DOI: http://dx.doi.org/10.26483/ijarcs.v13i3.6834

ISSN No. 0976-5697

Volume 13, No. 3, May-June 2022

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

MULTIVARIATE PIECEWISE REGRESSIVE AFRICAN BUFFALO OPTIMIZATION-BASED RESOURCE-AWARE TASK SCHEDULING IN CLOUD

Tamilsenthil S
Ph. D Research Scholar (Part Time),
Department of Computer Science, Government Arts College (Autonomous), Salem-7 and Assistant Professor,
Department of Computer Science, Padmavani Arts and Science College for women, Salem-11. India

Dr.Kangaiammal A
Assistant Professor, Department of Computer Applications,
Government Arts College (Autonomous), Salem-7.
indurath2007@gmail.com

Abstract: Distributed cloud computing handles a large number of tasks and provides many dynamic virtualized resources that aim to share as a service through the internet. While handling a large volume of tasks, task execution times, throughput, and makespan are the most significant metrics in practical scenarios. So, the scheduling task is essential to achieve accuracy and correctness on task completion. A novel technique called Multivariate Piecewise Regressive African Buffalo Optimization-based Resource Aware Task Scheduling (MPRABO-RATS) is introduced for improving the task scheduling efficiency and minimizing time consumption. First, the cloud user dynamically generates numerous heterogeneous tasks in the cloud environments. After receiving the tasks, the task scheduler in the cloud server finds the resource-optimized virtual machine using the Multivariate Piecewise Regressive African Buffalo Optimization technique. The proposed optimization technique uses the Multivariate Piecewise Regression function for analyzing the different resources availablity such as CPU Time, Memory, Bandwidth, and Energy before the task scheduling. Initially, the population of the virtual machine is defined. After that, the fitness is measured using Multivariate Piecewise Regression. Based on the fitness estimation, the resource-efficient virtual machine is determined. Finally, the task scheduler assigns the tasks to the resource-optimized virtual machine with higher efficiency. Experimental evaluation is carried out in the CloudSim simulator on the factors such as task scheduling Efficiency, Throughput, Makespan, and Memory Consumption with respect to a number of tasks. The observed results indicate that the MPRABO-RATS technique offers an efficient solution in terms of achieving higher task scheduling Efficiency, Throughput, and Minimizing the Makespan as well as Memory Consumption than the conventional scheduling techniques.

***Keywords*:** Cloud Computing, Resource-Aware Task Scheduling, Multivariate Piecewise Regressive African Buffalo Optimization

1. INTRODUCTION

Cloud computing is the rising technology in computer science where the different services are provided through the internet on-demand. Cloud computing is used for managing the huge number of tasks. In the cloud, task scheduling is a vital task. It is employed to handle the resources. In addition, the system permanence is improved. Some task scheduling methods are developed to enhance the load balancing performance and user's service quality with lesser task completion time.

An Enhanced Sunflower Optimization (ESFO) algorithm was designed in [1] for enhancing the performance of task scheduling with lesser makespan. However, the higher efficiency of task scheduling was not achieved. Task Schedule using Multi-Objective Grey Wolf Optimizer (TSMGWO) was introduced in [2] to determine near-optimal task scheduling solutions while handling a large number of tasks. However, it failed to focus on enhancing the task scheduling performance with lesser memory usage for peak loads of tasks.

An efficient approach using the Map-Reduce and Genetic Algorithm based Whale Optimization Algorithm GA-WOA framework was developed in [3] for scheduling tasks in an optimal manner for efficient scheduling of multiple tasks. The designed task scheduling failed to minimize the time taken for processing a specified task.

A Resource Allocation Security with Efficient Task Scheduling using Hybrid Machine Learning (RATS-HM) technique was designed in [4] to minimize the make-span time and enhance the throughput. However, the large amount of data was not handled with a real cloud environment to enhance the effectiveness of the proposed model. An Optimized Task Scheduling technique was designed in [5] for dynamic virtual machine allocation. The designed technique failed to focus on developing an optimal workflow scheduling approach for virtual machine selection and placement in the cloud.

Multi-objective Task Scheduling Optimization based on the Artificial Bee Colony Algorithm (ABC) was designed in [6] for reducing makespan and cost and to improve the throughput and average resource utilization. Task scheduling in a multi-cloud environment was challenging work using a Multi-objective task scheduling optimization algorithm. A modified Symbiotic Organism's Search algorithm (G-SOS) was designed in [7] to decrease the task execution time and response time for obtaining the optimal solution. The energy-aware task scheduling was not performed.

A fuzzy Self-Defense Algorithm was designed in [8] for multi-objective task scheduling optimization. However, the higher throughput of task scheduling optimization was not achieved. A Failure-Aware Task Scheduling approach was developed in [9] to minimize the runtime. But, the performance of makespan was not minimized.

Improving Data Locality using the Ant Colony Optimization (IDLACO) algorithm was designed in [10] to decrease the number of non-local executions and bandwidth consumption. The designed algorithm was not efficient to achieve higher efficiency.

1.1 Major Contributions of the Paper

In order to overcome the existing issues, a novel MPRABO-RATS is developed with the following contribution.

- In order to improve the efficiency of task scheduling in the cloud environment, the MPRABO-RATS scheduling technique is introduced by applying a Multivariate Piecewise Regressive African buffalo optimization technique.
- To increase the throughput and minimize the makespan, the MPRABO-RATS technique finds the resource-optimized virtual machine based on the Multivariate Piecewise Regression. The regression function analyzes the different resource availability by setting the threshold. The best optimal virtual machine is used to improve the scheduling time and throughput.
- An extensive and comparative simulation assessment is conducted to evaluate an in-depth analysis of the proposed MPRABO-RATS technique with existing methods via different metrics.

The remainder of the paper is arranged as follows. Section 2 focuses on related studies that investigate task scheduling and resource allocation. Section 3 describes the architecture of the proposed MPRABO-RATS. Section 4 focuses on evaluating the proposed MPRABO-RATS and the dataset description. Section 5 provides the results and discussion of the proposed MPRABO-RATS in comparison with the existing algorithms. Finally, concluding remarks are presented in Section "6".

2. RELATED WORKS

A Hybrid Optimization Algorithm was designed in [11] for task scheduling and virtual machine allocation. But the designed algorithm failed to enhance the efficiency of the proposed algorithm. An Adaptive Load-Balanced Task Scheduling (ALTS) method was introduced in [12] for scheduling the tasks to the optimal VMs to reduce the makespan and resource consumption. But, the energy-aware task scheduling was not designed in the real cloud environment. QoS-based Resource Allocation and Scheduling techniques were designed in [13] based on swarm-based ant colony optimization. However, resource management was not concentrated to enhance the scheduling efficiency.

A Performance and Energy Optimization Bi-Objective Algorithm were introduced in [14] for task scheduling. But the designed algorithm failed to evaluate the CPU-intensive, memory-intensive task scheduling in the cloud. A Distributed Grey Wolf Optimizer (DGWO) was developed in [15] for scheduling dependent tasks to VMs with minimum computation and data transmission costs. However, the designed algorithm failed to solve the scheduling problem for workflow applications using efficient optimization algorithms.

A Hybrid of Particle Swarm Optimization and Gray Wolf Optimization (PSO-GWO) algorithm was introduced in [16] for workflow scheduling. But, the total energy consumption, and response time was not considered for the evaluation purpose. An Improved Whale Optimization Algorithm was developed in [17] for minimizing the task scheduling time and large virtual machine load in cloud computing. But the designed algorithm failed to develop an efficient scheduling system suitable for various task workloads.

A Hybrid Spider Monkey Optimization (HSMO) algorithm was designed in [18] to optimize the makespan and the cost. However, it failed to select the VMs deployed in different physical regions and considered data transfer costs between different data centers. A new Hybrid Heuristic-based List Scheduling (HH-LiSch) algorithm was designed in [19] for solving task scheduling. But, it failed to apply a hybrid scheduling algorithm for power management of task scheduling in a cloud environment.

A Novel Chemical Reaction Partial Swarm Optimization algorithm was introduced in [20] for assigning multiple independent tasks on the available virtual machines. However, the designed optimization algorithm failed to consider the parameters like energy and power consumption to enhance the performance of task scheduling.

3. PROPOSED METHODOLOGY

In the Internet era, cloud computing is progressed as a well-organized distributed platform. But the major issue in a cloud platform is task scheduling. Allocating multiple tasks to the Virtual Machine is a challenging task in cloud computing. Many algorithms have been introduced to enhance the scheduling process in the cloud environment. But the existing algorithms have their drawbacks to achieve higher scheduling efficiency.

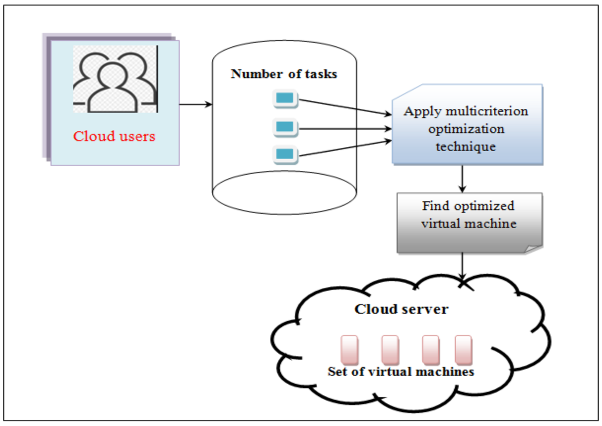

Figure 1: Block diagram of MPRABO-RATS Technique

This paper has proposed the Multivariate Piecewise Regressive African Buffalo Optimization-based Resource aware Task scheduling technique to allocate the task to the Virtual Machine with the help of a scheduler. The MPRABO-RATS technique reduces the overload of the virtual machine by improving the throughput and minimizing the makespan.

Figure 1 illustrates the architecture diagram of the proposed MPRABO-RATS technique to provide efficient task scheduling in the cloud. The cloud architecture comprises cloud users '$U = \{u_1, u_2, \dots, u_m\}$ who dynamically generates the multiple tasks $T = \{T_1, T_2, \dots, T_n\}$ arrived in the queue. The architecture also contains the cloud server that includes a set of virtual machines '$Vm = \{Vm_1, Vm_2, \dots, Vm_b\}$ for assigning the incoming tasks based on resource optimization. The proposed technique considers the major resources such as CPU, memory, bandwidth, and energy. These processes of the proposed MPRABO-RATS technique are explained in the following subsections.

3.1 Multivariate Piecewise Regressive African Buffalo Optimization-based Scheduling in Cloud

Task scheduling is a significant and most popular computing model in the cloud. Scheduling is the process of handling many tasks in the cloud computing environment. During the scheduling process, resources optimization plays a major impact in enhancing efficiency and minimizing the task execution time. Therefore, the proposed MPRABO-RATS technique is to improve the efficiency of task scheduling by applying a Multivariate Piecewise Regressive African buffalo optimization technique in this paper.

A Multivariate Regressed African Buffalo Optimization is a Metaheuristic Optimization Algorithm inspired by the movement of African Buffalos from one position to another across the vast African forests, and deserts in search of food. In contrast to optimization algorithm, proposed Multivariate Regressed African buffalo optimization is an easy to use, and vigorous. Also, it is used for efficient exploitation and exploration of the search space. Few learning parameters outcome are employed to ensure the quick convergence rate. The multivariate functions are handled by using proposed optimization technique. Multivariate optimization includes the several objective functions such as energy, bandwidth, CPU, and memory.

In this work, the solution to the task scheduling problem is symbolized as an array of tasks $T = \{T_1, T_2, \dots, T_n\}$ that are scheduled to virtual machines $Vm_1, Vm_2, \dots, Vm_b$ using Multivariate Regressed African buffalo optimization as shown in Figure 2. The proposed optimization technique helps for task scheduling based on resources allocation.

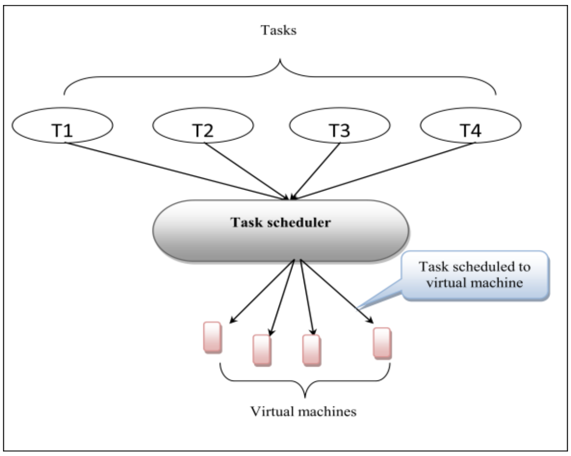

Figure 2: Diagram of Task Scheduling

Figure 2 illustrates the diagram of Task scheduling. First, the proposed MPRABO-RATS technique starts to initialize the populations of African buffalos in the search space. Here, the Buffalo is related to the virtual machines in the cloud server. Therefore, the initialization process of the proposed MPRABO-RATS technique is given in equation (1).

$$Vm = \{Vm_1, Vm_2, \dots, Vm_b\} \quad (1)$$

Where Vm denotes virtual machines. After the initialization of the virtual machine in search space, the optimal virtual machine is identified based on the CPU, memory, bandwidth, and energy.

First, CPU usage defines the amount of time consumed by the virtual machine to execute a certain task. The available resource of CPU is defined as in equation (2).

$$CPU_{AVL} = (CPU_T - CPU_{cd}) \quad (2)$$

Where CPU availability of the virtual machine 'CPU_{AVL}' is measured based on the difference between total CPU time 'CPU_T' and consumed time of Vm 'CPU_{cd}' to complete a certain task.

Second, the memory is another important task used for storage of the server in the cloud, which is calculated as in equation (3).

$$M_{avl} = (M_T - M_{ut}) \quad (3)$$

From (3), the memory availability of the virtual machine 'M_{avl}' is measured as the difference between the total memory of a virtual machine 'M_T 'and utilization of storage space of a virtual machine 'M_{ut}'.

Thirdly, bandwidth is defined as the volume of data transmitted at a given time. It is measured in bits per second and the bandwidth availability of the 'Vm' is defined as in equation (4).

$$B_{avl} = (B_T - B_{ut}) \quad (4)$$

From (4) available bandwidth of Vm 'B_{avl}' is defined as the difference between the total bandwidth 'B_T' and utilization of bandwidth 'B_{ut}'.

At last, the energy availability of the virtual machine is estimated based on the difference between the total energy of the virtual machine as in equation (5).

$$E_{AVL} = (E_T - E_C) \quad (5)$$

From (6), E_{AVL} indicates residual energy of the virtual machine, E_T indicates total energy, E_C denotes consumed energy.

The multivariate piecewise regression is applied to an optimization technique to analyze the estimated resources such as CPU, memory, bandwidth, and energy in the fitness estimation. Multivariate piecewise regression is a machine learning technique used to analyze the given estimated resource availability with the threshold value beyond or below which desired effects occur. The threshold value is used for decision making i.e. finding the resource optimized virtual machine. Therefore, the fitness of the virtual machine based on the piecewise regression is formulated as given in equation (6).

$$F = \begin{cases} (CPU_{AVL} > t_1) \text{ and } (M_{avl} > t_2) \text{ and } (B_{avl} > t_3) \text{ and } (E_{AVL} > t_4)\ ; Optimal\ Vm \\ Otherwise\ ; non-optimal\ Vm \end{cases}$$

Where, F indicates a fitness function, t_1 denotes a threshold for CPU availability 'CPU_{AVL}', t_2 denotes a threshold for memory availability 'M_{avl}', t_3 indicates a threshold for bandwidth availability, t_4 denotes a threshold for energy availability.

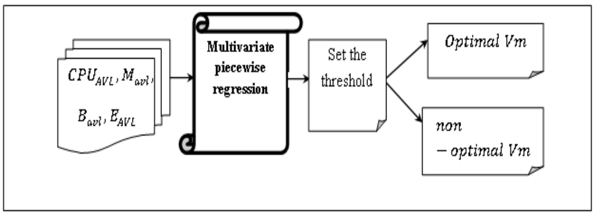

Figure 3: Multivariate Piecewise Regression-based Fitness Estimation

Figure 3 depicts the multivariate piecewise regression for fitness estimation to find the resource optimal virtual machine. Based on the fitness value, the two processes such as exploration and exploitation are performed. First, exploitation is the process of updating solutions from the existing ones based on the fitness function. Exploration is the process of discovering new solutions and searching for the best solution as in equation (7).

$$Q_b\,(t+1) = Q_b + k_1\,[Q_b(F) - ex_b] + k_2\,[Q_{ib} - ex_b] \qquad (7)$$

Where, $Q_b\,(t+1)$ indicates an updated solution using exploitation of the 'b' th buffalo, Q_b denotes a current solution of the b' th buffalo, ex_b designates an exploration of the $b\ th$ buffalos', k_1 and k_2 are the learning parameters and set the values from 0.1 to 0.6, $Q_b(F)$ specifies the best fitness of buffalo, Q_{ib} denotes an individual buffalo's best location. Followed by, the exploration of buffalos is updated as given in equation (8).

$$ex_b(t+1) = \frac{[ex_b + Q_b]}{\beta} \qquad (8)$$

Where, $ex_b(t+1)$ indicates an updated exploration of buffalos, β indicates a parameter value set as ±0.5. The above-said process gets repeated until the convergence is not met, otherwise stop the process. The flow chart of the proposed optimization process is given below,

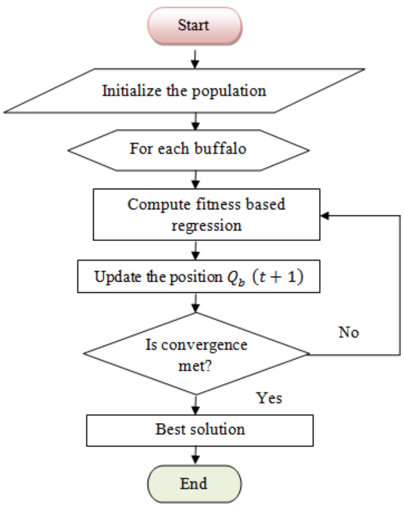

Figure 4: Flow Chart of Resource aware MPR-ABO Scheduling

Figure 4 illustrates the Resource aware Multivariate Piecewise Regressed African buffalo optimization scheduling for selecting the best optimum virtual machine to assign the incoming tasks in the cloud. The algorithmic process of the proposed Resource aware Multivariate Piecewise Regressed African buffalo optimization scheduling algorithm is described as follows,

// Algorithm 1 Multivariate Piecewise Regressive African Buffalo Optimization-based Resource Aware Task Scheduling
Input: users '$U = \{u_1, u_2, \dots, u_m\}$ tasks $T = \{T_1, T_2, \dots, T_n\}$', virtual machines '$VM = \{Vm_1, Vm_2, \dots, Vm_b\}$', task scheduler '$TS$ ', number of iterations '**T**'
Output: Enhance the scheduling efficiency **Begin** **1.** Collect the tasks '$T = \{T_1, T_2, \dots, T_n\}$ from users 'U' **2.** For each incoming task 'T' **3.** TS Initialize the population of $Vm_1, Vm_2, \dots, Vm_b$ **4.** **For each** virtual machine Vm_i **5.** Compute $CPU_{AVL}, M_{avl}, B_{avl}, E_{AVL}$ **6.** Measure the fitness 'F' **7.** **While** (T < Max_IT) **8.** **if** $\left(F(Vm_i) > F(Vm_j)\right)$ **then** **9.** Update buffalos' exploitation using (7) **10.** Update the location of buffalos using (8) **11.** **End if** **12.** **T= T+1** **13.** **end while** **14.** Obtain the resource optimized virtual machine **15.** TS assigns the tasks to resource optimized virtual machine **16.** **end for** **17.** **End**

Algorithm 1 given above describes the different processes of multivariate piecewise regressive African buffalo optimization-based resource-aware task scheduling to improve efficiency. The task scheduler initializes the population of the virtual machine in the search space. After that, the multivariate function is measured for each virtual machine in the population. Then the Piecewise regression function is applied to analyze the resource availability and find the best solution for the population. After the analysis, the optimal virtual machine is selected. If the fitness of one virtual machine is greater than the other, then the position of the 'buffalo gets updated. This process gets iterated until it reaches the maximum iteration. Finally, the task scheduler assigns the task to the resource-aware effective virtual machine. In this way, efficient scheduling of many tasks is performed with minimum time.

4. PERFORMANCE METRICS AND RESULTS ANALYSIS

In this section, the experimental evaluation of the MPRABO-RATS technique and two existing methods namely ESFO [1] and TSMGWO [2] are discussed with respect to various performance metrics such as task scheduling Efficiency, Throughput, Makespan, and Memory Consumption.

Task scheduling efficiency: It is defined as the ratio of the number of user-requested tasks that are correctly scheduled to the resource optimal virtual machines. The formula for calculating the task scheduling efficiency is given in equation (9).

$$TSE = \left[\frac{Number\ of\ tasks\ are\ correctly\ scheduled}{n}\right] * 100 \quad (9)$$

From (9), TSE indicates a task scheduling efficiency, 'n' represents a total number of user-requested tasks. The task scheduling efficiency is measured in percentage (%).

Throughput: It is defined as the number of cloud tasks executed per unit time. The formula for calculating the Throughput is calculated as given in equation (10).

$$T = \frac{Number\ of\ tasks\ executed}{t\ (seconds)} \quad (10)$$

Where 'T' indicates a throughput, t denotes a time. The throughput is measured in terms of tasks per second. A higher value of the throughput metric is desired for a better performing task scheduling method.

Makespan: It is defined as the time difference between starting and finishing a sequence of tasks allocated to the optimal virtual machine (equation (11).

$$Ms = T_{st} - T_{Fs} \quad (11)$$

Where, Ms denotes a makespan, T_{st} indicates a starting time of tasks allocated to an optimal virtual machine, T_{Fs} denotes finishing tasks allocated to the optimal virtual machine. It is measured in terms of milliseconds (ms).

Memory consumption: It is defined as the amount of storage space consumed by the algorithm to store numerous tasks. The overall memory consumption is formulated as given in equation (12).

$$MC = [n] * M\ [SST] \quad (12)$$

From (12), MC indicates memory consumption, n symbolizes the number of tasks and M symbolizes memory consumption, SST denotes scheduling the tasks. The memory consumption is measured in megabytes (MB)

Table I: Task Scheduling Efficiency

Number of User-Requested Tasks	Task Scheduling Efficiency (%)		
	MPRABO-RATS	**ESFO**	**TSMGWO**
5000	92	90	86
10000	93	90	87
15000	91	89	86
20000	92	88	86
25000	93	90	85
30000	91	88	84
35000	92	86	83
40000	92	88	84
45000	93	90	88
50000	92	89	85

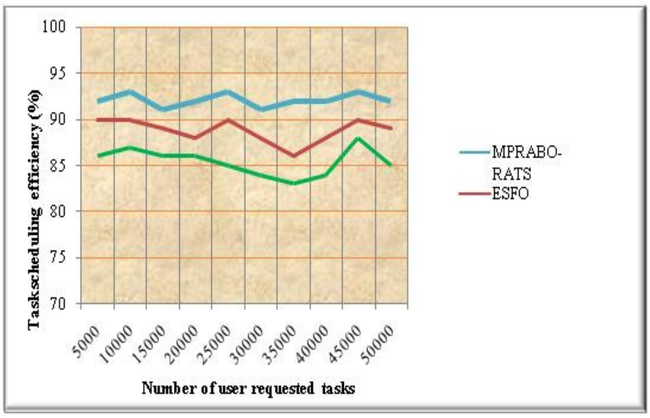

Figure 5: Comparative Analysis of Task Scheduling Efficiency

Figure 5 given above illustrates the graphical illustration of Task Scheduling Efficiency with respect to 50000 user-requested tasks. As shown in (Figure 5) graphical results, the Task Scheduling Efficiency is measured using three different methods namely the MPRABO-RATS technique and two existing methods namely ESFO [1] and TSMGWO [2]. The above graphical outcomes indicate that the task scheduling efficiency of the MPRABO-RATS technique increases the efficiency of scheduling multiple tasks to the cloud server. Let us consider 5000 tasks generated by the cloud user, 4625 user tasks are correctly scheduled to the resource-efficient virtual machine and the efficiency of the MPRABO-RATS technique is 92%. By applying ESFO [1] and TSMGWO [2], 4498 and 4321 tasks are correctly scheduled and the efficiency is 90% and 86% respectively. For each method, different results are observed with respect to a number of tasks. The overall results indicate that the Task scheduling efficiency of MPRABO-RATS is considerably improved by 4% when compared to ESFO [1] and 9% when compared to TSMGWO [2] respectively. This is due to the application of the MRP-ABO technique. The MPRABO-RATS technique uses the Multivariate Piecewise Regression function in the optimization process for analyzing the resources such as CPU time, memory, bandwidth, and energy. Based on the analysis, the optimal virtual machine is identified. The task scheduler assigns the incoming tasks to the resource-optimized virtual machine with higher efficiency.

Table II: Throughput

Number of User-Requested Tasks	**Throughput (Tasks/s)**		
	MPRABO-RATS	**ESFO**	**TSMGWO**
5000	565	410	370
10000	780	652	595
15000	875	765	690
20000	990	865	782
25000	1150	980	865
30000	1240	1150	980
35000	1398	1280	1050
40000	1580	1410	1280
45000	1655	1510	1405
50000	1780	1620	1520

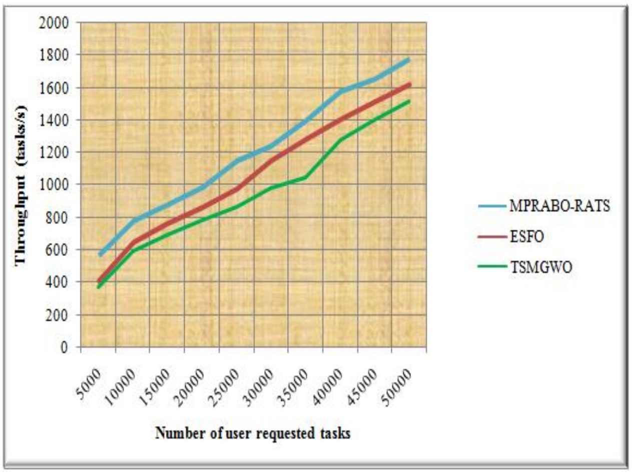

Figure 6: Comparative Analysis of Throughput

Figure 6 indicates the performance analysis of throughput versus a number of tasks between 5000 and 50000. In order to conduct the analysis of our results in terms of throughput, the performance of the MPRABO-RATS technique is compared to existing ESFO [1] and TSMGWO [2] respectively. Based on our analysis, the performance of throughput gets increased using the MPRABO-RATS technique than the existing methods. For each method, different results are observed. The final results of throughput results are compared to existing methods. The comparison results indicate that the proposed MPRABO-RATS technique is improved by 15% and 29% when compared to the existing [1] [2] respectively. The reason for this improvement was owing to the selection of optimal virtual machines using resource estimation. The task scheduler allocates the incoming tasks to the resource-optimized virtual machine. The optimal virtual machine executes multiple tasks resulting in improves throughput.

Table III: Makespan

Number of User-Requested Tasks	Makespan (ms)		
	MPRABO-RATS	ESFO	TSMGWO
5000	36	42	48
10000	43	50	55
15000	53	58	64
20000	62	70	82
25000	74	81	93
30000	83	88	98
35000	91	97	106
40000	105	112	120
45000	110	116	135
50000	122	128	140

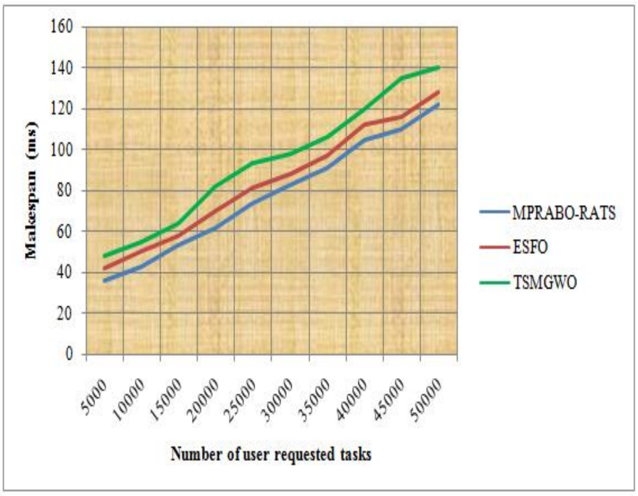

Figure 7: Comparative Analysis of Makespan

Figure 7 illustrates the performance results of makespan with respect to 5000 to 50000 user requests using three methods MPRABO-RATS technique is compared to existing ESFO [1] and TSMGWO [2]. The result of 10 different runs proves that the performance of makespan is considerably minimized than the conventional scheduling technique. Let us consider the number of user requests was 5000, MPRABO-RATS technique consumes 36ms whereas the ESFO [1] and TSMGWO [2] consume 42ms and 48ms respectively. Therefore, the average makespan time using the proposed MPRABO-RATS technique is lower as compared to other existing methods. In addition, by increasing the number of user requests, the makespan also gets increased. The overall ten comparison results indicate that the performance of makespan is comparatively minimized by 8% and 18% than the existing [1] [2] respectively. This is because finding a resource-efficient virtual machine also reduces the scheduling time of a large number of tasks.

Table IV: Memory Consumption

Number of User-Requested Tasks	Memory Consumption (MB)		
	MPRABO-RATS	ESFO	TSMGWO
5000	37	43	45
10000	45	48	51
15000	48	54	57
20000	54	58	62
25000	60	65	68
30000	66	69	72
35000	70	74	77
40000	76	80	84
45000	81	84	90
50000	85	88	93

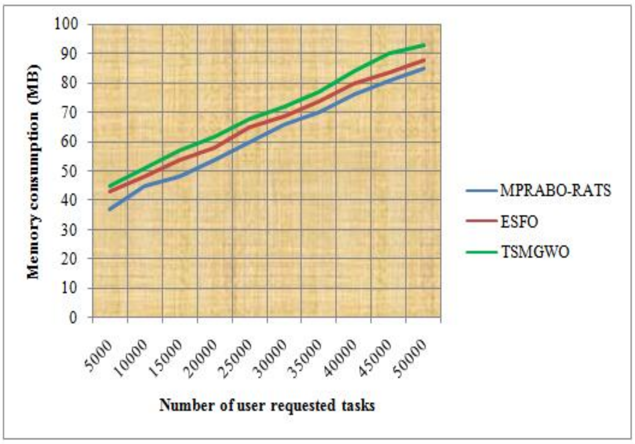

Figure 8: Comparative Analysis of Memory Consumption

Table IV and Figure 8 represent the performance results of memory consumption for scheduling multiple tasks by means of three different techniques MPRABO-RATS technique, ESFO [1], and TSMGWO [2]. The performance of memory consumption gets increased for all the methods by increasing the number of user-requested tasks. Among the three different scheduling techniques, the MPRABO-RATS technique provides superior performance than the other two conventional methods. However, the experiment is conducted with 5000 tasks in the first iteration. The proposed MPRABO-RATS technique utilizes $37MB$ for scheduling the multiple tasks, $43MB$ using [1] and $45MB$ using [2] respectively. The memory consumed for optimal resource allocation using the MPRABO-RATS technique was found to be comparatively smaller than [1] and [2]. Hence, the reason behind the efficient optimization-based task scheduling algorithm is observed. By applying this algorithm, the optimal virtual machine is correctly identified with minimum time to allocate the user requested task. Hence, the storage capacity of the MPRABO-RATS technique minimizes the memory consumption by 7% compared to [1] and 12% compared to [2] respectively.

5. CONCLUSION

Scheduling multiple user-submitted tasks to the virtual machines is considered the most challenging task in cloud computing. The main aim of scheduling is used to achieve different objectives like maximizing Throughput, minimizing makespan, and maximizing resource efficiency, among virtual machines in a cloud data center. Therefore, a novel MPRABO-RATS technique is developed for achieving the above-said objectives in cloud computing. The proposed scheduling technique provides the input as user-requested tasks to the task scheduler. After processing, a Multivariate Piecewise Regressive African buffalo optimization is designed for resource-aware task scheduling in cloud-based on multiple resources such as CPU time, memory, bandwidth, and energy. A series of experiments are conducted to test the performance of the MPRABO-RATS technique with different performance metrics with higher efficiency, throughput, and minimum makespan, as well as memory consumption when compared with the state-of-the-art methods.

REFERENCES

[1] Hojjat Emami, "Cloud task scheduling using enhanced sunflower optimization algorithm", ICT Express, Elsevier, Vol.8, No. 1, 2022, pp. 97-100. https://doi.org/10.1016/j.icte.2021.08.001

[2] Deafallah Alsadie, "TSMGWO: Optimizing Task Schedule Using Multi-Objectives Grey Wolf Optimizer for Cloud Data Centers", IEEE Access, Vol. 9, 2021, pp. 37707 – 37725. DOI: 10.1109/ACCESS.2021.3063723

[3] M.S. Sanaj, P.M. Joe Prathap, "An efficient approach to the map-reduce framework and genetic algorithm based whale optimization algorithm for task scheduling in cloud computing environment", Materials Today: Proceedings, Elsevier, Vol. 37, 2021, pp. 3199-3208. https://doi.org/10.1016/j.matpr.2020.09.064

[4] Prasanta Kumar Bal, Sudhir Kumar Mohapatra, Tapan Kumar Das, Kathiravan Srinivasan and Yuh-Chung Hu, "A Joint Resource Allocation, Security with Efficient Task Scheduling in Cloud Computing Using Hybrid Machine Learning Techniques", Vol. 22, Sensors, 2022, pp. 1-16. https://doi.org/10.3390/s22031242

[5] Deafallah Alsadie, "A Metaheuristic Framework for Dynamic Virtual Machine Allocation With Optimized Task Scheduling in Cloud Data Centers", IEEE Access, Vol. 9, 2021, pp. 74218-74233. DOI: 10.1109/ACCESS.2021.3077901

[6] Boonhatai Kruekaew, Warangkhana Kimpan, "Multi-Objective Task Scheduling Optimization for Load Balancing in Cloud Computing Environment Using Hybrid Artificial Bee Colony Algorithm With Reinforcement Learning", IEEE Access, Vol. 10, 2022, pp. 17803 – 17818. DOI: 10.1109/ACCESS.2022.3149955

[7] Ajoze Abdulraheem Zubair, Shukor Abd Razak, Md. Asri Ngadi, Arafat Al-Dhaqm,Wael M. S. Yafooz, Abdel-Hamid M. Emara, Aldosary Saad and Hussain Al-Aqrabi, "A Cloud Computing-Based Modified Symbiotic Organisms Search Algorithm (AI) for Optimal Task Scheduling", Vol. 22, Sensors, 2022, pp. 1-21, 1674. https://doi.org/10.3390/s22041674

[8] Xueying Guo, "Multi-objective task scheduling optimization in cloud computing based on fuzzy self-defense algorithm", Alexandria Engineering Journal, Elsevier, Vol. 60, Issue 6, December 2021, pp. 5603-5609. https://doi.org/10.1016/j.aej.2021.04.051

[9] Yanal Alahmad, Tariq Daradkeh, Anjali Agarwal, "Proactive Failure-Aware Task Scheduling Framework for Cloud Computing", IEEE Access, 2021, Vol. 9, pp. 106152-106168. DOI: 10.1109/ACCESS.2021.3101147

[10] Gurwinder Singh, Anil Sharma, Rathinaraja Jeyaraj, and Anand Paul, "Handling Non-Local Executions to Improve MapReduce Performance Using Ant Colony Optimization", IEEE Access, Vol. 9, 2021, pp. 96176-96188. DOI: 10.1109/ACCESS.2021.3091675

[11] G. Sreenivasulu and Ilango Paramasivam, "Hybrid optimization algorithm for task scheduling and virtual machine allocation in cloud computing", Evolutionary Intelligence, Springer, Vol. 14, 2021, pp. 1015–1022. https://doi.org/10.1007/s12065-020-00517-2

[12] Aroosa Mubeen, Muhammad Ibrahim, Nargis Bibi, Mohammed Baz and Habib Hamam and Omar Cheikhrouhou, "lts: An Adaptive Load Balanced Task Scheduling Approach for Cloud Computing", Processes, Vol. 9, 2021, pp. 1-15. https://doi.org/10.3390/pr9091514

[13] Harvinder Singh, Anshu Bhasin & Parag Ravikant Kaveri, "QRAS: efficient resource allocation for task scheduling in cloud computing", SN Applied Sciences, Springer, Vol. 3, 2021, pp. 1-7. https://doi.org/10.1007/s42452-021-04489-5

[14] Huned Materwala, and Leila Ismail, "Performance and energy-aware bi-objective tasks scheduling for cloud data centers", Procedia Computer Science, Elsevier, Vol. 197,

2022, pp. 238-246. https://doi.org/10.1016/j.procs.2021.12.137

[15] Bilal H. Abed-alguni, Noor Aldeen Alawad, "Distributed Grey Wolf Optimizer for scheduling of workflow applications in cloud environments", Applied Soft Computing Journal, Elsevier, Vol. 102, 2021, pp. 1-15. https://doi.org/10.1016/j.asoc.2021.107113

[16] Neeraj Arora and Rohitash K. Banyal, "Workflow scheduling using particle swarm optimization and gray wolf optimization algorithm in cloud computing", Concurrency and Computation: Practice and Experience, Wiley, 2021, Vol. 33, Issue 16, pp. 1-16. https://doi.org/10.1002/cpe.6281

[17] LiWei Jia, Kun Li, and Xiaoming Shi, "Cloud Computing Task Scheduling Model Based on Improved Whale Optimization Algorithm", Wireless Communications and Mobile Computing, Hindawi, Vol. 2021, December 2021, pp. 1-13. https://doi.org/10.1155/2021/4888154

[18] Naela Rizvi, Ramesh Dharavath, Damodar Reddy Edla, "Cost and makespan aware workflow scheduling in IaaS clouds using hybrid spider monkey optimization", Simulation Modelling Practice and Theory, Elsevier, Vol. 110, 2021, pp.1-24. https://doi.org/10.1016/j.simpat.2021.102328

[19] Mirsaeid Hosseini Shirvani, Reza Noorian Talouki, "A novel hybrid heuristic-based list scheduling algorithm in heterogeneous cloud computing environment for makespan optimization", Parallel Computing, Elsevier, Vol. 108, 2021, pp. 1-13. https://doi.org/10.1016/j.parco.2021.102828

[20] Kalka Dubey, S.C. Sharma, "A novel multi-objective CR-PSO task scheduling algorithm with deadline", Sustainable Computing: Informatics and Systems, Elsevier, Vol. 32, 2021, pp. 1-20. https://doi.org/10.1016/j.suscom.2021.100605.