Bivariate Correlative Oppositional Based Artificial Fish Swarm Resource Optimized Task Scheduling In Cloud

K.M.Ajitha,ResearchScholar and
Dr.N.Chenthalir Indra,Assistant Professor, S. T Hindu College,Nagercoil.
Manonmanium Sundaranar University, Tirunelveli, India

Cloud computing is an Internet-based approach provisioning of various computing services, to the users. In cloud, task scheduling is a significant process to allocate the workload for the different servers. Different evolutionary algorithms have been designed to solve the task scheduling issues in the cloud. But, the makespan and resource utilization performance were not improved during task scheduling by using population-based algorithms. In order to improve the task scheduling efficiency with minimum makespan and resource utilization, Bivariate Correlative Oppositional based Multiobjective Artificial Fish Swarm Resource Optimized Task Scheduling (BCO-MAFSROTS) technique is introduced. The main objective of the BCO-MAFSROTS technique is to reduce the workload across the cloud server by distributing the number of user-requested tasks to the optimal virtual machines. Initially, the number of user tasks are taken from the database. Then, incoming user tasks is given to the cloud sever. In the cloud server, bivariate correlation-based tasks prioritization method is performed for prioritized task as a high priority and low priority. Based on the prioritization, user tasks are distributed to the optimal virtual machines with the help of optimization algorithm. In the proposed BCO-MAFSROTS technique, oppositional based multiobjective artificial fish swarm optimization algorithm is utilized to perform task scheduling according to identifying optimal virtual machines in the cloud. The fitness is determined for each virtual machines based on multiple objective functions such as CPU time, bandwidth, memory and energy. From the fitness function estimation cloud server finds the optimal virtual machines for processing user-request task. Next, the experimental evaluation is carried out on factors such as task scheduling efficiency, false-positive rate, makespan and memory consumption with respect to a number of user tasks. The results discussion proves that the presented BCO-MAFSROTS technique improves the task scheduling efficiency and minimizes false-positive rate, makespan as well as memory consumption as compared to state-of-the-art methods.

Keywords: Cloud computing, task scheduling, Bivariate Correlation, virtual machine, Oppositional based Artificial Fish Swarm Optimization, multiple objective functions

1. INTRODUCTION

Cloud computing provides on-demand and cost-effective services to users over the internet. As thousands of users submit their tasks to the cloud server, the task scheduling method plays a vital role in cloud computing environments.Task scheduling is the process of assigning a group of tasks to the virtual machines (VMs) in the cloud server. The task is defined as a users queries send to the cloud server, and these queries are processed within the required time period. Therefore, the major challenging of task scheduling is to perform resource optimization for providing the quality of services.

A Fuzzy system and Modified Particle Swarm Optimization (FMPSO) technique was developed in (Mansouri, Zade, and Javidi, 2019) to improve the scheduling efficiency.The designed method failed to minimize the fault occurring in the task scheduling. A Moth Search Algorithm using Differential Evolution (MSDE) was introduced in (Elaziz, Xiong, Jayasena, and Li, 2019) scheduling the tasks. But it failed to consider the multi-objective optimization of the task scheduling

Author's address: K.M.Ajitha,ResearchScholar, Reg No:19123152282020, S.T. Hindu College, Nagercoil,Affiliated by Manonmaniam Sundaranar University, Tirunelveli-627012, TamilNadu, India;Dr.N.Chenthalir Indra, Research Supervisor and Assistant Professor, S. T Hindu College,Nagercoil, Affiliated by Manonmanium Sundaranar University, Tirunelveli-627012, TamilNadu, India.

model in the cloud environment. In order to overcome these existing issues, the proposed BCO-MAFSROTS technique is designed in this paper. At first, bivariate correlation coefficient value is measured for user tasks and to assign priority of the tasks. Then, multi objective optimization algorithm is used to identify the optimal virtual machines among different virtual machines for performing high priority tasks based on the fitness function evaluation.

Multiobjective hybrid algorithms called genetic algorithms and the bacterial foraging (BF) algorithms were developed in (Srichandan, Kumar, and Bibhudatta, 2018) for task scheduling with lesser makespan. The designed hybrid algorithms failed to speed up the convergence rate. The heuristic algorithm based task scheduling was introduced in (Gawali and Shinde, 2018) for effectively allocating the resources with higher utility. But the algorithm failed to focus on more efficient scheduling algorithms for minimizing the scheduling time.

An iterated heuristic framework was introduced in (Zhu, Li, Ruiz, and Xu, 2018) to schedule the job event by job collection and event scheduling process. However, resource provisioning for the scenario with fuzzy processing times and deadlines was not considered.A whale optimization algorithm (WOA) was designed in (Sreenu and Sreelatha, 2017) for scheduling the tasks using a multi-objective model. Though makespan was reduced, the memory consumption was not minimized.

A hybrid particle swarm optimization and hill-climbing algorithm were introduced in (Dordaie and Navimipour, 2018) to optimize the task scheduling makespan. But, the designed algorithm failed to consider more load balancing resources.An improved PSO (IPSO) algorithm was introduced in (Saleh, Nashaat, Saber, and Harb, 2018) for allocating the more tasks with minimum makespan.But the designed algorithm failed to consider other objective functions such as cost or energy.

Rapid Local Convolution Optimization (RLCO) technique was introduced in [(Liu, Liu, Hu, Zou, and Cheng,) for performing the task scheduling with lesser energy cost. The performance of makespan remained unaddressed. A particle swarm optimization algorithm with ant colony optimization was introduced in (Chen and Long, 2017) for more reliable and optimal task scheduling. The designed algorithms failed to perform the task scheduling with multiple tasks.

1.1 Proposal Contributions

The major issues are identified from the existing surveys are overcome by introducing the novel BCO-MAFSROTS technique. The overall contribution of the proposed work is summarized as follows,

—To improve the task scheduling efficiency in the cloud, the BCO-MAFSROTS technique is introduced. The oppositional based multi-objective artificial fish swarm algorithm considers multiple resources such as CPU time, bandwidth, energy,and memory for selecting the optimum virtual machine through the fitness evaluation. The cloud manager allocates the incoming tasks to the resource-efficient virtual machine.

—To minimize the false positive rate, a multi-objective artificial fish swarm algorithm uses the gradient ascent function in the fitness evaluation for identifying the resource optimized virtual machine to distribute the tasks. Besides, the behaviors of all the artificial fish are also described based on the fitness to update the current best solution for task scheduling among the population.

—To minimize the makespan, the BCO-MAFSROTS technique performs priority-based task scheduling which sort the user tasks based on the priority. Priority is calculated using bivariate correlation with the different parameters. Based on the priority assignments, the tasks are scheduled to virtual machines with minimum time.

1.2 Paper Outline

The rest of this paper is organized into five different sections. Section 2 elaborates on the issues and challenges of task scheduling in the literature survey. Section 3 describes the BCO-

MAFSROTS technique for task scheduling based on multiple objective functions. In section 4, experimental evaluation is carried out with a dataset and the performance results of various metrics are discussed in section 5. Finally, section 6 provides the conclusion of the work.

2. RELATED WORKS

An Energy-Aware Task Scheduling algorithm was introduced on cloud Virtual Machines (Ismail and Materwala, 2018) that allocates multiple tasks to the VM with minimum energy consumption. But the designed task scheduling algorithm failed to consider a different resource of VM such as bandwidth, memory and so on.

A pair-based task scheduling algorithm was introduced in (Panda, Nanda, and Bhoi,) with minimum time. But the designed algorithm was not efficient to process a large number of tasks. A modified particle swarm optimization (M-PSO) algorithm based task scheduling was introduced in (Zhou, Chang, Hu, Yu, and Li, 2018) to handle the problem of local optimum and slow convergence rate. But the algorithm failed to lessen the incorrect task scheduling in the cloud environment. A multi-objective Artificial Bee Colony Algorithm (TA-ABC) was introduced in (Jena, 2017) for scheduling the user tasks. Though the algorithm uses multiple objective functions, the scheduling efficiency was not improved.

A chaotic symbiotic organism search (CMSOS) algorithm was designed in (Abdullahi, Ngadi, Dishing, Ahmad, et al., 2019) to solve multi-objective task scheduling problems. The proposed algorithm failed to handle the scheduling process in the very large workload instances.A Harmony-Inspired Genetic Algorithm (HIGA) was introduced in (Sharma and Garg, 2019) for energy-efficient task scheduling and also minimizes the number of required resources. Though the designed algorithm reduces the makespan, the scheduling efficiency was not improved.

Laxity and ant colony system algorithm (LBP-ACS) was designed in (Xu, Hao, Zhang, and Sun, 2019)for scheduling the tasks based on the priority. The algorithm failed to minimize the makespan of task scheduling in the cloud.A modify Heterogeneous Earliest Finish Time (HEFT) algorithm was introduced in (Dubey, Kumar, and Sharma, 2018) that allocates the workload among the processor. Though the algorithm reduces the makespan time of applications, the performance efficiency was not improved.

In (Kumar and Venkatesan, 2019), a Hybrid Genetic-Particle Swarm Optimization (HGPSO) algorithm was introduced to evaluate the suitable resources for scheduling the user tasks. But, the false-positive rate was not reduced using HGPSO. An imperialist competitive algorithm and firefly algorithm was introduced in (Kashikolaei, Hosseinabadi, Saemi, Shareh, Sangaiah, and Bian, 2019) for scheduling the user tasks. But the algorithm did not solve the multiple objective problems of task scheduling in a cloud environment.

The major problems identified from the above-said literature are addressed by introducing a novel technique called BCO-MAFSROTS technique. The detailed processes of the BCO-MAFSROTS technique are explained in the following section.

3. METHODOLOGY

Cloud computing is arising technology in distributed computing which helps to provides the various service model as per the user demand along with their requirements. Cloud computing worked as the basis of the client-server architecture. The cloud users i.e. clients send their queries i.e. tasks to the various servers, and these queries also completed within the required time period. In order to provide the user accessing services, cloud users submitted their tasks to the cloud server for On-demand services provision with higher scalability and resource availability. As the data volume from all fields increases exponentially, the size of the tasks also rapidly increased. Therefore, an efficient task scheduling algorithm is required to process such kind of user tasks with minimum time. Based on this motivation, the BCO-MAFSROTS technique is introduced. On the contrary to the existing Artificial Fish Swarm optimization, the proposed technique solves the multiple objective functions and also uses the opposition based learning concept for efficient scheduling of the user tasks. The architecture of the BCO-MAFSROTS technique is shown in figure1 Figure 1 shows the architecture of the proposed BCO-MAFSROTS technique to

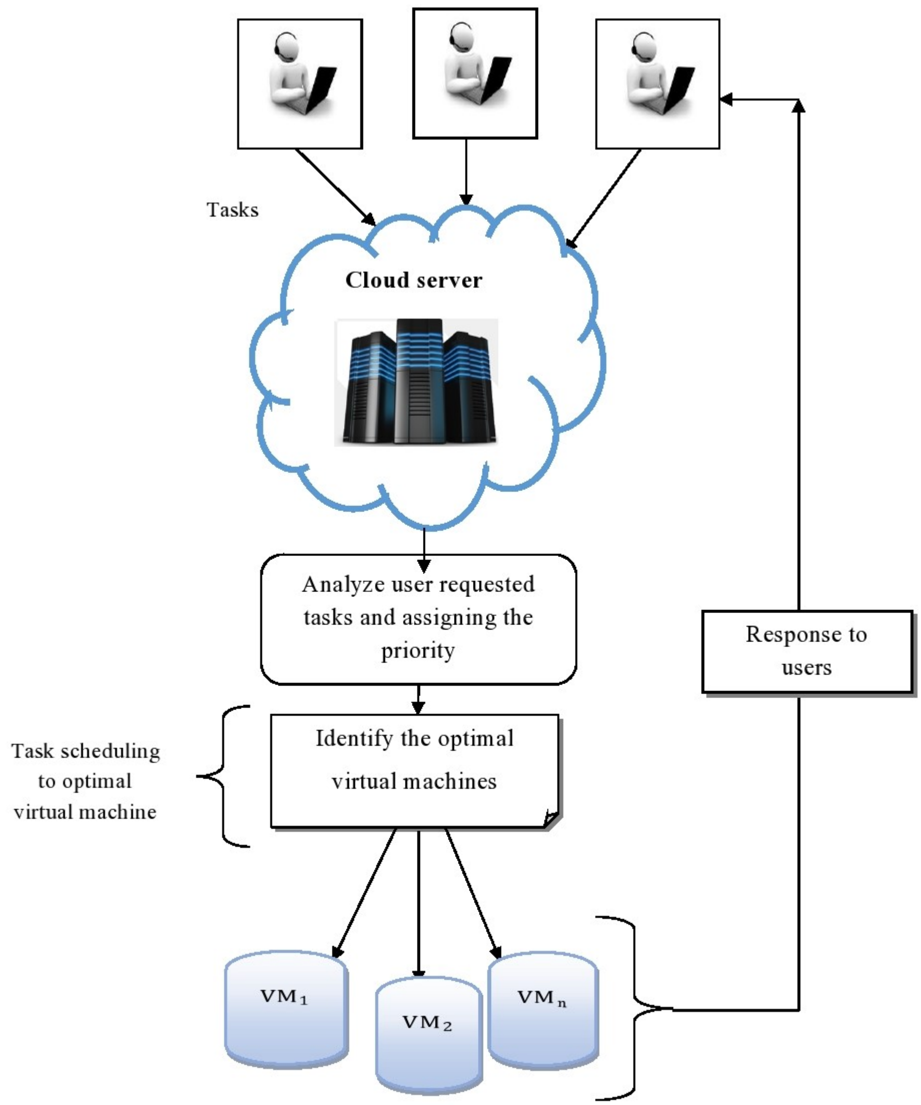

Figure 1. Architecture of BCO-MAFSROTS technique

schedules the multiple tasks into virtual machines in the cloud environment with minimum time. The proposed BCO-MAFSROTS technique improves task scheduling efficiency with minimum makespan. In the cloud, the multiple users send their requests (i.e. tasks) to a cloud server. After receiving the request from the cloud, the manager in the cloud server analyzes the user-requested tasks and identifies the priority of the tasks. The priority of the tasks is identified through the bivariate correlation with the different parameters such as task size, task arrival time and predicted task completion time. After that, the high priority tasks are scheduled first than the low priority tasks. Followed by, the cloud manager identifies the resource optimized virtual machine in the cloud server for scheduling the high priority tasks. The process of the BCO-MAFSROTS technique is described in the following sections.

3.1 Bivariate correlation-based tasks prioritization

In the cloud, a number of users generate different tasks having various sizes, different deadlines (i.e. ending time), arrived time, sending time. Based on tasks and their time constraints, the proposed BCO-MAFSROTS technique analyzes the user tasks based on their priority assignments. Let us consider the number of incoming tasks t_1,t_2,t_3,… t_n(i.e. user request) sent to the cloud server (C_s). In the cloud server, the Cloud Manager (C_M) analyzes the incoming tasks and assigns to virtual machines v_{m_1},v_{m_2},v_{m_3}… v_{m_n}. Based on the above system model, the task scheduling is performed in the following sections. Figure 2 illustrates the bivariate correlation-

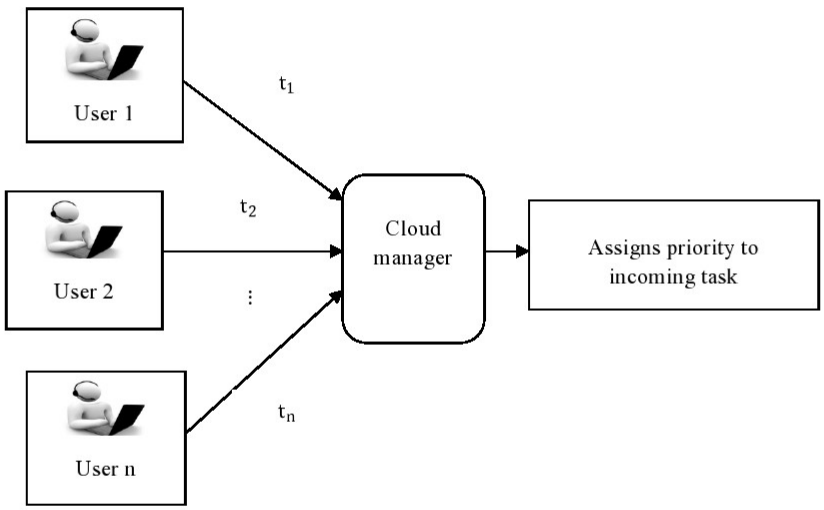

Figure 2.Bivariate correlation based priority assignments

based priority assignments. For each incoming task, the request parameters such as task size, request arrival time and the predicted task completion time are measured.Task size is the length of the tasks. Request arrival time is the time of user request arrived at the cloud server. The predicted task completion time of the virtual machine is measured as follows,

$$t_{p_c} = t_e - t_s \tag{1}$$

Where, t_{p_c} denotes a predicted task completion time, t_e represents a ending time to complete the task and t_s represents an starting time to process the particular task. Based on these three parameters, the bivariate correlation between the user tasks is measured as follows,

$$\beta = \frac{\Sigma t_i t_j - \frac{\Sigma t_i * \Sigma t_j}{n}}{\sqrt{(\Sigma t_i^2 - \frac{(\Sigma t_i)^2}{n})} * \sqrt{(\Sigma t_j^2 - \frac{(\Sigma t_j)^2}{n})}} \tag{2}$$

Where β denotes a bivariate correlation coefficient,n represents a number of tasks,t_i,t_j denotes a user tasks, $\Sigma t_i t_j$ refers to the sum of the product of paired score, Σt_i is the sum of t_i score, Σt_j is the sum of t_j score, Σt_i^2 is the sum of the squared score of t_i and Σt_j^2 is the sum of the squared score of t_j. The Bivariate correlation coefficient provides the value between +1 and +1. The Bivariate correlation coefficient provides +1 which means the positive correlation. The correlation coefficient provides -1 represents the negative correlation. The negative correlation means that the tasks t_2 is assigned as high priority and tasks t_1 is assigned as low priority. The positive correlation represents the tasks t_1 is assigned as high priority and tasks 't_2 is assigned as low priority. Based on the analysis, the tasks are prioritized as a high priority and low priority for performing the task scheduling in the cloud.

3.2 Oppositional based Multiobjective Artificial Fish Swarm optimization-based Task Scheduling

After prioritizing, task scheduling is carried out using Oppositional based Multiobjective Artificial Fish Swarm Optimization. This helps to distribute the user-requested tasks to resource-efficient virtual machines. Cloud server consists of a collection of a virtual machine which includes both computational and storage facilities. The main objective of cloud computing is to offer the various services to remote and geographically distributed resources.The oppositional based multiobjective artificial fish swarm optimization is a swarm intelligence that works depends on the population and stochastic search. Swarm Intelligence is the collective behavior of living animals like birds, fishes, ants and so on. The behavior of the animal is a movement and seeking its food source. On the contrary to the existing optimization algorithm, the proposed optimization algorithm is the best approach that having higher convergence speed, flexibility, error tolerance,and higher accuracy. Artificial fish swarm is a metaheuristic algorithm depends on the behaviors of fish such as prey, swarm, and follows. Here the Artificial fish is related to the number of virtual machines in the cloud server and food source is the resources i.e. CPU time, bandwidth, memory,and energy. Based on these resources, an optimal virtual machine (i.e. artificial fish) is selected among the population for assigning the high priority tasks.In addition, the proposed optimization technique uses opposition based learning to avoid the local optimum by selecting better individuals for the next generation. The main aim of the opposition based learning concept is to consider the opposite estimates or actions attempt to improve the coverage of the solution space and it helps to improve the accuracy with lesser computation time.

The above optimization algorithm is related to task scheduling in the cloud. Initialize the population of n artificial fish swarms (i.e. virtual machine) V=$v_{m_1}, v_{m_2} v_{m_3}, \ldots v_{m_n}$ randomly in the search space. By applying the opposition based learning concept, the opposite artificial fish swarms population is also generated in order to obtain a better solution.The opposite candidate solution provides the global optimum solution than a random. Therefore, the opposition based artificial fish swarms population generation is mathematically expressed as follows,

$$V' = s_i + t_i - V \tag{3}$$

Where, V' denotes an opposite solution of current population V, s_i and t_i represents the minimum and maximum value of the dimensions in the current population V. The current population at the same time opposite of the current population is generated in search space. After the initialization, the fitness is calculated for each fish swarm in the current population as well as the opposite population of the swarm.The fitness is calculated based on past data of the fish swarm using multiple objective functions such as CPU time, bandwidth, memory and energy. Initially, the CPU time of the virtual machine is calculated as follows,

$$CPU_t = cpu_T - cpu_c \tag{4}$$

Where, CPU_t denotes a remaining CPU time capacity of virtual machine, cpu_T represents a total time and cpu_c represents a consumed time of the virtual machine to process the particular task. The bandwidth availability of a virtual machine is calculated based on the difference between the total bandwidth and consumed bandwidth as follows,

$$b_{wa} = (b_w(t) - b_w(c)) \tag{5}$$

Where,$b_w a$ denotes a bandwidth availability,b_w (t)is the total bandwidth, $b_w(c)$ denotes a consumed bandwidth. The memory is the other major resources of the virtual machine in task scheduling processes. It is defined as an amount of storage space required to process the high priority user tasks. The memory availability is calculated as follows,

$$m_{vm}(a) = m_t - m_{us} \tag{6}$$

Where, $m_(vm$ (u)) represents the memory availability of the virtual machine, m_t represents a total memory space of virtual machine and m_us denotes an utilized memory space of virtual machine. The residual energy of the virtual machine is measured using the following the mathematical formula,

$$E_{vm}(r) = E_t - E_c \tag{7}$$

Where,$E_(vm(c))$ denotes a residual energy of the virtual machine, E_t denotes a total energy, E_c is the consumed energy. Based on the above said resources, the cloud manager finds the optimal virtual machine through the fitness measure for scheduling the number of virtual machines with different processing capacities. In the fitness measure, the optimization technique uses the gradient ascent function to find the fittest for selecting the optimal one among the population. The gradient ascent function is used to find the maximum resource availability of the virtual machine for processing the specific task.

$$F(x) = \arg max(t_{pc}, b_w, m_{vm(u)}), E_{vm(c)} \tag{8}$$

Where, F(x) denotes a fitness, arg max denotes a gradient ascent function to find the virtual machine with maximum resource availability. After that, the current population and opposite swarm populations are merged and sorting the artificial fishes according to their fitness value. Finally, select n best artificial fishes from the collection are selected for further processing. Based on the fitness value, three behaviors of the artificial fish positions such as search or prey, swarm and follow are carried out to identify the global best solution as follows,

3.2.1 *Search or prey behavior of fish.* Prey is a fundamental behavior of the artificial fish that helps to find the food. In general, the fish identifies the awareness of food in water by vision or sense. Let us consider, the current position of the fish is p_i and the new position of fish is $p_i(t+1)$. If the fitness of one fish is greater than the other i.e. $F(x_i) < F(x_j)$, then search or prey behavior of fish is executed and then the position of the fish is updated as follows,

$$p_i(t+1) = p_i(t) + r * \delta * \Big(\frac{(p_j - p_i)}{\| p_j - p_i \|}\Big) \tag{9}$$

Where, p_i (t+1) denotes a updated position of fish , p_i is the current position, r denotes a random number varied from zero to one $(0 < r < 1)$,δ denotes a step of the fish moving which is a random positive number, $\| p_j - p_i \|$ is the visual distance between the position of the j th fish and the position of the i th fish.

3.2.2 *swarm behavior of fish.* In this behavior, the fishes are grouped in the moving process for avoiding the risks. Let us consider the current position of the fish is p_i , p_c is the center position of the several fish. The swarm behavior of the artificial fish is executed when the $F(x_c) < F(x_i) \&\& \Big(\frac{(n_b}{n} < \beta\Big)$ Where,F (x_c) is the fitness of artificial fish at the center position,n_b denotes a number of companions within the current neighborhood, n denotes a total number of fish, β denotes a crowd factor values from 0 to 1.It means that the center of fishes, there is more food (i.e. higher fitness value).Followed by, the position of the artificial fish is updated as follows,

$$p_i(t+1) = p_i(t) + r * \delta * \Big(\frac{(p_c - p_i)}{\| p_c - p_i \|}\Big) \tag{10}$$

Where, p_i (t+1) denotes a updated position of fish , p_i is the current position, r denotes a random number varied from zero to one $(0 < r < 1)$,δ denotes a step of the fish moving which is a random positive number, $\| p_c - p_i \|$is the visual distance between the position of the j th fish and the central position of the fish in its current neighborhood.

3.2.3 *Follow behavior of fish.* In the moving process of the fish swarm, when a single fish or number of fishes finds their food, neighborhood trails and reaches the food in a very fast manner.Let $p_i$be the current position of the fish, and it uses the companion p_j in the neighborhood.If $(x_j) > F(x_i)\&\&\left(\frac{n_b}{n} < \beta\right)$, then the follow behavior is executed which means that the companion x_j state has higher food concentration (i.e. higher fitness value). For artificial following fishes, the position-updating is formulated as,

$$p_i(t+1) = p_i(t) + r * \delta * \left(\frac{(p_{max} - p_i)}{\| p_{max} - p_i \|}\right) \tag{11}$$

Where, p_i (t+1) denotes a updated position of fish,p_i is the current position,p_{max} denotes a position having the best fitness function value inside the visual,r denotes a random number varied from zero to one (0¡r¡1),δ denotes a step of the fish moving which is a random positive number, $\| p_{max} - p_i \|$ is the visual distance between the position of the i th fish and the central position of the fish having the best fitness function. Replace the old fish into a new optimal one based on the fitness. This process is repeated until the maximum iteration is reached. Finally, the cloud manger assigns the incoming tasks into the resource efficient virtual machine. The algorithmic process of optimization technique is described as follows, As shown in algorithm 1, step by step process of task scheduling is described using a multiobjective optimization technique. For each incoming user task, the cloud manager analyzes the tasks with the different parameters and identifies the high and low priority tasks. After that, the cloud manager starts to perform the task scheduling with the high priority tasks using a multi-objective optimization technique. The number of fish populations and their positions, as well as opposite swarm populations, is initialized in the search space. The proposed optimization technique calculates the fitness to all the virtual machine (fishes) in the current population and the opposite swarm populations with multiple resources such as bandwidth, memory, energy, CPU time of the virtual machine. Followed by, virtual machines in the two populations are merged and sorted based on their fitness value. Then the current best n virtual machines are selected for finding the global optimum solution. Then the behavior of the fishes is estimated and updates the positions. Finally, the global best resource-efficient virtual machine is selected for processing the user-requested tasks. The entire process gets iterated until the maximum iteration is reached. In the above algorithms, t represents the iteration count of the algorithm. Finally, the cloud manager allocates the high priority tasks to the resource optimal virtual machine. As a result, the proposed optimization technique solves the multiple objective problems resulting in minimizes the error and improves the scheduling efficiency.

The above description shows that the proposed BCO-MAFSROTS technique scheduling the incoming tasks to the optimal virtual machines in the server with higher efficiency. The above-said algorithmic processes are implemented to show the performance of the proposed scheduling algorithm compared to existing methods.

4. EXPERIMENTAL EVALUATION AND PARAMETER SETTINGS

Experimental evaluations of proposed BCO-MAFSROTS technique and existing methods namely FMPSO [1] and MSDE [2] are implemented using Java language with the CloudSim network simulator. The Personal Cloud Datasets is taken from http://cloudspaces.eu/results/datasets .Cloud simulator contains user interface structures, VM services, cloud services, cloud resources and network applications. In cloud simulator, number of user tasks given to the optimal virtual machines for performing task scheduling. The optimal virtual machines are selected based on the maximum resource availability. In cloud simulator, at first high priority tasks are processed by optimal virtual machines then the low priority task are processed.This dataset comprises the 17 attributes (i.e. columns) and 66245 instances (i.e., user tasks). The objective of this dataset is to perform load and transfer tests. The 17 attributes are row id, account id, file size (i.e. task size), operation_time_start, operation_time_end, time zone, operation_id, operation type, bandwidth

Input: Number of cloud user tasks $t_1, t_2, t_3, .t_n$, virtual machines $(v_{m_1}, v_{m_2}, v_{m_3}, \ldots v_{m_n})$
Output: Improve the task scheduling efficiency
Begin
For each incoming task t_i
Find the request parameters task size, request arrival time and predicted task completion time Measure the bivariate correlation β
Find higher and low priority tasks
end for
\\ **Apply optimization technique** Initialize the current population of $V = v_{m_1}, v_{m_2}, v_{m_3}, \ldots v_{m_n}$
Initialize opposite populations V'
for each higher priority t_i
for each v_{m_i} in V and V'
Calculate the fitness F(x)
Merge the populations V and V'
Sort the virtual machines
Select current best n virtual machines
While (t $<$ $maximum\_iteration$)
if $(F(x_i) < F(x_j))$ then
perform preying behaviors and updates the position p_i (t+1)
else if(F$(x_c) < F(x_i)$&&$(\frac{n_b}{n} < \beta)$) then
perform swarming behaviors and updates the position p_i (t+1)
else if $(F(x_j) > F(x_i)$&&$(\frac{n_b}{n} < \beta)$)then
perform the following behaviors and updates the position p_i (t+1)
end if
Replace the old v_m into current best
end for
end for
t= t+1
end while
Obtain best optimal solution
Schedule high priority t_i to optimal v_{m_i}
End

Algorithm 1: Bivariate Correlative Oppositional based Multiobjective Artificial Fish Swarm Resource Optimized Task Scheduling

trace, node_ip, node_name, quoto_start, quoto_end, quoto_total (storage capacity), capped, failed and failure info. Among 17 attributes, two attributes namely time zone and capped are not used. The remaining attributes are considered for scheduling the tasks to multiple virtual machines with the resources in the cloud.Performance analysis of the proposed technique is compared with existing results with the following parameters listed below,

—Task schedulingefficiency

—False-positive rate

—Makespan

—Memory consumption

5. PERFORMANCE EVALUATION UNDER VARIOUS PARAMETERS

In this section, the performance of the proposedBCO-MAFSROTS technique and existing methods namely FMPSO (Mansouri et al., 2019) and MSDE (Elaziz et al., 2019) are discussed with different metrics such as task scheduling efficiency, false-positive rate, makespan, and memory consumption. The performances of three methods are discussed either table or graphical representation. For each subsection, the mathematical calculation is provided to identify the improvement of the proposed technique compared to existing methods.

5.1 Impact of task scheduling efficiency

Task Scheduling Efficiency is defined as a number of incoming tasks are correctly scheduled to the resource optimized virtual machine from the total number of tasks. The task scheduling efficiency is mathematically calculated as follows,

$$T_{SE} = \left(\frac{Numberoft_{correctlysheduled}}{n}\right) * 100 \tag{12}$$

Where, T_{SE} denotes a task scheduling efficiency, Number of $t_{correctlysheduled}$ represents the number of tasks correctly scheduled, n represents the number of tasks. The task scheduling efficiency is measured in terms of percentage (For conducting the experimental evaluation, 100 to 1000 user-requested tasks are considered. Let us consider the 100 tasks for calculating the scheduling efficiency. The BCO-MAFSROTS technique correctly scheduled 95 tasks and their percentage is 95% and the scheduling efficiency of FMPSO (Mansouri et al., 2019) and MSDE (Elaziz et al., 2019) are 90% and 88% respectively. From the observed results, it is significant that the BCO-MAFSROTS technique achieves higher task scheduling efficiency than conventional optimization techniques. Totally, ten various results of scheduling efficiency are described in below Table 1. Table 1 describes the various results of task scheduling efficiency versus a number of user-requested tasks varied from 100 to 1000. The above-reported results evidently prove that the task scheduling efficiency of the three methods BCO-MAFSROTS technique is higher than the FMPSO [1] and MSDE [2]. This is because of BCO-MAFSROTS technique initially identifies the task priority level with the help of bivariate correlation. After that, the high priority tasks are assigned first than the other tasks. The multi-objective optimization technique is applied in the BCO-MAFSROTS technique for finding the optimal virtual machine for processing the high priority tasks. The cloud manager accurately finds the virtual machine by calculating the fitness using a gradient ascent function. The fitness function is calculated based on the CPU time, bandwidth availability, residual energy,and memory availability. The virtual machine which has maximum availability is selected as optimal and the cloud manager assigns the tasks to that virtual machine. As a result of scheduling, all the tasks are scheduled for the resource-efficient virtual machine. The efficiency of the BCO-MAFSROTS technique is compared to the existing technique. The average value of comparative results improves the scheduling efficiency by 8% and 11% as compared to FMPSO (Mansouri et al., 2019) and (Elaziz et al., 2019) [2] respectively.

Number of tasks	**Task scheduling efficiency (%)**		
	BCO-MAFSROTS	**FMPSO**	**MSDE**
100	95	90	88
200	94	89	86
300	93	87	85
400	92	85	84
500	91	84	82
600	90	83	80
700	89	82	79
800	88	81	78
900	87	80	77
1000	86	79	76

Table I: Task scheduling efficiency versus number of tasks

5.2 Impact of the false-positive rate

The false-positive rate is defined as a number of incoming tasks that are incorrectly scheduled to the resource optimized virtual machine from the total number of tasks. The false-positive rate is mathematically calculated as follows,

$$FPR = \left(\frac{Numberof t_{incorrectlysheduled}}{n} * 100\right) \tag{13}$$

Where FPR denotes a false positive rate, Number of $t_{incorrectlysheduled}$ represents the number of tasks incorrectly scheduled, n represents the number of tasks. The false positive rate is measured in terms of percentage (%). Figure 3 given above illustrates the comparative analysis of the

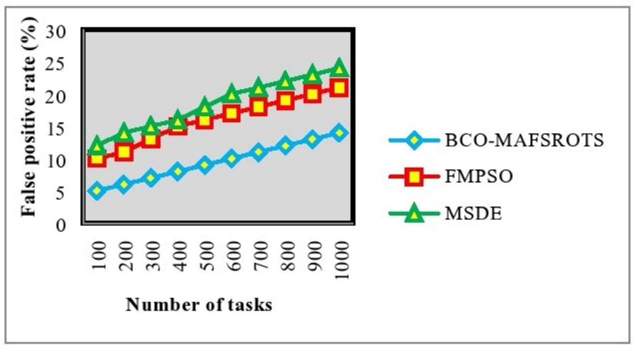

Figure 3.Comparative analysis of the false-positive rate

false-positive rate with respect to a number of incoming tasks. As shown in the graph, the x axis refers to the number of tasks and the yaxis refers to the false positive rate of the task scheduling in the cloud.Lesser the rate of false positive, more efficient the method is said to be. The false positive rate of the task scheduling efficiency is minimized by applying the BCO-MAFSROTS technique as compared to the existing techniques. This is because of the fact that the BCO-MAFSROTS technique applied oppositional based learning concept in the multi-objective optimization technique. The oppositional based learning concept creates the opposite population of the swarm for the initial population. Then the fitness is computed for each virtual machine in both the populations. Finally, the current population and the opposite populations are combined and sorting in a descending order according to their fitness. This process of optimization technique effectively finds the global optimum solution and avoids the local optimum in search space. Then the cloud manger accurately finds the resource efficient virtual machine for scheduling the tasks. This in turn minimizes the incorrect task scheduling in the cloud.The result of the false positive rate is significantly minimized by 42% as compared to FMPSO (Mansouri et al., 2019) and 50% compared to MSDE (Elaziz et al., 2019).

5.3 Impact of Makespan

Makespan is defined as an amount of time taken to schedule the tasks to resource optimized virtual machines. The overall Makespan is mathematically calculated as follows,

$$Makespan = n * time(scheduleonetask) \tag{14}$$

Where n represents the number of tasks and the Makespan is measured interms of milliseconds (ms). Table 2 clearly describes the experimental results of makespan using three different methods namely the BCO-MAFSROTStechnique, FMPSO (Mansouri et al., 2019) and MSDE (Elaziz et al., 2019). Totally ten various results are obtained as shown in the table. For each run, the

Number of tasks	Makespan (ms) (%)		
	BCO-MAFSROTS	FMPSO	MSDE
100	30	35	40
200	34	38	42
300	41	45	48
400	44	48	56
500	50	55	60
600	54	60	63
700	56	63	67
800	60	64	69
900	64	68	74
1000	66	70	75

Table II: Makespan versus the number of tasks

various numbers of input tasks are taken to calculate the makespan. The results clearly show that the makespan is found to be minimized using the BCO-MAFSROTS technique as compared to existing scheduling techniques. This is because the cloud manager performs a correlation between the incoming tasks with respect to the size, arrival time, predicted task completion time. Based on the analysis, the cloud manager prioritizes the incoming tasks. The task with higher priority is scheduled first than the low priority tasks. Then the cloud manager finds the optimal virtual machine in the cloud server for processing the higher priority tasks. The proposed optimization technique finds the global resource-efficient virtual machine to handle a large number of tasks. Hence, the makespan involved in task scheduling is found to be reduced by 9% using the BCO-MAFSROTS technique compared to FMPSO [1] and 17% compared to MSDE [2].

5.4 Impact of memory consumption

Memory consumption is defined as an amount of storage space utilized by the virtual machine to store the tasks. The memory consumption is mathematically computed as follows,

$$MC = n * space(storingonetask) \tag{15}$$

Where n represents the number of tasks and the memory consumption is measured in terms of megabytes (MB).Lower the memory consumption involved in scheduling the tasks, the method is said to be more efficient. The sample scenarios for calculating the memory consumption using the proposed BCO-MAFSROTS technique and two other existing methods are given below. The experimental results of the memory consumption of the three methods are plotted in Figure 4. As shown in the figure, the memory consumption of three scheduling techniques BCO-MAFSROTS technique, FMPSO [1] and MSDE [2] are represented in blue, red and green colors of line. The graphical results indicate that the BCO-MAFSROTS technique provides lesser memory consumption for storing user tasks. The proposed technique finds the resource-efficient virtual machine to store multiple user tasks. The optimal virtual machine consumes lesser storage space. This is clearly evident using mathematical calculations. Let us taken 100 tasks for conducting the experimental, the BCO-MAFSROTS technique consumes 27MB of memory for storing the 100 tasks whereas the existing scheduling techniques FMPSO (Mansouri et al., 2019) and MSDE (Elaziz et al., 2019) consumes 30MB and 35MB respectively. Followed by, the remaining nine runs are carried out and compare the results of proposed and existing methods. The comparison results show that the BCO-MAFSROTS technique considerably reduces the memory consumption by 8% and 15% than the existing FMPSO [1] and MSDE [2] respectively.

From the above discussion, it clearly proves that the BCO-MAFSROTS technique achieves better task scheduling efficiency with minimum makespan as well as memory consumption.

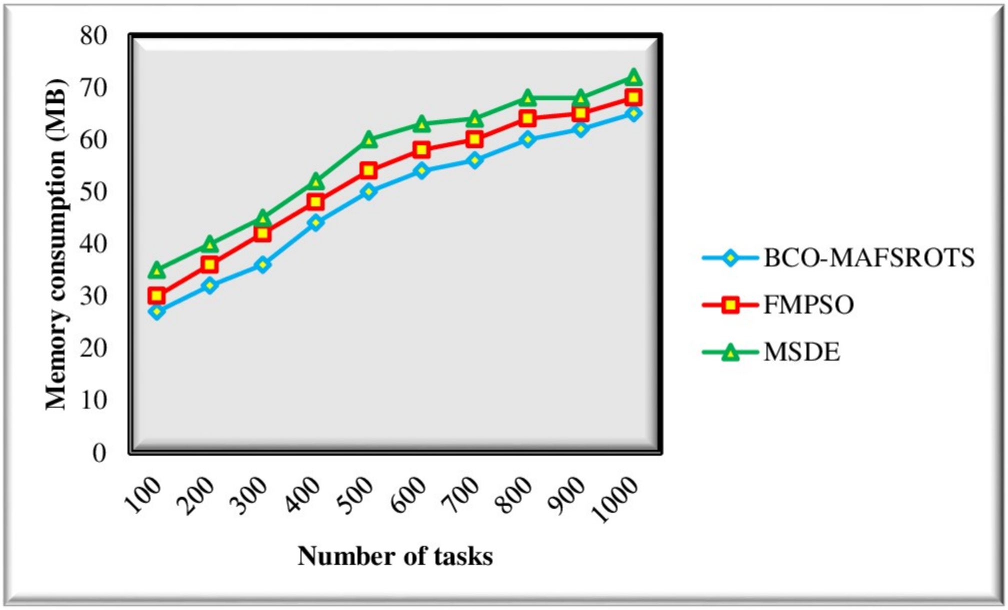

Figure 4.Comparative analysis of memory consumption

6. CONCLUSION

The BCO-MAFSROTS technique is introduced with the objective of improving task scheduling efficiency with lesser makespan in cloud computing. First, the incoming tasks from the cloud users are analyzed by the cloud manager with the different tasks scheduling parameters. Then the cloud manager identifies the higher and low priority tasks for performing the task scheduling. Next, with the higher priority tasks, an oppositional based multi-objective artificial fish swarm algorithm is applied to find the resource-efficient virtual machine through the fitness evaluation. Finally, a cloud manager schedules the incoming tasks to the selected virtual machine. Then the selected virtual machine processed the higher priority tasks and response to the cloud users with minimum time. To evaluate the performance, a comparative analysis has been undertaken among the proposed algorithm and the state-of-the-art scheduling algorithms. The experimental result shows that the BCO-MAFSROTS technique provides better performance for scheduling the number of tasks to a cloud server with an improvement of efficiency and minimization of makespan as compared to state-of-the-art works.In future, the proposed BCO-MAFSROTS technique further extend to analyze the service-level agreement (SLA)violations and Quality of Service (QoS) for improving performance of scheduling efficiency.

References

Abdullahi, M., Ngadi, M. A., Dishing, S. I., Ahmad, B. I., et al. 2019. An efficient symbiotic organisms search algorithm with chaotic optimization strategy for multi-objective task scheduling problems in cloud computing environment. *Journal of Network and Computer*

Applications 133, 60–74.

Chen, X. and Long, D. 2017. Task scheduling of cloud computing using integrated particle swarm algorithm and ant colony algorithm. *Cluster Computing*, 1–9.

Dordaie, N. and Navimipour, N. J. 2018. A hybrid particle swarm optimization and hill climbing algorithm for task scheduling in the cloud environments. *ICT Express 4,* 4, 199–202.

Dubey, K., Kumar, M., and Sharma, S. 2018. Modified heft algorithm for task scheduling in cloud environment. *Procedia Computer Science 125*, 725–732.

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

Gawali, M. B. and Shinde, S. K. 2018. Task scheduling and resource allocation in cloud computing using a heuristic approach. *Journal of Cloud Computing 7,* 1, 1–16.

Ismail, L. and Materwala, H. 2018. Eatsvm: Energy-aware task scheduling on cloud virtual machines. *Procedia Computer Science 135*, 248–258.

Jena, R. 2017. Task scheduling in cloud environment: A multi-objective abc framework. *Journal of Information and Optimization Sciences 38,* 1, 1–19.

Kashikolaei, S. M. G., Hosseinabadi, A. A. R., Saemi, B., Shareh, M. B., Sangaiah, A. K., and Bian, G.-B. 2019. An enhancement of task scheduling in cloud computing based on imperialist competitive algorithm and firefly algorithm. *The Journal of Supercomputing*, 1–28.

Kumar, A. S. and Venkatesan, M. 2019. Task scheduling in a cloud computing environment using hgpso algorithm. *Cluster Computing 22,* 1, 2179–2185.

Liu, X., Liu, P., Hu, L., Zou, C., and Cheng, Z. Energy-aware task scheduling with time constraint for heterogeneous cloud datacenters. *Concurrency and Computation: Practice and Experience*, 1–17.

Mansouri, N., Zade, B. M. H., and Javidi, M. M. 2019. Hybrid task scheduling strategy for cloud computing by modified particle swarm optimization and fuzzy theory. *Computers & Industrial Engineering 130*, 597–633.

Panda, S. K., Nanda, S. S., and Bhoi, S. K. A pair-based task scheduling algorithm for cloud computing environment. *Journal of King Saud University-Computer and Information Sciences*, 1–12.

Saleh, H., Nashaat, H., Saber, W., and Harb, H. M. 2018. Ipso task scheduling algorithm for large scale data in cloud computing environment. *IEEE Access 7*, 5412–5420.

Sharma, M. and Garg, R. 2019. Higa: Harmony-inspired genetic algorithm for rack-aware energy-efficient task scheduling in cloud data centers. *Engineering Science and Technology, an International Journal*, 1–14.

Sreenu, K. and Sreelatha, M. 2017. W-scheduler: whale optimization for task scheduling in cloud computing. *Cluster Computing*, 1–12.

Srichandan, S., Kumar, T. A., and Bibhudatta, S. 2018. Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm. *Future Computing and Informatics Journal 3,* 2, 210–230.

Xu, J., Hao, Z., Zhang, R., and Sun, X. 2019. A method based on the combination of laxity and ant colony system for cloud-fog task scheduling. *IEEE Access 7*, 116218–116226.

Zhou, Z., Chang, J., Hu, Z., Yu, J., and Li, F. 2018. A modified pso algorithm for task scheduling optimization in cloud computing. *Concurrency and Computation: Practice and Experience 30,* 24, 1–11.

Zhu, J., Li, X., Ruiz, R., and Xu, X. 2018. Scheduling stochastic multi-stage jobs to elastic hybrid cloud resources. *IEEE Transactions on Parallel and Distributed Systems 29,* 6, 1401–1415.

K.M.Ajitha is the Research Scholar in S.T Hindu College, Nagercoil,Kanyakumari (India) and nearly 7 years experience as an Assistant Professor in Computer Science Department.Paper Publication in conference is Exploration on task scheduling techniques in cloud environment, International Conference on Computational Sciences(ICCS 2019), Alagappa University.

Dr N Chenthalir Indra is an Assistant Professor in S.T Hindu College, Nagercoil,Kanyakumari (India). She has more than 15 years of teaching Experience.Her research interests are in Image processing,Data mining and Cloud Computing.Paper Publications in international conferences is Self Acquiring Image Knowledge Using Maximum Value Metric based Self Organizing Maps in IEEE International Advanced Computing Conference.Got Best Paper Award on, Similar Dissimilar Victor Measure Analysis to Improve Image Knowledge Discovery Capacity of SOM in International Conference on Information and Communication Technologies.