{Pos(i)} = \begin{cases} S \times I3 \times cos(2I\pi) \times \begin{pmatrix} FBest - L^{Pos(i)} \\ +FBest, \; I4 < 0.5 \end{pmatrix} \\ S \times I3 \times cos(2I\pi) \times \begin{pmatrix} FBest - L^{Pos(i)} \\ -FBest, \; I4 \geq 0.5 \end{pmatrix} \end{cases} \tag{17}$$

Here, $S = 2 - R\left(\frac{1}{S_t}\right), S^t$ represents maximum iteration, and R represents the current iteration.

Cauchy mutation: Cauchy mutation is also employed in this algorithm to produce the solution. Due to its wider search range, Cauchy mutation has a significant ability to seek globally. This ensures the high convergence rate. Below Algorithm 1 gives the pseudocode of suggested CUCMCA model:

Algorithm 1: Pseudo code of Chimp Updated and Cauchy Mutated Coot (CUCMCA)

Input: Randomly initialize coot population (Tasks and VMs)
Output: Optimally mapped VMs and Tasks
Parameter initialization $\Pr = 0.5$, Lc, Cnt_{coot} (coot count)
$Cnt_{coot} = Cnt_{pop} - Cnt1$
Randomly select the leader of coot
Coot as well as leader fitness calculation
Identify best leader or coot as $FBest$
While end condition is not met
Calculate t, S parameter
if $rnd < \Pr$
$I, I1, I3$ were random numbers along the problem dimension
Else
$I, I1, I3$ were random number
end if
$I, I1, I3$ were random number
for $i = 1$ to Cnt_{coot}
Evaluate parameter of D
if $rnd > 0.5$
Position update using new evaluation given in eq. (16)
Else
if $rnd < 0.5i$ ~=1
update coot position by eq. (7)
In eq. (7), random number K^2 is calculated using the Tent map as per proposed model
end if
end if
Evaluate fitness using eq. (4)
if $C^{fitness} < L^{fitness}(O)$
$temp = L(O)$
$L(O) = coot$

$coot = temp$

end if

for leader count

Leader position update by eq. (17.1)

Else

Leader position update by eq. (17.2)

end for

if $L^{fitness} < FBest$

$temp = FBest$

$FBest = L$

$L = temp$

end if

end for

$S^t = S^t + 1$

Cauchy mutation is performed for global search

end while

V. Results and Discussion

A. Simulation Procedure

The proposed Chimp Updated and Cauchy Mutated Coot Algorithm (CUCMCA) method for task Scheduling and Resource Management was done in Cloudsim. The dataset considered for our work are extracted from internet sources i.e. google cluster traces 2019. The assessment was done on Bald Eagle Search (BES), Arithmetic Optimization Algorithm Method (AOAM) [26], Moth Flame Optimization (MFO), Hybrid Swarm Optimization (HSO) [27], Elephant Herding Optimization (EHO), Chimp and COOT, regarding Communication Cost, Execution Time, Fitness, Makespan and Memory Utilization. Also, it was examined by altering the number of virtual machines to 10, 20, 30, 40 and 50.

B. Dataset Description

This is a trace of the workloads running on eight Google Borg compute clusters for the month of May 2019 [25]. The trace describes every job submission, scheduling decision, and resource usage data for the jobs that ran in those clusters.

It builds on the May 2011 trace of one cluster, which has enabled a wide range of research on advancing the state-of-the-art for cluster schedulers and cloud computing, and has been used to generate hundreds of analyses and studies.

Since 2011, machines and software have evolved, workloads have changed, and the importance of workload variance has become even clearer. The new trace allows researchers to explore these changes.

The new dataset includes additional data, including:

- CPU usage information histograms for each 5 minute period, not just a point sample;
- Information about allow sets (shared resource reservations used by jobs); and
- Job-parent information for master/worker relationships such as MapReduce jobs.

Just like the last trace, these new ones focus on resource requests and usage, and contain no information about end users, their data, or access patterns to storage systems and other services.”

C. Evaluation of Communication Cost

The proposed CUCMCA method is compared to extant systems in terms of communication cost. Fig. 4 depicts the study of communication cost. A variety of Virtual Machines, including 10, 20, 30, 40 and 50 are evaluated. A successful system should have minimal communication costs. On examining the communication cost, the values gained by recommended method are considerably lower than other models. That is, the communication cost of (~) 125 is obtained by adopted method at the 40th VM. In contrast, the compared models like BES, AOAM, MFO, HSO, EHO, Chimp and COOT has obtained relatively higher communication cost of 174, 176, 132, 177, 175, 163 and 164, respectively. While compared to other schemes such as BES, AOAM, MFO, HSO, EHO, Chimp and COOT, the communication cost achieved using suggested approach is the smallest in the 50th VM. Thus, the outcomes of the experiment reveal that the proposed CUCMCA method's communication cost score is preferable to the established approaches.

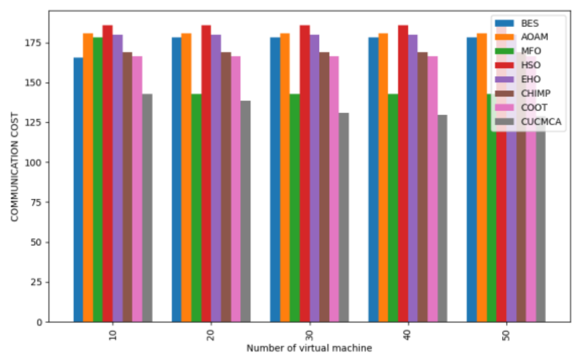

Fig. 4. Communication cost analysis: proposed CUCMCA method vs conventional models.

D. Evaluation on Execution Time

In this section, the analysis on Execution Time is examined for varied VMs. The analysis on developed CUCMCA method over BES, AOAM, MFO, HSO, EHO, Chimp and COOT for varied VMs is exposed in Fig. 5. In order to improve the system's performance, the execution time should be reduced. The developed approach holds minimal execution time of 8.24 at the VM 50; whereas, the traditional models holds the highest execution time for BES (39.53), AOAM (47.92), MFO (49.67), HSO (62.18), EHO (18.65), Chimp (48.85) and COOT (53.67), respectively. Moreover, it is observed that, the suggested method has accomplished better outcomes at 30th VM than at 10th and 20th VMs. As a result, the superiority of the suggested CUCMCA work is proved.

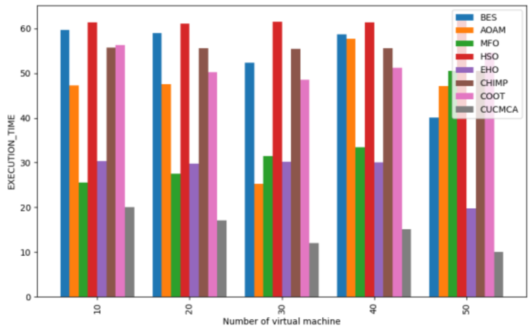

Fig. 5. Execution time analysis: proposed CUCMCA method vs conventional models.

E. Evaluation of Fitness

The Fitness analysis of the proposed CUCMCA classifier is computed over the existing classifiers and the graphical illustration is represented in Fig. 6. Further, the Fitness of the suggested model attains better outcomes than other conventional approaches. The models like BES, AOAM, MFO, HSO, EHO, Chimp and COOT acquired the highest fitness of (~) 28.69, 23.54, 20.71, 28.67, 17.46, 26.82 and 25.18, whilst the proposed strategy yielded the lowest fitness of 17.82, at the 30th VM. Likewise, the fitness of the adopted approach obtained the better value of (~) 14.28; however, the existing schemes like BES, AOAM, MFO, HSO, EHO, Chimp and COOT holds the lowest values for the VM 40. Hence, the improvement of the suggested CUCMCA model is established over others in terms of fitness.

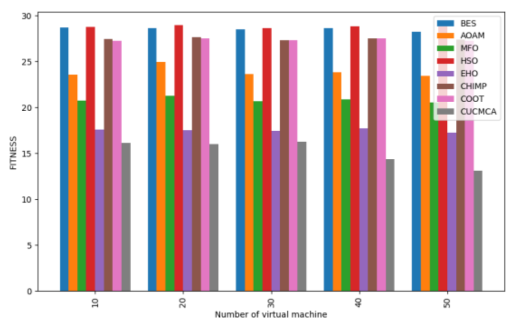

Fig. 6. Fitness analysis: proposed CUCMCA method vs conventional models.

F. Evaluation of Makespan Time

The Makespan using suggested CUCMCA method is analyzed over BES, AOAM, MFO, HSO, EHO, Chimp and COOT for varied VMs as shown in Fig. 7. Low makespan is required for enhanced system performance. The suggested method successfully achieves our goal, whose Makespan values exceed the traditional approaches. Moreover, the Makespan of the developed model attains lower value of 43.56, in the 20th VM than other existing classifiers like BES (82.67), AOAM (68.42), MFO (76.84), HSO (63.45), EHO (81.96), Chimp (80.87) and COOT (78.12), respectively. This analysis shows that the developed CUCMCA approach makes the system more robust at scheduling workloads and managing resource than the conventional approaches.

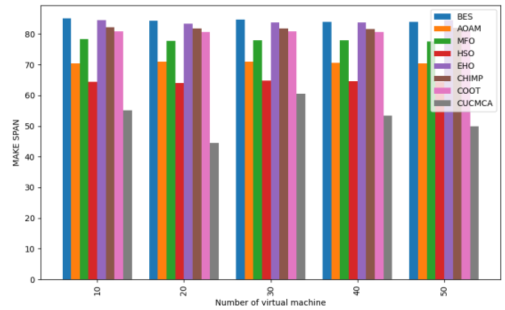

Fig. 7. Makespan analysis: proposed CUCMCA method vs conventional models.

G. Analysis on Memory Utilization

The performance of adopted CUCMCA model regarding Memory Utilization is displayed in the Fig. 8. The memory utilization should be minimal for better system performance. In this manner, it is observed that the adopted model achieves least memory utilization when compared to models like BES, AOAM, MFO, HSO, EHO, Chimp, and COOT, respectively. Particularly, incredible outcomes for both proposed and current approaches have been obtained for all measures at the 10th, 30th, and 50th VM. Nevertheless, the developed approach has delivered more determinative outcomes than distinct strategies for every VM. For instance, at 50th VM, the memory utilization of suggested approach is 0.04, which is better than the values obtained for existing schemes like BES is 0.05, AOAM is 0.06, EHO is 1.2, COOT is 1.1 and HSO is 0.07, respectively. As a consequence, this assessment proves that the proposed CUCMCA model is better to make an efficient workload scheduling when compared to other conventional models.

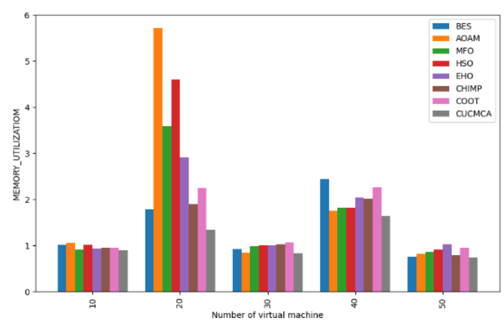

Fig. 8. Memory utilization analysis: proposed CUCMCA method vs conventional models.

H. Convergence Analysis

The convergence study of the proposed CUCMCA work is contrasted to the traditional methods like BES, AOAM, MFO, HSO, EHO, Chimp and COOT are shown in Fig. 9. In addition, it was examined by altering the iterations to 5, 10, 15, 20 and 25, respectively. When comparing the suggested method to other extant schemes, the findings show that the developed model has the lowest error rate. Here, the COOT algorithm has exhibited the worst performance in the initial (0th) iteration. From iteration 5 through iteration 25, the proposed approach and other current classifiers have lower error rates. Nevertheless, at the last 25th iteration, the adopted approach recorded the lowest error rate of (~) 1.0. Thus, recommended strategy resulted in a slightly lower error rate than BES, AOAM, MFO, HSO, EHO, Chimp and COOT. Therefore, the proposed CUCMCA strategy is appropriate for the workload scheduling and resource management. The improvisation in the performance of proposed logic shows the impact of proposed hybrid algorithm in enhancing the convergence rate and speed.

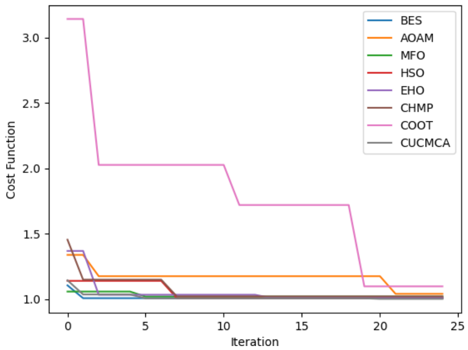

Fig. 9. Convergence analysis: proposed CUCMCA method vs conventional models.

I. Statistical Analysis

Table III represents the statistical analysis with respect to Fitness, Makespan, Communication Cost, Memory Utilization and Execution Time for the proposed CUCMCA method over the established models. Also, the analysis was carried out with five different case scenarios including Mean, Maximum, Standard Deviation, Minimum and Median. The optimization schemes are stochastic, and to substantiate the fair assessment, every model is examined several times. On examining the resultants, the suggested scheme has achieved minimal values for the majority of the scenarios. Based on the mean case scenario analysis, the proposed model obtains lower execution time of 12.06345 than other traditional models like AOAM is 44.99084, MFO is 33.69218, HSO is 61.50506, EHO is 28.04894, Chimp is 54.63884, COOT is 52.19459 respectively. According to the Median analysis, the Memory utilization of the suggested work is 0.892668, which is superior to the existing models like BES, AOAM, MFO, HSO, EHO, Chimp and COOT.

TABLE III. STATISTICAL ANALYSIS WITH RESPECT TO OBJECTIVE FUNCTION

Fitness								
	BES	AOAM	MFO	HSO	EHO	CHIMP	COOT	CUCMCA
Mean	28.54696	23.87702	20.79797	28.80408	17.49941	27.45825	27.3536	15.17077
Maximum	28.70305	24.95545	21.27113	28.96292	17.69324	27.62056	27.50861	16.2401
Standard Deviation	0.153195	0.554059	0.260911	0.119192	0.14309	0.102286	0.125922	1.229137
Minimum	28.27161	23.43405	20.51185	28.60521	17.2719	27.34739	27.22187	13.12047
Median	28.60423	23.61853	20.71881	28.84593	17.47804	27.46037	27.28607	15.99167
Makespan								
	BES	AOAM	MFO	HSO	EHO	CHIMP	COOT	CUCMCA
Mean	84.38679	70.68699	77.88326	64.44666	83.98068	81.87688	80.84271	54.48431
Maximum	85.10643	71.04397	78.3	64.82723	84.51164	82.1506	81.07052	55.07267
Standard Deviation	0.43992	0.27665	0.252149	0.259594	0.427563	0.176513	0.151499	0.35984
Minimum	83.9178	70.32722	77.54787	64.01315	83.39151	81.65603	80.657	53.93727
Median	84.28038	70.64215	77.90364	64.45928	83.84175	81.84449	80.85056	54.4618
Communication Cost								
	BES	AOAM	MFO	HSO	EHO	CHIMP	COOT	CUCMCA
Mean	175.4903	180.7314	149.715	185.774	179.7471	168.8011	166.3097	141.8741
Maximum	178.0185	180.7314	178.0185	185.774	179.7471	168.8011	166.3097	142.6391
Standard Deviation	5.056402	0	14.15177	0	0	0	0	0.38248
Minimum	165.3775	180.7314	142.6391	185.774	179.7471	168.8011	166.3097	141.6829
Median	178.0185	180.7314	142.6391	185.774	179.7471	168.8011	166.3097	141.6829
Memory Utilization								
	BES	AOAM	MFO	HSO	EHO	CHIMP	COOT	CUCMCA
Mean	1.380963	2.036544	1.634815	1.869821	1.586945	1.334175	1.496101	1.08832
Maximum	2.442056	5.718419	3.585442	4.599554	2.915935	2.009994	2.267544	1.637185

Standard Deviation	0.635769	1.871789	1.03697	1.403908	0.780297	0.513353	0.621799	0.342108
Minimum	0.753165	0.812619	0.859829	0.914704	0.940985	0.793372	0.95027	0.74304
Median	1.015725	1.058777	0.993748	1.016745	1.026041	1.025408	1.063376	0.892668
Execution Time								
	BES	AOAM	MFO	HSO	EHO	CHIMP	COOT	CUCMCA
Mean	53.95939	44.99084	33.69218	61.50506	28.04894	54.63884	52.19459	12.06345
Maximum	59.70747	57.66733	50.49382	62.11024	30.40323	55.78705	56.38825	20.06346
Standard Deviation	7.443277	10.623	8.849737	0.337976	4.119379	2.040145	2.887903	6.542089
Minimum	40.02318	25.31515	25.57135	61.08178	19.81941	50.56235	48.49736	1.063507
Median	58.68128	47.28658	31.5239	61.39719	30.01077	55.6951	51.19773	12.06383

VI. Conclusion and Future Scope

Cloud system must be able to manage its resources autonomously and adaptively in response to the changes in workload needs with its vast computing power and flexibility for ever evolving challenges. Despite credibility features, it faces many challenges such as Scheduling, Security, and Energy Management etc. Among aforementioned issues the one of concern that owes to be improved is Task scheduling, intending the maximization of user-favoured application QoS parameters our proposed hybrid algorithm performs task scheduling and allocates resources efficiently in cloud computing environments. Our work considers Google cloud (May 2019) workload traces as input using modified AHP to rank the task via FCM. Furthermore, the optimal task scheduling and resource allocation are done by the developed Chimp Updated and Cauchy Mutated Coot Algorithm (CUCMCA). Deliberately the outcome promises improved results in comparison to the existing conventional BES, AOAM, MFO, HSO, EHO, Chimp, COOT models with respect to makespan time, execution time, communication cost and memory utilization gives improved results. As a scope for further enhancement classifying user tasks prior to scheduling promises improved results in terms of QoS metrics and extends scope for better resource allocation.

References

[1] Mahdi Abbasi, Mina Yaghoobikia, Milad Rafiee, Alireza Jolfaei, Mohammad R. Khosravi,, “Efficient resource management and workload allocation in fog-cloudcomputing paradigm in IoT using learning classifier systems”, Computer Communications, vol.153, 2020

[2] Y. Chen et al., "Stochastic Workload Scheduling for Uncoordinated Datacenter Clouds with Multiple QoS Constraints," in IEEE Transactions on Cloud Computing, vol. 8, no. 4, pp. 1284-1295, 1 Oct.-Dec. 2020, doi: 10.1109/TCC.2016.2586048.

[3] D. Saxena, A. K. Singh and R. Buyya, "OP-MLB: An Online VM Prediction based Multi-objective Load Balancing Framework for Resource Management at Cloud Datacenter," in IEEE Transactions on Cloud Computing, doi: 10.1109/TCC.2021.3059096.

[4] U. Tadakamalla and D. A. Menasce, "Autonomic Resource Management for Fog Computing," in IEEE Transactions on Cloud Computing, doi: 10.1109/TCC.2021.3064629.

[5] L. Yu, L. Chen, Z. Cai, H. Shen, Y. Liang and Y. Pan, "Stochastic Load Balancing for Virtual Resource Management in Datacenters," in IEEE Transactions on Cloud Computing, vol. 8, no. 2, pp. 459-472, 1 April-June 2020, doi: 10.1109/TCC.2016.2525984.

[6] M. Hasan Jamal, M. Tayyab Chaudhry, Usama Tahir, Furqan Rustam, SoojungHur and Imran Ashraf,”Hotspot-Aware Workload Scheduling and Server Placement forHeterogeneous Cloud Data Centers”, 2022

[7] Ali Asghari, Mohammad Karim Sohrabi, FarzinYaghmaee,”A cloud resource management framework for multiple online scientific workflows using cooperative reinforcement learning agents”, Computer Networks, vol. 179, 2020

[8] Senthil Kumar, A.M., Padmanaban, K., Velmurugan, A.K. et al. A novel resource management framework in a cloud computing environment using hybrid cat swarm BAT (HCSBAT) algorithm. Distrib Parallel Databases (2021) https://doi.org/10.1007/s10619-021-07339-w

[9] Sukhpal Singh Gill, et al. “ThermoSim: Deep learning-based framework for modeling and simulation of thermal-aware resource management for cloud computing environments”, Journal of Systems and Software, vol.166, 2020

[10] Neda Khorasani, SaeidAbrishami, Mehdi Feizi, Mahdi AbolfazliEsfahani, FaezeRamezani,"Resource management in the federated cloud environment using Cournotand Bertrand competitions”, Future Generation Computer Systems, vol.113, 2020

[11] Murgesh V. Jambigi, Dr. M.V. Vijay Kumar, Dr. D.V. Ashoka, Dr. Prabha R,”Energy Aware Resource Utilization Technique for Workflow Scheduling in Cloud Computing Environment”, Turkish Journal of Computer and Mathematics Education, vol.12, 2021

[12] Yanqi Zhang, Weizhe Hua, Zhuangzhuang Zhou, G. Edward Suh, Christina Delimitrou,"Sinan: ML-Based and QoS-Aware Resource Management for Cloud Microservices"2021

[13] Mostafa HadadianNejadYousefi, AmirmasoudGhiassi, Boshra Sadat Hashemi, MaziarGoudarzi, “Workload Scheduling on heterogeneous Mobile Edge Cloudin 5G networks to Minimize SLA Violation”, 2020

[14] Shreshth Tuli, Sukhpal Singh Gill, Minxian Xu, Peter Garraghan, Rami Bahsoon, SchahramDustdar, RizosSakellariou,Omer Rana, Rajkumar Buyya, Giuliano Casale, Nicholas R. Jennings,”HUNTER: AI based Holistic Resource Management for Sustainable Cloud Computing’, Distributed, Parallel, and Cluster Computing, 2021

[15] R. Pinciroli, A. Ali, F. Yan and E. Smirni, "CEDULE+: Resource Management for Burstable Cloud Instances Using Predictive Analytics," in IEEE Transactions on Network and Service Management, vol. 18, no. 1, pp. 945-957, March 2021, doi: 10.1109/TNSM.2020.3039942.

[16] Sohan Kumar Pande, Sanjaya Kumar Panda, Satyabrata Das, Kshira Sagar Sahoo, Ashish Kr. Luhach, N. Z. Jhanjhi, RoobaeaAlroobaea and Sivakumar Sivanesan,” A Resource Management Algorithm for Virtual MachineMigration in Vehicular Cloud Computing”, Tech Science Press, vol.67, 2021

[17] M.Buvana, Dr.K.Loheswaran, et al., ”Improved Resource Management and Utilization Based on a Fog-Cloud Computing System with IoT Incorporated with Classifier Systems”,Microprocessors and micro systems,2021.

[18] FarzinZaker, Marin Litoiu, and Mark Shtern,, “Formally Verified Scalable Look Ahead Planning For Cloud ResourceManagement”, 2022

[19] Wang, B., Wang, C., Song, Y. et al. A survey and taxonomy on workload scheduling and resource provisioning in hybrid clouds. Cluster Comput 23, 2809–2834 (2020).https://doi.org/10.1007/s10586-020-03048-8

[20] Saif, M.A.N., Niranjan, S.K. & Al-ariki, H.D.E. Efficient autonomic and elastic resource management techniques in cloud environment: taxonomy and analysis. Wireless Netw 27, 2829–2866 (2021). https://doi.org/10.1007/s11276-021-02614-1.

[21] IrajNaruei and FarshidKeynia, "A new optimization method based on COOT bird natural life model ", Expert Systems With Applications, vol. 183, 2021

[22] M.Khishe and M.R.Mosavi, "Chimp optimization algorithm", Expert Systems with Applications, vol. 149, 2020.

[23] Alena Paulikova, Zdenka Gyurák Babel'ová and Monika Ubárová, "Analysis of the Impact of Human–Cobot Collaborative Manufacturing Implementation on the Occupational Health and Safety and the Quality Requirements", International Journal of Environmental Research and Public Health, 2021.

[24] Mahendra Bhatu Gawali and Subhash K. Shinde, "Task scheduling and resource allocation in cloud computing using a heuristic approach", Journal of Cloud Computing: Advances, Systems and Applications, 2018.

[25] Datasets https://research.google/tools/datasets/google-cluster-workload-traces-2019/

[26] Mohamed Abd Elaziz , Laith Abualigah , Rehab Ali Ibrahim and Ibrahim Attiya,"IoT Workflow Scheduling Using Intelligent Arithmetic Optimization Algorithm in Fog Computing", Computational Intelligence and Neuroscience, 2021.

[27] Heba M. Eldesokey, Saied M. Abd El-atty, Walid El-Shafai, Mohammed Amoon, Fathi E. Abd El-Samie, "Hybrid swarm optimization algorithm based on task scheduling in a cloud environment", communication systems, vol.34, 2021.

[28] Syed Karimunnisa,Vijaya Sri ,"Cloud Computing: Review on recent research progress and issues", International Journal of Advanced Trends in Computer Science and Engineering, 2019, 8(2), pp. 216–223.

[29] Pradeep K, Sharma, and Jishnu , "Review of Task Scheduling based on different parameters in Cloud Environment", Micro Processors and Microsystems,2023.