Turkish Journal of Computer and Mathematics Education Vol.12 No. 11 (2021), 3885- 3898
Research Article

A Particle Swarm and Ant Colony Optimization based Load Balancing and Virtual Machine Scheduling Algorithm for Cloud Computing Environment

Kumar Surjeet Chaudhury[a], Prof.(Dr.) Sabyasachi Pattnaik[b], Dr. Ashanta Ranjan Routray[c]

[a] P.G. Department of Information and Communication Technology, Fakir Mohan University Vyasa Vihar, Balasore.
[b] Department of Information and Communication Technology, Fakir Mohan University Vyasa Vihar, Balasore
[c] Department of Information and Communication Technology, Fakir Mohan University Vyasa Vihar, Balasore

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 10 May 2021

Abstract: In cloud computing balancing the load of the Virtual Machine (VM) is very much essential. Load balancing efficient utilization of resource and fairly balance the resource usage. In the real time scenario the request for the Virtual Machine (VM)s and tasks submission could be dynamic, whereas system creates the Virtual Machine (VM) according to the customer demand and map it to suitable Physical Machine (PM). These Virtual Machine (VM) could be created without knowing the detailed information about the task.Hence, the scheduling of these taskscould not be optimised using traditional task scheduling algorithms. In this paper a hybrid meta heuristic approach for scheduling these tasks is proposed. Two different optimization techniques for Virtual Machine (VM) scheduling has been used in this paper. We combine Particle Swarm optimization and Ant Colony Optimization approaches called (PSACO). The PSACO uses the historical information regarding the Virtual Machine (VM)s and task submission to predict The workload of new task submission and resource request in dynamic environment without extra information. The proposed approach also rejects the computing request which does not satisfied the current resource constraints. It reduces the computation time for scheduling. the experiment results shows that the proposed metaheuristic algorithm balance the load with the dynamic environment and outperformed the existing algorithms.

Keywords: Load Balancing, Virtual Machine, Cloud Computing, Physical Machine, Particle Swarm Optimization, Ant Colony Optimization.

1. Introduction

The cloud computing is not a recent technology, but the advancement of computing resources and the demand of computing resources, increases the requirement of cloud computing related services. By migrating the IT infrastructure to cloud many companies saves more than 50% of their annual infrastructure maintenance revenue. In the opposite side many tech giants flourish by using the cloud computing technology and providing the services. In recent era cloud resource requirement is in peak and these tech giants are also increasing their cloud computing infrastructure to satisfy the need. Cloud computing services are mainly divided into three categories: Infrastructure as a Service (IaaS),Platform as a Service (PaaS), and Software as a Service (SaaS). In the Infrastructure as a Service, the cloud service providers give the facility of hardware resources and it is the responsibility of the cloud service user to maintain the system software and application software along with data. Whereas in the Platform as a Service, the cloud service provider provides the hardware and essential system software Anne it is the responsibility of the cloud service user to maintain their data and application software. In the Software as a Service, the hardware and software will be maintained by the cloud service provider and the cloud user can only use the service without bothering about the hardware and the software. The cloud service could also be deployed mainly in three categories: Public, Private, and Hybrid. In the Public deployment any cloud service user can use the services can mainly resource pooling characteristics of the cloud computing is used [18]. Whereas in the Private cloud deployment the cloud services are dedicated to the cloud service user and it's provides more security to the data. Because it is using a dedicated hardware resource for the cloud service user, it is more expensive then public deployment. To balance the security and expenses bear by the cloud service user a third category of cloud deployment is used called Hybrid cloud deployment model. In this both Public and Private cloud deployment model's advantages are used and provide better service to the cloud computing users.

In the services such as Infrastructure as a Service, the Virtual Machine (VM) is a sign to the cloud service user based on the requirement and demand. Later it has been mapped to the Physical Machine (PM). Due to the limited hardware resources available with the cloud service provider it is essential to allocate the resources by considering many constraints,arbitrary and random location of Virtual Machine (VM) may result to low performance output. Therefore the proper allocation hardware resources and task scheduling may lead to balance the load of the Physical Machine (PM) an increase the performance of the cloud computing infrastructure [19]. Hence the Virtual Machine (VM) scheduling become and essential characteristics in cloud computing. The existing cloud computing solutions may not be suitable for fulfilling the dynamic request of

computing resources and also, they required different information related to the tasks. The existing system may not get the exact map of the workload load with the Virtual Machine (VM) Before the execution of the tasks. Also, it required to execute the Virtual Machine (VM) scheduling with the less amount of information. All these limitations may lead to inconsistent utilization of computing recourses and unbalanced Physical Machine (PM). Which may cause the business revenue of the cloud service provider. In this paper we proposed a meta heuristic algorithm called (PSACO), based on the particle swarm optimization and on colony optimization for Virtual Machine (VM) scheduling and balancing the load in cloud computing. The proposed algorithm can work in the dynamic environment where the request are generated randomly.So, in this dynamic environment we can evaluate the performance of the proposed algorithm in terms of different matrices.

The main objectives of this paper are:

1. To propose a hybrid meta-heuristic algorithm for load balancingand Virtual Machine (VM) scheduling in cloud computing.
2. To enhance the Virtual Machine (VM) scheduling time by reducing the dimension of the result and fine tune the ant colony optimization algorithm by using particle swarm optimization operator.
3. To propose an efficient algorithm, which takes historical and minimal information about the workload to fulfil the requirement of new workload requests.
4. To implement the proposed hybrid meta-heuristic algorithm with dynamic workload request and to compare the result with the existing algorithm under different matrices.

The rest of the paper is organized as follows;In Section 2 the related research has been discussed in literature survey. In Section 3 the proposed protocol is described with the help of algorithm and equations. The simulation results and analysis of the proposed algorithm is done in Section 4followed by Conclusion and References

2.Literature Survey

The scheduling in the cloud computing environment is completely different then the scheduling in uni-processor operating system. In the traditional uni-processor operating system the limited number of tasks are scheduled for a single processor whereas in the cloud computing environment there are ample amount of task request submitted to cloud as well as the configuration of age Virtual Machines (VM) are different and based on that the cost as well as the execution time is decided. It is quite difficult while scheduling to consider different parameters such as cost execution time completion time etcetera in the cloud computing environment. As the Virtual Machine (VM) configuration is based on the cloud computing user and the task submitted to it. The mapping of Virtual Machine (VM) to Physical Machine (PM) is same as the task are mapped based on their property to multi-core processor unit. These mapping problems could be optimized using different optimization algorithm which comes into the category of NP-Complete problem. Many natures inspired Virtual Machine (VM) scheduling and optimization algorithm has been proposed.

Kaur et al. [1] Proposed the virtual machine load balancing technique based on the hybrid approach. In this the combination of two metaheuristic algorithms are used and two different frameworks are proposed. The first approach is called HDD-PLBwhich is the combination of Predict Earliest Finish Time (PEFT) Heuristic and Ant Colony Optimization (ACO). The second approach is based on Hybrid Heterogeneous Earliest Finish Time (HEFT) heuristic and ACO. The main objective of this approach is to find the optimal performance of the cloud computing system based on the performance metrices such as makespan and cost.

Jena et al. [2] presented a novel technique for balancing the load in the virtual machine in dynamic environment. in this technique two optimization method are used to make a hybrid approach in which the modified particle swarm optimization and the modified Q learning technique are called QMPSO .It used to improve the performance of the physical machine by increasing the throughput of the virtual machine providing the balance among the virtual machine and reducing the waiting time of the task-request.

Muhammad Junaid et al. [4] classified the input request using support vector machine and based on the classification they provide the task request to the proposed approach which is the hybrid metaheuristic approach based on the Ant colony optimization with file type formatting. They claim that the proposed hybrid metaheuristic algorithm can balance the load in the cloud environment. They also compared the performance of the proposed approach by using the matrices such SLA violation, migration time, overhead time, throughput and QoS.

Srinivasa Rao et al. [5] provided very basic approach for load balancing by combining software and hardware approaches. The author claim to provide effective service with proper scheduling of tasks in the peak hours and maintaining the equilibrium in the cloud computing load. They also consider three parameter for load balancing such as makespan, waiting time and burst time.

Subhadarshini et al. [6] proposed the metaheuristic approach for resource scheduling in changeable environment. This hybrid approach is based on the particle some optimization which utilizes the behavior of the particles swarm to balance the loadof the physical machine. In this paper they claim that the proposed approach will minimize the task overhead also maximize the resource utilization.

Ahmad M. Manasrah et al. [7] used the hybrid approach to balance the load in the cloud environment where the resources are available in a heterogeneous manner. This algorithm is called hybrid GA-PSO which allocates the task to the resources in an efficient way. It is based on the genetic algorithm and the particle some optimization algorithm. authors claim that this algorithm reduces the Max pain and cost as well as balance the load of cloud computing environment.

Mala Yadav et al.[12] Proposed a hybrid metaheuristic algorithm. it is based on genetic algorithm and particle swarm optimization algorithm. the main focus of this paper is to find the approximate solution for balancing the load of the virtual machine. Authors claim that the result after experiments are the optimize result that they obtained.

Gang Li et al.[9] focused on the task scheduling problem with the system wide information management (SWIM) load imbalance. In this paper the classical on colony optimization algorithm is used for large scale network task scheduling with hardware performance. the pheromone produced by the urn this updated based on the load standard deviation function the proposed algorithm is called can't colony task scheduling and load balancing algorithm (ACTS-LB).

Shabnam et al. [10] proposed a hybrid algorithm for load balancing and virtual machine optimization using the bat algorithm. This swarm based bad algorithm has been modified and used for optimizing the virtual machine load and eventually lead to balancing the load of the physical machine.Youssef Fahim et al. [11] also proposed metaheuristic bat algorithm for assigning the tasks to the available virtual machine.

There are some AntColony Optimization based algorithms[8] [13] [14] exists in which some are based on the dynamic workload, multi-function objective or modified version of ACO. As well as some are based on Particle Swarm Optimization [15-17].

The proposed PSACO algorithm is based on two metaheuristic algorithm such as Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO). This hybrid approach will use the functionality of both the meta here stick approach for scheduling the virtual machine and balancing the load in cloud computing environment.

3.Proposed Algorithm

In this section the proposed hybrid meta-heuristic algorithm call (PSACO) is described with the problem definition the request model the system model and objective. The PSACO algorithm is also described for Virtual Machine (VM) scheduling and optimization with explanation of different operators.

3.1 Environment Model:

The load balancing and Virtual Machine (VM) scheduling in the cloud computing environment are the functions which map the Virtual Machine (VM) to the Physical Machine (PM) and balances the load of the Physical Machine (PM). If we consider PM as the Physical Machine (PM)and there are N number of Physical Machines (PMs) available then that can be represented as {PM1, PM2, PM3, …PMN}. In the Virtual Machine (VM) the resources are represented as the m-dimensional array, where every dimension represent a computing resource. Whereas in the Physical Machine (PM) the physical computing resources are available that could be CPU, Memory and Storage etc. The problem that we are looking to solve in this paper could be formulated as follows:

In the system model of the proposed problem, N number of Physical Machines (PMs) are available which are homogeneous in which one of the Physical Machine (PM) is assigned as central Physical Machine (PM). the work off the central Physical Machine (PM) is to process the request to create the Virtual Machine (VM). Every Physical Machine (PM) can create the Virtual Machine (VM) according to the request received by the central Physical Machine (PM) including the central Physical Machine (PM). For creating the Virtual Machine (VM) minimum memory should be there in the Physical Machine (PM).

The task-request model in the proposed work consist of the request for the Virtual Machine (VM). Generally in the Infrastructure as a Service (IaaS) the virtual machine request are based on some computing resources and it does not required any other detailed information about the task that to be run it's execution time or the workload detail. So let Virtual Machine (VM) request is a set of resources which is required by the cloud service user, which can be denoted as CRj = {cr1, cr2, cr3, ….crm}. Here the CR are the set of computing resources requested at jth time. All the request came to the system for Virtual Machine (VM)are independent to each other. The individual request cri = (Mr, Pc) admitted into the system where Mr is the memory required and Pc is the CPU Processor core required for Virtual Machine (VM). In this proposed paper the Virtual Machine (VM) request is fulfilled and the system creates the VMs according to the request of the memory and the processing capacity.

3.2 Calculation of the Balance Level of the Cloud

The main objective of the virtual machine (VM) scheduling along with balancing the load is to increase the Load balancing for the utilization of resource and allow maximum request possible. In this proposed model for

evaluating the cloud computing system's balance state the features like distribution of the resourceutilizations(Dru) and the resource utilisations maximum difference (MDru) are used. To evaluate the cloud computing systems state, the balance level (BL) is used as shown in Equationbelow.

$$BL = x_1 * D_{ru} + x_2 * (1 - MD_{ru}) \quad \text{.....(1)}$$

In the Equation 1. x1 , x2 denotes the weights applied to the Distribution of resource utilization as Dru and Maximum Deference of the resource utilization as MDru. So the Cloud computing balance level represented as BL and calculated as above.

$$MD_{ru} = \omega_1 * \frac{CU_{max} - CU_{min}}{Max_{CPU\_Utilization}} + \omega_2 * \frac{RM_{max} - RM_{min}}{Max_{Memory}} + \omega_3 * \frac{SU_{max} - SU_{min}}{Max_{Storage\_Utilization}} \quad \text{.....(2)}$$

In the Equation 2. ω1 , ω2 , ω3 are the weights applied to get the maximum difference on resource utilization for CPU utilization (CU), Residual Memory (RU) and Storage Utilization (SU). In combining all these resources the Maximum Difference on resource utilization MDru is calculated.

$$D_{ru} = \omega_1 * (1 - CF_c) + \omega_2 * (1 - CF_{rm}) + \omega_3 * (1 - CF_s) \quad \text{.....(3)}$$

In the Equation 2. ω1 , ω2 , ω3 are the weights applied to the coefficient factor for the residual resources such as residual CPU core (CFc), residual memory (CFrm) and residual storage (CFs) and the distribution of all these utilized resources or residual resources can be represented as Dru.

In the paper we consider the physical memory with three resources mainly CPU core, Memory (RAM/Primary Memory) and Storage (Disk/Secondary Memory). The coefficient factor represents the utilization factor. For example if the CFrm value is small, it means the remaining or residual memory is more (1- CFrm), hence the system is very well balanced and not over utilized.

3.3 Working Principle

The proposed load balancing and virtual machine (VM) scheduling algorithm for cloud computing environment combines the property of particle swarm optimization and Ant colony optimization algorithm. Hence the name of the proposed algorithm is PSACO. In PSACO, the task request submitted for requesting the resources to execute the task will be fulfilled based on different criteria as illustrated in Algorithm 1. If the minimum requirement will be fulfilled by the available resources in the cloud computing environment then only the tasks are allowed or submitted into the system. After that all the parameter related to the task request and the physical machine are collected based on the current scenario and the proposed algorithm provides the solution in the form of virtual machine. For balancing the load among the physical machine in the proposed protocol assessment phase is used which provides the balancing level of the physical machine. Based on this assessment further load will be decided into the physical machine. The entire proposed hybrid meta heuristicalgorithm consist of different phases namely; pre refuse phase, parameter initialization phase, searching phase, particle some optimization phase, assessment phase and pheromone updatingphase. The detailed description on each phases are presented in below Sections.

3.3.1 Pre-Refuse Phase

This is the first phase of the proposed PSACO algorithm for virtual machine scheduling and load balancing. In this phase whenever a task request submitted to the cloud computing system it will first check whether the request could be fulfilled or not. To find out that the pre refuse procedure is applied, in this the total residual memory available in the physical machine is calculated and compared with the requested memory. If the requested memory is less than the residual memory in the physical machine the pre-refuse phase will allow the request to the next step otherwise the request will be refused in the first place. In this way, the size of the solution for PSACO algorithm would be reduced by the pre refuse method and resulting to the less processing time for the proposed algorithm.

For example if two tasks requests cr1, cr2arrive in the cloud computing system with the memory requirement of 2 GB and 4 GB respectively. Let's assume that in the tow available physical machine PM1 and PM2has 1 GB

and 2 GB of residual memory available respectively. As the maximum available memory is two GB, then the task request cr1requirement could be fulfilled and the pre-refuse method can provide PM2with 2 GB of resident memory for task request cr1, but the task request cr2would be refused due to unavailability of the minimum requirement of memory. So the task request cr1will be accepted for the proposed method to go into the next step.

The Proposed PSACO Algorithm for VM Scheduling

Input : Task-Requests and Physical Machine Resources.
Output: Scheduling for the Task-Requests.

Each time when Task-Request submitted into the system;
do Pre-Refuse(); ► Refuse the Task-Request with doesn't satisfy the condition.
if *(Task Request are admitted into the system for scheduling)* **then**
Parameter_Initialization(); ► All parameter need to be initialized.
while *(Termination condition not met)* **do**
Searching(); ► Gives the two-dimension Probability Matrix
Calculated with Equation 6,7,8,5 and 4.

$$\beta CU_n = \frac{Max_{CPU\_Utilization} - CU_n}{Max_{CPU\_Utilization}}$$

$$\beta M_n = \frac{RM_n}{Max_{Memory}}$$

$$\beta SU_n = \frac{Max_{Storage\_Utilization} - SU_n}{Max_{Storage\_Utilization}}$$

$$\beta_{n,m} = \Omega_1 * \beta CU_n + \Omega_2 * \beta M_n + \Omega_3 * \beta SU_n$$

$$P_{n,m} = \frac{\sigma_{n,m}^{\gamma} * \beta_{n,m}^{\lambda}}{\sum_{T_r \in R_r} \left(\sigma_{n,m}^{\gamma} * \beta_{n,m}^{\lambda}\right)}$$

Particle_Swarm_Optimization(); ► Provides the solution for ith ant with j number of iterations
Calculated with Equation 9 and Equation 10.

$$\delta_{i,j} = rw_1 * BS_{l,i} + rw_2 * BS_g$$

$$\alpha_{i,j} = Sol_{i,j} + \delta_{i,j}$$

Assessment(); ► Finds the fitness of the Physical Machine by Balance Level & Fine
Calculated with Equation 12 and Equation 11.

$$Fine = F_o(N_r) * F_r(R_r)$$

$$f_{fitness} = \begin{cases} BL * Fine & if(BL \leq 0) \\ \frac{BL}{Fine} & Otherwise \end{cases}$$

Pheromone_Updating(); ► Gets the updated value of Pheromone with the Present Best Solution
Calculated with Equation 14 and Equation 13.

$$\sigma_p = \begin{cases} f_{fitness}(BS_p) & if(PresentBestSolution) \\ 0 & Otherwise \end{cases}$$

$$\sigma_{j+1} = (1 - \varepsilon) * \sigma_j + \varepsilon * \sigma_p$$

end
end

Algorithm 1: The Proposed PSACO algorithm for Virtual Machine Scheduling.

This step may drastically reduce the solution size and help the scheduling algorithm with less search and resulting would be the efficient optimization algorithm. after the completion of this phase, the task requests who survives the pre reject method will get the opportunity to go into the proposed scheduling process.

3.3.2Parameter Initialization Phase

Initially in the PSACO algorithm, all the parameters need to be initialized as we will see in the subsequent phases. The parameters related to the physical machines resources such as CPU core, memory, disk storage its current status, maximum availability, coefficient and factors need to be initiated. Likewise the parameter related to the task request it's resource requirement are also initiated. All these initial values are used in the proposed algorithm for getting the global best solution.

3.3.3Searching Phase

After the pre refuse face the proposed PSACO algorithm, searches for the solution for each task requests available in the cloud system to fulfilled. in this proposed algorithm the Ant colony optimization is used to get the optimize solution for the requests. Here the ants are the task requests and the selected path for the aunts are the solution (computing resource). The solution for all ants can be calculated with the equation shown below. The result of this equation is the probability matrix which can map the ant and it's associative solution.

$$P_{n,m} = \frac{\sigma_{n,m}^{\gamma} * \beta_{n,m}^{\lambda}}{\sum_{T_r \in R_r} \left(\sigma_{n,m}^{\gamma} * \beta_{n,m}^{\lambda} \right)} \quad \text{.....(4)}$$

In the Equation 4, Pn,mrepresents the probability matrix, which gives the mapping of m^{th} task request to thenth Physical Machine. σn,mdenotes the pheromonesecreted by the ant andβn,mis used as a heuristic function. Whereasγ and λ are the influencer factor for both pheromone secreted and heuristic function respectively. Rr represents the residual resources available in the physical machine, it means that resources has not been allocated two any task request (Tr) in the cloud computing system.

The heuristic function βn,mcan be calculated based on the resources utilized and the distribution of the available resources. as we know in this paper the computing resources could be CPU, memory and storage. For calculating heuristic function we can use the equation shown below.

$$\beta_{n,m} = \Omega_1 * \beta CU_n + \Omega_2 * \beta M_n + \Omega_3 * \beta SU_n \quad \text{.....(5)}$$

In the Equation 5, βn,mrepresents the heuristic function based on the m^{th} task request and n^{th}physical machine mapping. Here Ω1, Ω2, Ω3 denotes the effective weights for the CPU core utilization, memory utilization and storage utilization respectively for the nth physical machine. Whereas the heuristic value of CPU utilization, memory utilization and storage utilizationrepresented byβCUn , βMn and βSUnrespectivelyand they also shows that how good the nth physical machine is in the basis of these three resources. It can be calculated using the equation shown below:

$$\beta CU_n = \frac{Max_{CPU\_Utilization} - CU_n}{Max_{CPU\_Utilization}} \quad \text{.....(6)}$$

In the Equation 6, the heuristic CPU utilization of nth physical machine could be represented as βCUn. This will be calculated by finding out the ratio of the difference between the maximum CPU utilization (MaxCPU_Utilization) and CPU utilization(CUn) of nth physical machine and maximum CPU utilization. If the CPU core is less utilized means more number of CPU cores are available in the nth physical machine.

$$\beta M_n = \frac{RM_n}{Max_{Memory}} \quad \text{.....(7)}$$

In the Equation 7, the heuristic Memory utilization of nth physical machine could be represented as βMn. This will be calculated by finding out the ratio of the residual memoryof nth physical machine and maximum memory. If the residual memory is more represents that the more amount of memory is underutilised and can be used for further task requests.

$$\beta SU_n = \frac{Max_{Storage\_Utilization} - SU_n}{Max_{Storage\_Utilization}} \quad \text{.....(8)}$$

In the Equation 8, the heuristic storage utilization of nth physical machine could be represented as βSUn. This will be calculated by finding out the ratio of the difference between the maximum storage utilization (MaxStorage_Utilization) and storageutilization (SUn) of nth physical machine and maximum storage utilization. If the storage is less utilised means more number of storage are available in the nth physical machine.

In the proposed PSACO algorithm these heuristic function gives the freedom for the ant to choose their path by taking the local decision. It means based on the availability of the resources the task requests could be mappedwith physical machines.

For example, if the heuristic values βCUn, βMn and βSUn are small it means the nthphysical machine is overutilized and need to be balanced or else it is showing a good performance and resources are underutilized.By using these values calculated in Equation 6, 7 and 8 the total resources available can be calculated through Equation 5.

3.3.4Particle Swarm Optimization Phase

In this phase after the searching and finding the local best solution by using the searching phase the particle swarm optimization is used to find more optimized result due to the advantage of using the local best solution and the global best solution using particle some optimization. By using this the searching speed for getting the optimised solution would be faster, itis another advantage of using particle swarm optimization.The solution could be found using the equation shown below.

$$\delta_{i,j} = rw_1 * BS_{l,i} + rw_2 * BS_g \quad \text{.....(9)}$$

In the Equation 9, δi, jrepresents the displacement of the ant with the speed δ. Here we can say that the displacement speed of ithant after the j number of iterations. Here rw1and rw2are the random weight applied on the local best solution of ithant after j number of iteration denoted as BSl,i, and global best solution denoted as BSgrespectively. From the Equation 9, we can find out that the displacement speed could be the result from the best solution from local or the best solution from global.

$$\alpha_{i,j} = Sol_{i,j} + \delta_{i,j} \quad \text{.....(10)}$$

In the Equation 10, αi, j is the solution of ith ant after the j number of iterations. Which can be calculated with the solution for ith ant after the j number of iterations denoted as Soli,j, having the displacement speed δi, j.

In the particle swarm optimization the displacement speed δwould be the speed of the roulette wheel,for selecting the solution from local bestBSl to global bestBSg.

3.3.5 Assessment Phase

At the end of particles sawm optimization phase, every ant are associated with its solution, it means the task requests are assigned with the available physical machine.The assessment phase can further help the ant to calculate the score for the scheduling and also for the current iterationthis assessment phase provides the best solution as well as it will update the global best solution. To finding the score the equation is used as shown below.

$$f_{fitness} = \begin{cases} BL * Fine & if(BL \leq 0) \\ \frac{BL}{Fine} & Otherwise \end{cases} \quad \text{.....(11)}$$

In the Equation 11, $f_{fitness}$represents the fitness function Which gives the score of the scheduling by using the balance level of the load in the physical machine. As shown in the above question if the balance level is less i.e BL ≤ 0 the Fine is applied to multiply with the balance level to provide the score. If the balance level (BL) is satisfactory,the Fine is applied to divide with balance level to provide the score. In this the Fine is calculated as shown in equation below.

$$Fine = F_o(N_r) * F_r(R_r) \quad \text{.....(12)}$$

In the Equation 12, the calculation of fine is represented. The fine is used to regulate the unsatisfactory solution. It can be calculated using two different fines. one is the fine on overloaded physical machine, when the earned chooses a physical machine that is already overloaded or this election may cause the overloading into the physical machine, which produce the lower fitness value. The second part is the fine caused by refusing the request.

The Fodenotes the fine for the overloaded physical machine. whereas Nr represent the number of physical machine selected by the new task requests which leads to the overloading by adding the number of timesthe same physical machine selected. whereas Fr denotes the penalty for refusing the task request and Rr represents the total number of refused requests. The value of Fine will be more if the number of overloaded physical machine and the number of refused requests will be more.

In this paper, the cloud computing system is creating the virtual machine to fulfil the requirement of submitted tasks. In this the type of the tasks or the detail information of the workload in the virtual machine are not known to the system. In the proposed PSACO algorithm the virtual machine scheduling is done based on the workload prediction for the new task requests by using the information such as status of the physical machine and the number of requests already scheduled.

3.3.6 Pheromone updating Phase

In the proposed PSACO algorithm, the Pheromone are updating based on the global solution. whereas in the current iteration whatever is the present best solution obtained are taken into consideration for the path where the pheromoneincreases. The pheromone decreases in the remaining paths with vaporization. The upgradation on the pheromonecan be calculated using the equation as shown below

$$\sigma_{j+1} = (1-\varepsilon) * \sigma_j + \varepsilon * \sigma_p \quad \text{.....(13)}$$

$$\sigma_p = \begin{cases} f_{fitness}(BS_p) & if(PresentBestSolution) \\ 0 & Otherwise \end{cases} \quad \text{.....(14)}$$

In the Equation 13, σj+1represents the next updated value of pheromone (σ). It will be calculated based the vaporization rate εusing the pheromone in the current jth iteration σj or the present best pheromone valueσp.

In the Equation 14, the present best pheromoneσp can be found using the fitness function with the present best solution$f_{fitness(}$BSp),if it is the best solution in the current iteration else zero otherwise.

3.3.7 Termination Phase

The proposed PSACO algorithm gives the scheduling solution for the virtual machine while updating the solution from local best to global best. In this process the algorithm need to executive in an iteration for multiple number of times. the algorithm may stop its execution when it's searches are finished and optimize result obtained. To fulfil that requirement, for the proposed algorithm to termination conditions are identified. When the maximum number of iteration is achieved the algorithm stops its execution and whatever the result is consider as the optimal result, that is the first condition for termination. In the second condition, if the global best solution is found then the algorithm will stop its iteration. It means that the solution what we get currently is not changing over subsequent iteration in that point we consider the solution as global best solution.

3.3.8 Time Complexity analysis of the Proposed Algorithm

In the journey of getting the best solution for load balancing and virtual machine (VM) schedulingthe algorithm perform many operation and executes many number of functions. To find out how complex or how simple the proposed algorithm is, we calculate the time complexity of the proposed algorithm. Initially we calculate the time complexity of all the phases individually and then the combined time complexity will be calculated.

In the first, pre-refuse phase of the PSACO algorithm where refuse those task requests, which required more memory than we currently have in the physical machineand allow only those task request who's memory requirement can be fulfilled by the current residual memory of the physical machine. So let's consider that n number of Physical machines and m number of task requests are there. So the time complexity to make the decision on pre-refuse phase would be O(m+n).

In the parameter initialization phase the parameters related to n number of Physical machines and m number of task requests need to be initialised. Hence, the time complexity will be O(m*n).

In the searching phase, every ant need to search from the m number of pathsand total n number of aunts are there which ultimately creates a solution matrix then the complexity will be of O(m2*n).

In the particle swarm optimization phase it need to establish m-dimension result space two times as well as path selection is done using two different solution. So the complexity of this phase will be O(m).

In the assessment phase, the status of the physical machine need to check by the algorithm. As we assume nnumber of physical machines are there in the system.So the time complexity would be O(n).

In the pheromone updating phase, the pheromone need to be updated for m*n path hence the update process is done m*n number of times as we have the establishment of n number of Physical machines and m number of task requests. So the time complexity will be O(m*n).

If we assume that in the system there are I number of ants which are looking for the optimal pathand the maximum number of iteration is j for getting the global best solution by the algorithm. hence the total time complexity of PSACO algorithm would be :O(m+n) +O(m*n) + O(i*j(m2*n + m + n + (m*n)) = O(m2* n * i * j) = O(m2nij).

4. Simulation Results and Analysis

In this section we analyse the performance of the proposed hybrid algorithm with different performance matrix and compare the result with some existing hybrid load balancing algorithm such as QMPSO[2] and GA-PSO[7]. The simulation is performed using the CloudSim3.0 tool for cloud computing and the stool is executing in the system with the Window 7 operating system having the hardware resources as Intel core i7 processor, 8 GB RAM, 3.4 GHz CPU.The cloud computing environment for the experiment is described throughTable 1 to Table 3. The simulation parameters and its associativeinitial values are illustrated in Table 4.

Table 1: Data Centre (Number - 1)

Parameter	$Architecture$	$OperatingSystem$	$VMMonitor$	$Cost$	$CostperMemory$	$CostperStorage$
Value	x86	Linux	Xen	3.0	0.05	0.001

Table 2: Virtual Machine (Number 10-60)

Parameter	$ProcessorSpeed$	$Memory$	$Bandwidth$	$ImageSize$	$VMMonitor$
Value	9726 MIPS	0.5 GB	1 GBPS	10 GB	Xen

Table 3: Physical Machine (Number - 20)

Parameter	$MIPS$	$Storage$	$VMMonitor$	RAM	$Bandwidth$	$Core$
Value	177,730	8.0 TB	Xen	16.0 GB	15 GBPS	6

There are two different scenarios have been considered for finding out the performance of the proposed algorithm; one is with the static workload environment where the task-requests are known to the system initially beforehand. Whereas in the second scenario a dynamic workload environment is considered where the task-requestsare dynamic and it is submitted to the system any time during the experiment. We considered total virtual machines varying from 10 to 60 and total number of task-requests from 100 to 600. The value of the maximum number of iteration MaxIterationwould be 200. The simulation results of the proposed algorithm and the existing algorithm are analyzed based on the load balancing metrices such as Maxspan,Energy Utilisation and Standard Deviations discussed below.

Table 4: Simulation Parameter Initialization

Parameter	σ_0	γ	λ	Ω_1	Ω_2	Ω_1	x_1	x_2	ε
Value	0.5	1	1	0.4	0.3	0.3	0.5	0.5	0.1
Parameter	Max_{Memory}	$Max_{CPU\_Utilization}$	$Max_{Storage\_Utilization}$	ω_1	ω_2	ω_1	F_o	F_r	
Value	32 GB	100	100	0.4	0.3	0.3	0.5	0.1	

4.1 Makespan

Makespan is the total time taken by a virtual machine to complete its scheduled tasks that can be measured in millisecond. In this paper two different scenario have been considered in the first scenario we find out the total Max pain for the static workload environment with fixed number of virtual machine as shown in Figure 1. The number of virtual machine we considered here is 50 which is fixed during this experiment. As the resulting graph shows that the proposed PSACO algorithm takes the less time than the existing protocol QMPSO[2] and GA-PSO[7] and completing the task faster with different number of task-requests.

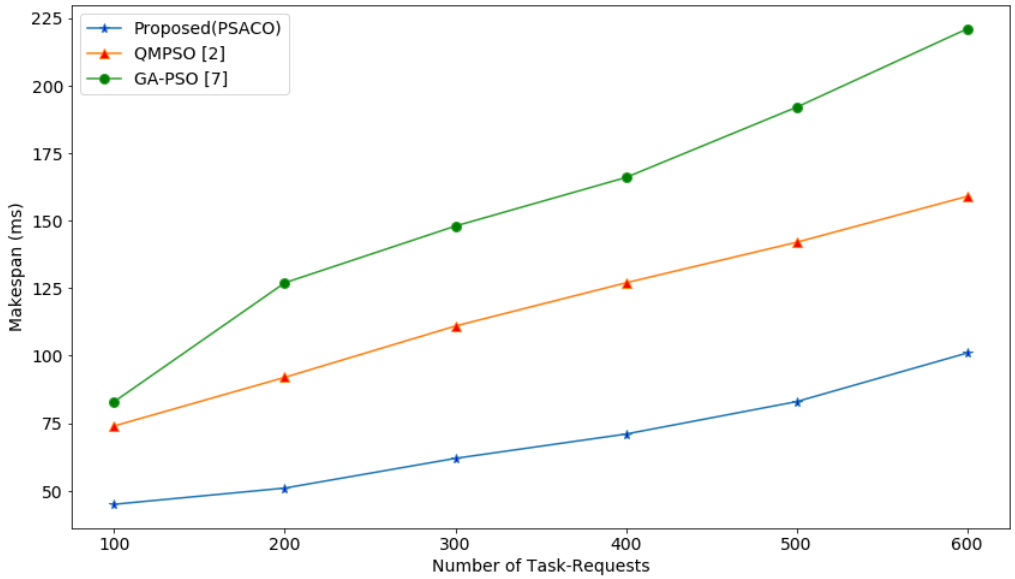

Figure 1: Total Makespan in the Static Workload Environment with Fixed Virtual Machines.

In the second scenario where we find out the total Max span in the dynamic workload environment with fixed virtual machines has illustrated in Figure 2. The number of virtual machine we considered here is 50 which is fixed during this experiment.As we can see from the resulting graph, the proposed PSACO algorithm adopting the dynamic scenario of workload and performing well in terms of total execution time of the task submitted to the virtual machine. In this result also the proposed algorithm outperformed the existing protocol QMPSO[2] and GA-PSO[7] by completing the execution of tasks earlier.

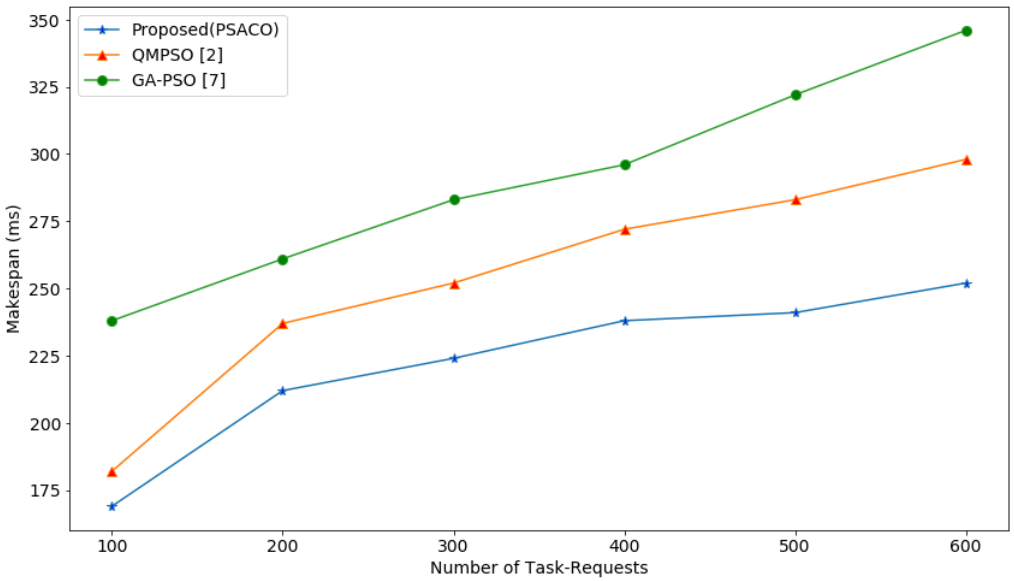

Figure 2: Total Makespan in the Dynamic Workload Environment with Fixed Virtual Machines.

In another environmental scenario we find out the total Makespan in the static workload environment by varying the number of virtual machines. Here we try to find out the performance of the proposed PSACO algorithm with fixed task-requests of400 with different number of virtual machine wearing from 10 to 60. The resulting graph can be shown in Figure 3, where it is clearly visible that the Makespan time is reducing based on the number of virtual machine available in the cloud environment. As a result the proposed algorithm is performing well and completing their task faster than the existing algorithms QMPSO[2] and GA-PSO[7].

4.2 Energy Utilization

The energy utilization performance metrics, shows the amount of energy consumed during the load balancing process by the algorithm. In the simulation during the execution of the task in differential radio are considered and averaged the energy utilized during these experiments. For finding out the energy utilization for the load balancing, we consider two different scenarios. In the first scenario the energy consumption is calculated for fixed number of virtual machine and different number of task-requests as shown in Figure 4. this resulting graph shows that if the number of tasks requests are wearing then how algorithm is consuming the energy for balancing the load in the cloud computing environment. As shown in the graph the proposed algorithm PSACO consuming less energy than the existing algorithms such as QMPSO[2] and GA-PSO[7].

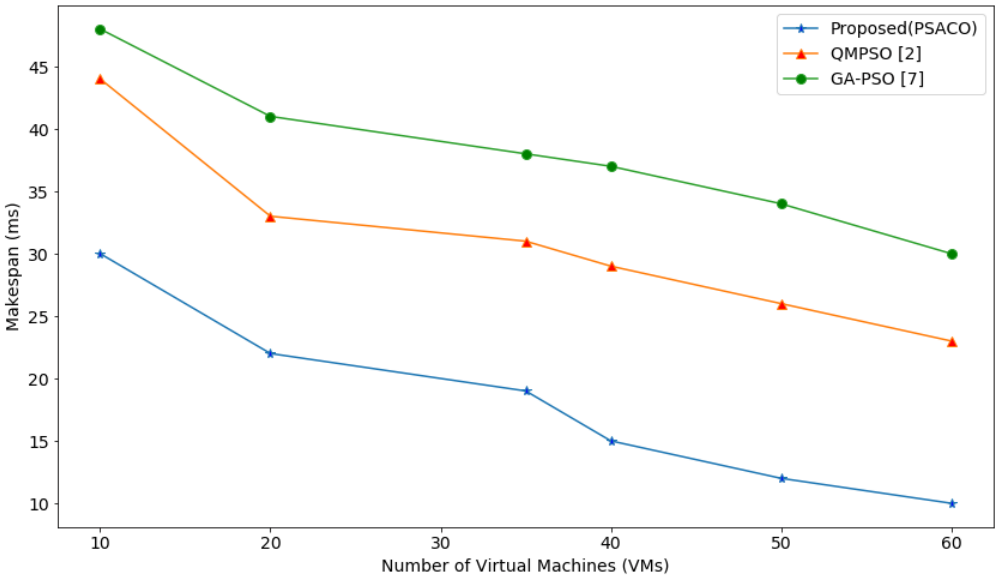

Figure 3: Total Makespan in the Static Workload Environment with Varying Virtual Machines.

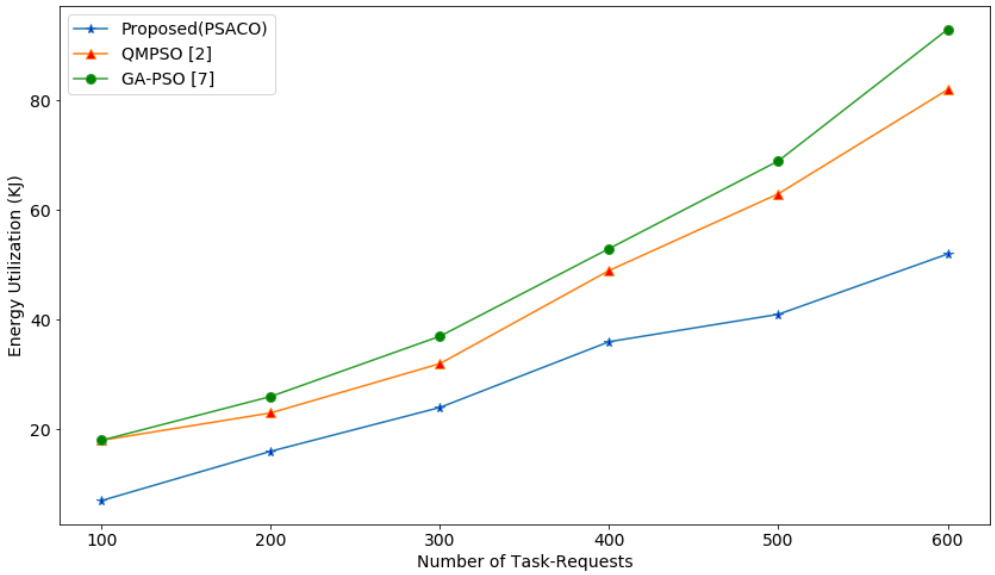

Figure 4: Energy Utilization for Fixed Virtual Machines & Varying Task-Requests.

In the second scenario, where the number of task request are fixed but there are different number of virtual machines in the system. the energy consumption for balancing the load by the algorithm with this scenario can be visualized with the Figure 5, where the number of virtual machines are varying from 10to 60 with the fixed number of task-request as 400. As the resulting graph shows that the energy consumed by the proposed algorithm is less than the existing algorithms QMPSO[2] and GA-PSO[7]. In this observation we can say that for balancing the load of the cloud computing environment the proposed algorithm is performing efficiently.

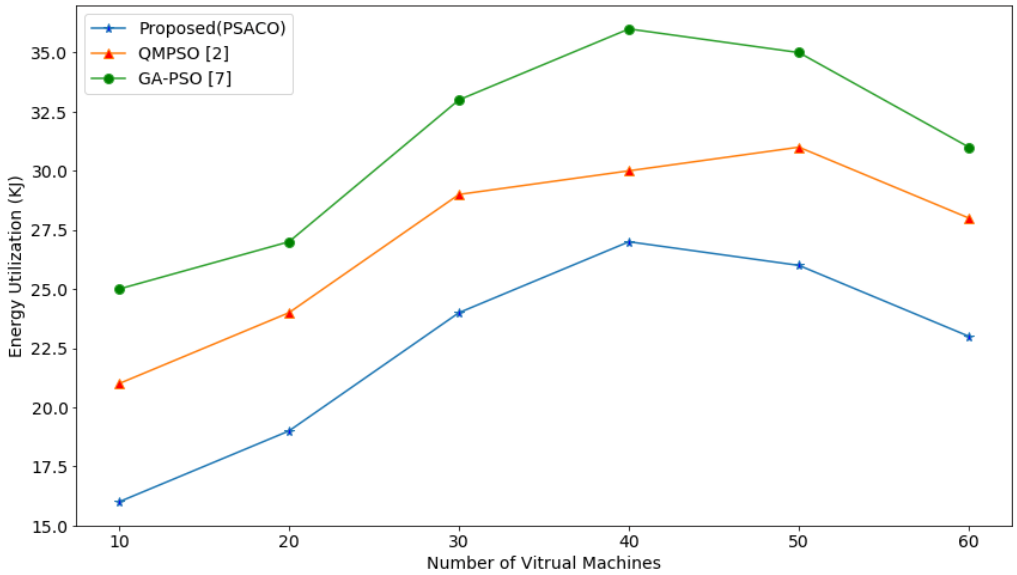

Figure 5: Energy Utilization for Fixed Task-Requests & Varying Virtual Machines.

4.3 Standard Deviation

This performance matrix shows the effectiveness of the algorithm used for virtual machine scheduling and load balancing. It illustrate the efficiency of the algorithm for getting the optimal solution. As shown in the resulting graph Figure 6, The standard deviation of the proposed algorithm is very less than the existing hybrid load balancing algorithms QMPSO[2] and GA-PSO[7]. By the result we can observe that the proposed PSACO algorithm is having the better capabilities of utilizing the resources as well as it can efficiently balance the load of the system as compared to other hybrid algorithms [2][7].

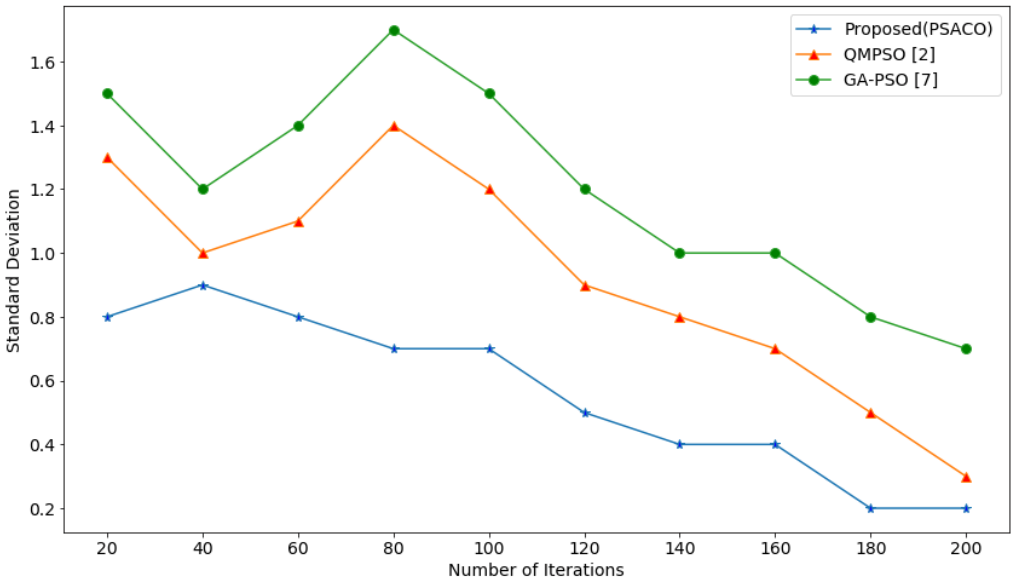

Figure 6: Standard Deviation of the Algorithms with number of Iterations.

5. Conclusion

In this paper ahybrid metaheuristic basedLoad Balancing and Virtual Machine (VM) scheduling algorithm has been proposed. This algorithm is based on Particle Swarm Optimization (PSO) and AntColony Optimization (ACO) called PSACO. The proposed algorithm consist of six different phases such as pre-refuse phase, initialization phase, searching phase, particle swarm optimization phase, assessment phase, pheromone updating phase. The proposed algorithm can work on the dynamic workload environment with minimum information of the task-request. It finds theglobal best solution for mapping the task-request to the virtual machine with load balancing. Finally the proposed algorithm has been implemented using CloudSim simulator and compared the result with two existing hybrid metaheuristic algorithms QMPSO[2] and GA-PSO[7]. For analyzing the performance of the algorithms the metrics such as Maxspan, energy utilization and standard deviation are used in different task-request and virtual machine scenarios. From the result it is concluded that the proposed hybrid metaheuristic algorithm (PSACO) for load balancing and virtual machine (VM) scheduling and is an efficient approach which provides the optimal solution for the cloud computing environment.

References (APA)

1. Amanpreet Kaur, Bikrampal Kaur, "Load Balancing Optimization based on Hybrid Heuristic-Metaheuristic Techniques in Cloud Environment", Journal of King Saud University – Computer and Information Sciences, Pages 1-12, 2019.
2. U.K. Jena, P.K. Das , M.R. Kabat, "Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment", Journal of King Saud University – Computer andInformation Sciences, Pages 1-11, 2020.
3. Geetinder Kaur and Sarabjit Kaur, "Improved Hyper-Heuristic Scheduling with Load-Balancing and RASA for Cloud Computing Systems", International Journal of Grid andDistributed Computing, Vol. 9, No. 1,Pages13-24, 2016.
4. Muhammad Junaid, Adnan Sohail, Adeel Ahmed, Abdullah Baz, Imran Ali Khan and HosamAlhakami, "A Hybrid Model for Load Balancing in Cloud Using File Type Formatting", Vol.8, Pages 118135-118155, 2020.
5. Srinivasa Rao Gundu, T. Anuradha, "Improved Hybrid Algorithm Approach based Load Balancing Technique in Cloud Computing", Global Journal of Computer Science andTechnology: B Cloud and Distributed, Vol. 19, Issue 2, Pages 35-42, 2019.
6. Subhadarshini Mohanty, Prashanta Kumar Patra, Mitrabinda Ray, Subasish Mohapatra, "A Novel Meta-Heuristic Approach for Load Balancing in Cloud Computing", InternationalJournal of Knowledge-Based Organizations, Vol. 8, Issue 1, Pages 29-39, 2018.
7. Ahmad M. Manasrah, Hanan Ba Ali, "Workflow Scheduling Using Hybrid GA-PSO Algorithm in Cloud Computing", Wireless Communications and Mobile Computing,Hindawi, Vol. 2018, Pages 1-16, 2018.
8. Keng-Mao Cho, Pang-Wei Tsai, Chun-Wei Tsai, Chu-Sing Yang, "A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing", Neural Computing &Applications, Vol. 26, Pages 1297–1309, 2015.
9. Gang Li, Zhijun Wu, "Ant Colony Optimization Task Scheduling Algorithm for SWIM Basedon Load Balancing", MDPI Future Internet, Vol. 11, Issue 4, Pages 1-18, 2019.
10. Shabnam Sharma, Sahil Verma, Kiran Jyoti, Kavita, "Hybrid Bat Algorithm for Balancing Load in Cloud Computing", International Journal of Engineering & Technology, Vol. 7, No. 4,Pages 26-29, 2018.
11. Youssef Fahim, Hamza Rahhali, Mohamed Hanine, El-Habib Benlahmar, El-Houssine Labriji, Mostafa Hanoune, Ahmed Eddaoui, "Load Balancing in Cloud Computing Using Meta-Heuristic Algorithm", Journal of Information Processing System, Vol. 14, No. 3, Pages 569-589, 2018.
12. Mala Yadav & Sachin Gupta, "Hybrid Meta-Heuristic VM Load Balancing Optimization Approach", Journal of Information and Optimization Sciences, Vol. 41, No. 2, Pages 577-586, 2020.
13. Shanchen Pang, Weiguang Zhang, Tongmao Ma, and Qian Gao, "Ant Colony Optimization Algorithm to Dynamic Energy Management in Cloud Data Center", Mathematical Problems in Engineering , Hindawi, Vol. 2017, Pages 1-10, 2017.
14. Gogi Reddy NarendrababuReddy,SingamsettyPhanikumar, "Multi Objective Task Scheduling Using Modified Ant Colony Optimization in Cloud Computing", InternationalJournal of intelligent Engineering & system, Vol. 11, No. 3, Pages 242- 250, 2018
15. A.I.Awad, N.A.El-Hefnawy, H.M.Abdel_kader, "Enhanced Particle Swarm Optimization For Task Scheduling In Cloud Computing Environments", International Conference on Communication, Management and Information Technology (ICCMIT 2015), Procedia Computer Science, Vol. 65, Pages 920-929, 2015.

16. R. M. Alguliyev, Y. N. Imamverdiyev, F. J. Abdullayeva, "PSO-based Load Balancing Method in Cloud Computing", Automatic Control and Computer Sciences, Vol. 53, No. 1, Pages 45–55, 2019.
17. Jean Pepe Buanga Mapetu1,2 & Zhen Chen1,2 &Lingfu Kong, "Low-time Complexity and Low-Cost Binary Particle Swarm Optimization Algorithm for Task Scheduling and Load Balancing In Cloud Computing", Applied Intelligence, Springer, Vol. 49, Pages 3308–3330, 2019.
18. Kumar Surjeet Chaudhury, Dr. Sabyasachi Pattanaik, Dr. Himanshu Sekhar Moharana, Sitakanta Pradhan "Static Load Balancing Algorithms in Cloud Computing: Challenges and Solutions" International Conference on Soft Computing And Signal Processing and published in Proceedings of 2nd ICSCSP 2019 named Advances in Intelligent Systems and Computing (Springer), Vol. 1118, 2019
19. Kumar Surjeet Chaudhury, Sabyasachi Pattanaik, Ashanta Ranjan Routray "Modified Meta-Heuristic Optimized Load-Balancing Algorithm For Cloud Computing Infrastructure" Solid State Technology (Scopus Index), Vol. 63, Issue 6, Pages 20687-207000, 2020.