Multi-Objective Dynamic Resource Scheduling Model for Allocating User Tasks in the Cloud Computing

G.B. Hima Bindu[a], K. Ramani[b] and C. Shoba Bindu[c]
[a] Research Scholar, Dept of CSE, JNTUACE, Anantapuramu, AP, India.
[b]Professor, Dept of Information Technology, SVCE, Tirupati, AP, India.
[c]Professor, Dept of CSE, JNTUACE, Anantapuramu, AP, India.

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 20 April 2021

Abstract: Cloud computing is the efficient distributed platform which manages all types of resources in the virtual manner. The major functionality of the cloud computing is to schedule the user tasks to the virtual machines. Many algorithms are proposed by the researchers to address the scheduling in cloud. Though there is some research gap need to be focused in the cloud scheduling. Minimizing the cost, makespan and deadline violations are difficult when large volume of tasks are assigned to the cloud. The multi-objective mechanisms are needed to address the two more objectives in the cloud scheduling model. This paper proposed the multi-objective algorithm based on the non-dominated method and crowding distance method. The proposed method computes the quality of service for the virtual machines before allocating to tasks to fulfil the requirements of the users. The performance of the proposed multi-objective model is evaluated based on the makespan, deadline violations and cost. The results prove the efficiency of the multi –objective model.

Keywords: Data Center, Scheduler, Resource Manager, Makespan

1. Introduction

Cloud computing is emerged as the potential distributed computing environment in the recent years. The major challenge faced by the cloud computing platform is scheduling of tasks. Many industries and academicians are concentrated on developing the solutions to the scheduling issues in Cloud computing [1-3]. The main difficulty in scheduling is proper allocation of tasks to the Virtual machines and effective management of resources and tasks in the large scale distributed cloud environment [4]. The major motive of task scheduling is to find the suitable resources and it will be easy when there is small number of tasks and resources. Similarly, when the user requirements are more to the cloud service, then there will be the requirement for optimal selection of virtual machines to obtain the quality service [5-7]. Another important aspect in cloud is load balancing and it is crucial for improving scalability and flexibility of the data centers. Load balancing mechanism transfers the tasks from overloaded virtual machines to under loaded virtual machines dynamically and improves the response time of the resources [8]. Resource scheduling mechanisms must have proper load distribution process among virtual machines. Some virtual machines are in idle state at the time of allocation and it leads to the economic issues [9]. It is clear that, with large volume of tasks and their complexity made the scheduling of proper resources to the tasks almost impossible [10]. Therefore, an efficient scheduling algorithm is required for resource management and load balancing.

This paper concentrates on developing the multi objective heuristic model where it has to minimize the makespan, task execution time and cost of the resources. The rest of the paper is organized as follows. Section 2 deals with the related work with respect to scheduling and load balancing in the cloud. Section 3 explains about the problem statement for resource scheduling. Section 4 deals with the multi objective resource scheduling model for cloud. Section 5 explains about the performance evaluation of the proposed and existing models. Finally section 6 concludes the research work.

2. Literature Survey

Many researchers addressed scheduling algorithms for cloud computing and their contribution is differentiated with traditional algorithms and Meta-heuristic algorithms. In traditional algorithms, the virtual machines need to be allocated through the physical machines and it incurs less cost and high performance. Meta heuristic algorithms are more efficient than the traditional algorithms on larger tasks.

2.1 Traditional Algorithms:

In [11], central load balancing algorithm was proposed for distributing the load to the virtual machines which are heterogeneous in nature. This method deals with the hardware of the virtual machines. In [12], the authors developed the min-min algorithm for minimizing the makespan. The load balancing is done based on the assigning the smaller tasks to the low powered virtual machines. In [13], max-min algorithm was proposed for task scheduling. The nature of the max-min algorithm is same as that of the min-min algorithm, but the smaller difference is to detect the maximum power virtual machines.

2.2 Meta Heuristic Algorithms:

In [14], the authors proposed the trust based scheduling model which deals with the cost of the model. It increments the cost based on the execution time of the tasks. The balance policy is employed to manage the trust, time and cost. The proposed algorithm is efficient in optimizing the time and cost. But this model is not sufficient to execute the large number of tasks. In [15], the authors developed the ant colony model which contains seven heuristics to find the optimal solution to the workflow scheduling. The ACO model allows the ants to find the optimal path with the help of pheromone value. The major drawback of the proposed model is the completion time.

In [16], the authors proposed improved scheduling algorithm for resources scheduling in cloud. This algorithm considered the CPU utilization and resource availability as the objectives for the algorithm. The results proved the improvement in resource availability and CPU scheduling. The major limitation of the improved algorithm is makespan. In [17], the Fuzzy logic based Genetic Algorithm was proposed to optimize the task scheduling. The authors considered the task clustering as the major part in resource scheduling to finalize the decision. In [18], the authors concentrated on developing the heuristic approach based on the makespan and completion time for optimal scheduling. The drawback of the proposed approach is less concentration on energy consumption. In [19], the authors developed the algorithm for partitioning the direct acyclic graph (DAG) and allocate the threshold finishing time for subtasks based on the requirements set by the clients. This algorithm allocates the resources to the partitions and the execution time is reduced with lower cost. In [20], the authors developed a backward scheduling algorithm called as particle critical paths (PCP). This algorithm considers time constraint at the time of scheduling process. This scheduling algorithm failed to reach time constraint and they have to be rescheduled using the MDP. It involves high time complexity due to the number of rescheduling's happens at the time of algorithm execution.

3. Problem Definition

Task scheduling is defined as the process of assigning task to the virtual machines based on their requirements. Load balancing is the major part which should be consider for scheduling. In scheduling, two solutions need to be taken care at the time of load balancing: the primary solution is the execution order of tasks which decides the makespan of the algorithm and second solution is to execution each task in separate processor [21-24]. This paper addressed the limitations of the cloud computing. Cloud computing consists of data centers. Each data center is associated with virtual machines and each virtual machine contains the homogenous or heterogeneous CPUs to execute the tasks [25]. Figure 1 explains about the framework of cloud computing.

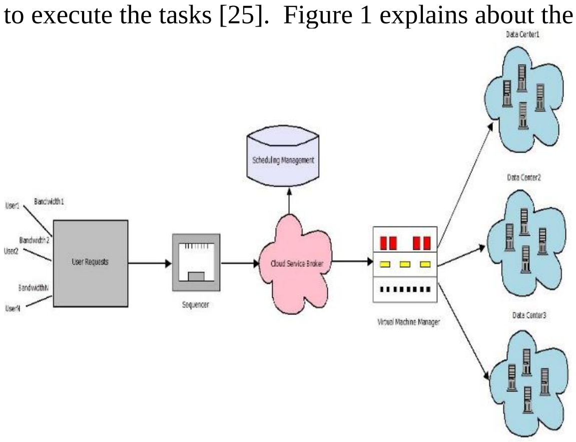

Figure 1 – Framework of Cloud Computing

Cloud computing works on the environment of internet and the users from different regions access the resources by request/reply mechanism. Cloud data centers are responsible for processing the user requests which are located in different geographical regions. Cloud server broker is handles the resource management in the cloud computing. Each cloud service broker is associated with sequencer, scheduling management and virtual machine manager.

Sequencer: The Job of the sequencer is to prioritize the tasks based on their dependency in the form of directed acyclic graph (DAG) and it is submitted to the cloud service broker. The tasks can be computed based on the architectures designed for the particular machines.

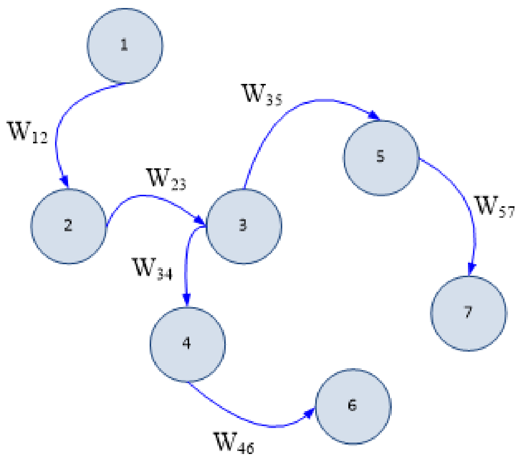

Figure 2 – Directed Acyclic Graph for User Tasks

Scheduling Management: The cloud service broker is responsible for performing the scheduling and load balancing with respect to decreasing the transition time. The service broker estimates the completion time of each task and also dependencies of the tasks based on the DAG and performing the scheduling process to reduce the completion time.

Virtual Machine Management: The service broker monitors the active virtual machines and their characteristics like CPU, bandwidth, and memory for allocation of tasks.

The user requests in the cloud computing is handled by the service broker in the form of directed acyclic graphs. Figure 2 shows the sample DAG for tasks with their dependencies.

4. Proposed Algorithm For Task Scheduling

The proposed algorithm concentrates on the multi objective optimization where it has to minimize the makespan, task execution time and cost of the resources. We can omit any of the objectives in designing the task scheduling algorithms, because it compromises the other objectives.

In task scheduling algorithm, we consider n number of tasks T={t_1, t_2,….t_n} and m number of virtual machines VM={VM_1, VM_2…VM_m} for scheduling. In this research work, multi objective scheduling model is considered based on the modified non-dominated sorting algorithm. In the initial step, user submits the tasks to the cloud environment. The service broker in the cloud receives the tasks and submits to the scheduler. The multi objective scheduler separates the tasks in to non-dominated sets.

4.1 Traditional Algorithms

To reduce the cost of VMs and minimize the constraint of deadline, it is required to remove the under loaded and overloaded virtual machines. The number of Virtual machines selection is depends on the received tasks. Eq. 1 shows the count of virtual machines based on the total tasks received.

$$C(VM) = \frac{Load(Total\_Tasks)}{Max(VM\_Mips)} \quad (1)$$

$$Load(Total\_Tasks) = \sum_{m=1}^{n} Load(Tasks) \quad (2)$$

4.2 Sorting The VMs

After finding the count of virtual machines which is required for scheduling, sorting of the virtual machines need to be performed. Quality of Service (QoS) is required to measure the user requirements. The user has their requirements in selecting the services from the cloud. In this research work, Bandwidth, cost and execution time are considered as the QoS to defines the services. Eq. 3 shows the QoS function for service which is obtained through the QoS vectors.

$$QoS(s) = \sum_{m=1}^{r} \frac{Q_{j,m}^{\max} - q_m(s)}{Q_{j,m}^{\max} - Q_{j,m}^{\min}} \quad (3)$$

Where r represents the number of attributes of QoS selected for virtual machine, qm(s) represents the value of the attribute for a service s. $Q_{j,m}^{\max}$ and $Q_{j,m}^{\min}$ are the maximum and minimum values of the mth attribute selected . After computing the QoS function, sorting is applied on the virtual machines and arranged in the descending order. The larger tasks are sent to the virtual machines with highest computing capacity.

4.3 Non-Dominates Sorting Approach For Tasks

In multi objective approach, there will be at least two objectives need to be considered for evaluation. In some situations, there will be common priority for both objectives and they cannot be sorted in the domination approach. In this sorting approach, the tasks are selected based on the minimum size which reflects the execution cost and makespan. Eq. 4 and Eq. 5 shows the two objective functions defined for scheduling.

$$Min\ f(T(size_m) = T(size_m) \quad \because \forall j \exists i \quad f(T(size_i)) \leq f(T(size_j)) \tag{4}$$

$$Min\ f(T(\cos t_m) = T(\cos t_m) \quad \because \forall j \exists i \quad f(T(\cos t_i)) \leq f(T(\cos t_j)) \tag{5}$$

Where T(size) represents the size of the tasks, T(cost) represents the cost incurs to execute the tasks. Eq. 6 shows the calculation of cost for task execution in virtual machine.

$$T(\cos t) = \sum_{m \in P(VM)} VM(\cos t_m) \times T(Exec_m) \tag{6}$$

Where P(VM) represents the count of service providers which are providing the virtual machines. The cost of the virtual machine is dependent on the number of CPUs allocated. Eq. 7 shows the cost of the virtual machine with respect to the service provider and Eq. 8 shows the execution time of the task based on the virtual machine allocated.

$$VM(\cos t) = \frac{\cos t\_per\_\sec ond()}{MIPS\_to\_execute\_PE()} \tag{7}$$

$$T(Exec_m) = \sum_{i=1}^{n} \frac{PE \times Length + Size\_of\_the\_Output}{MIPS \times PE} \tag{8}$$

4.4 Calculation Of Crowding Distance

The crowding distance (CD) is calculated by computing the difference between the two individuals and for next two individuals to it. Eq. 9 shows the crowding distance of the objective functions are calculated.

$$CD = \left| f_{T(size)}^{i+1} - f_{T(size)}^{i-1} \right| + \left| f_{T(\cos t)}^{i+1} - f_{T(\cos t)}^{i-1} \right| \tag{9}$$

Where, f^{i+1} and f^{i-1} are the previous and next individuals of the present individual in each objective.

4.5 Scheduling Tasks To Virtual Machines

The sorting is applied for virtual machines and Tasks. In the next step, the tasks are placed in the execution queue based on the priority and allocate the first task to the first virtual machine which is there in the sorted queue. Eq. 10 shows the normal execution rate of the virtual machine. The virtual machine normal execution rate is compared with the threshold rate. If the execution rate is less than the threshold, then the next task is allocated to the virtual machine. If the execution rate is more the threshold then the new virtual machine is allocated. This process is chosen due to overcome the deadline constraint issue.

$$VM(N\_Rate) = \frac{Current\_load(VM)}{Max\_MIPS(VM)} \tag{10}$$

4.6 Penalty Function

In cloud computing, the resources are allocated at any time to the customers based on the requirements. In this research work, resource allocation approach was proposed to reduce the deadline violation and total cost. Eq. 11 shows the total cost of the scheduling, where it considers the virtual machine cost and the penalty cost which incurs at the time deadline violation.

$$Min(Tot\_\cos t) = Cost(VM) + Pen\_\cos t \tag{11}$$

$$Cost(VM) = \sum_{i=1}^{I} Cost(VM_i) \tag{12}$$

$$Pen\_\cos t = \sum_{i=1}^{r} pen\_\cos t_i \tag{13}$$

Where, I and r represents the count of the VMs and count of the tasks. Eq. 14 shows the task penalty cost.

$$Pen\_\cos t_i = Missed(deadline) \times Pen\_rate \tag{14}$$

Where *Pen_rate* represents the cost/unit which incurs on the time delay and *Missed(deadline)* represents the how much time taken after the deadline to complete the task execution.

4.7 Multi Objective Scheduling Algorithm

Algorithm 1 shows the multi objective scheduling algorithm, where the set of tasks and virtual machines are selected based on the Non-Dominated mechanism and finds the objective functions using the crowding distance process.

Algorithm 1: Multi objective scheduling

Input: $T=\{t_1, t_2,...t_n\}$, $VM=\{VM_1, VM_2...VM_m\}$
Output: Scheduling
Begin

1. Apply Eq. 1 to find the count of virtual machines required for executing arrival tasks
2. Compute the QoS for virtual machines using Eq. 3
3. Sort the virtual machines in the form of descending order
4. for i in 1 to VM-1 do
5. for j in 1 to t-1 do
6. Compute the normal execution rate of the virtual machine using Eq. 10
7. If VM(N_Rate) < Th_Rate then
8. Map t_i to VM_j
9. End if
10. End for
11. End for
12. For the next tasks in the queue
13. Compute the *Min(Tot_cost)* of the model from Eq. 11
14. If *Min(Tot_cost)* < *Th_rate* then
15. Assign the tasks to th virtual machines
16. Else
17. Virtual machines are overloaded
18. Assign virtual machines by calculating the cost function using the Eq. 12
19. End if
20. End for
21. If *Missed(Deadline)* then
22. Impose the pen_cost by using the Eq. 13
23. End if
24. If Load(VM) =0 then
25. Release the virtual machine
26. End if
27. Return scheduling set of tasks and virtual machines

End

5. Experimental Analysis

The proposed model is simulated using the cloudsim 3.0.3 toolkit [26]. The major objectives selected for the simulation is to minimize the makespan value, removing the deadline violation of the users, reducing the cost for execution and effective utilization of VMs. Tables 1 and 2 show the characteristics of the tasks and resources.

Table 1: Task parameters

Parameter	Value
Size of the File	1024 MB-4096 MB
Number of Tasks	100-1000
Length of the Tasks	2000-4000MIPs
Output Size	50 MB

Table 2: Virtual machine Parameters

Parameter	Value
No. of VMs	5-20
Storage	30GB-128GB
RAM	2GB- 4GB
Computing Capacity of the CPU	1860 MIPs
Bandwidth	200 Mbps
No. of CPUs	1-4
Cost	[0.12$ to 1.2$]

The performance of the proposed model is compared with other existing model like MOTS [27] and OSACO [28] algorithms. Figure 3 shows the makespan of the proposed and existing algorithms. The makespan is directly proportional to the total number of tasks. When the assigned tasks are high we can observe the major difference in the makespan of the algorithms. For instance, when the tasks are 1000, then the proposed model recorded 23% less makespan compared with MOTS and 19% less compared with OSACO. It is due to the proposed model where it selects the VMs with high capacity for larger tasks.

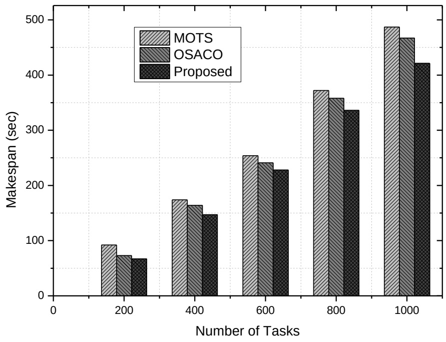

Figure 3 – Makespan Vs Number of Tasks

Figure 4 shows the deadline violation of the tasks in the proposed and existing algorithms. It is observed that the deadline violations of the proposed model are less compared with MOTS and OSACO. The deadline violation is used in the proposed model is to reduce the cost. The proposed model does not consider the load for allocation of virtual machines.

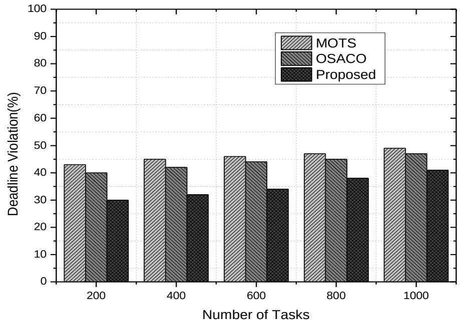

Figure 4 – Deadline Violations Vs Number of Tasks

Figure 5 shows the comparison of the system cost in proposed and existing models. The system cost will increase based on the increase in allocating virtual machines and number of tasks. In the proposed model, the cost is more study than the existing algorithms. The cost is taken in to consideration in all the stages of the proposed model. Hence the proposed model incurs less cost to execute the tasks compared with existing algorithms.

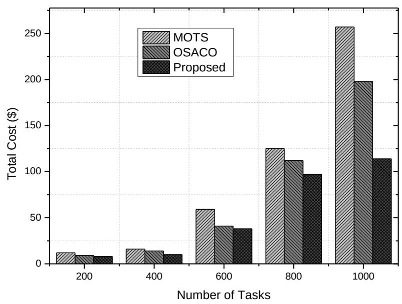

Figure 4 – Total cost Vs Number of Tasks

6. Conclusion

This paper proposed multi-objective dynamic resource scheduling model for cloud computing. The multi-objective approaches considered for scheduling are minimizing the cost, execution time and deadline violations. The proposed model considered the non-dominates method for sorting the virtual machines and task in the execution queue. The crowding distance method is also used to reduce the cost of the system. The performance of the proposed model is compared with other existing model like MOTS and OSACO algorithms. The proposed model recorded 23% less makespan compared with MOTS and 19% less compared with OSACO when there are 1000 tasks for execution. The deadline violations and cost of the system also reduced in the proposed model.

References

1. C. J. Huang, C. T. Guan, H. M. Chen, Y. M. Wang, S. C. Chang, C. YuLi, C. H. Weng, “An adaptive resource management scheme in cloud computing.” *Eng Appl Artif Intell*, Vol. 12, issue. 26, 382–389, 2013.

2. R. Arunarani, D. Manjula, V. Sugumaran "Task scheduling techniques in cloud computing: a literature survey," *Future Gener Comput Syst*, Vol. 91, 407–415, 2019.
3. R. T. Neto, M. G. Filho, "Literature review regarding ant colony optimization applied to scheduling problems: guidelines for implementation and directions for future research." *Eng Appl Artif Intell* Vol. 26, issue 2, 150–161, 2013.
4. S. K. Panda, S. S. Nanda, S. K. Bhoi, "A pair-based task scheduling algorithm for cloud computing environment." *J King Saud Univ Comput Inf Sci,* 1–12, 2018.
5. L. F. Bittencourt, A. Goldman, E. R. M. Madeira,N. L. S da Fonseca, R. Sakellariou, "Scheduling in distributed systems: a cloud computing perspective." *Comput Sci Rev* Vol. 30, 31–54, 2018.
6. N. Bansal, A. Maurya, T. Kumar,M. Singh, S. Bansal, "Cost performance of QoS driven task scheduling in cloud computing." *Proc Comput Sci*, Vol. 57, 126–130, 2015.
7. T. S. Somasundaram, K. Govindarajan, "CLOUDRB: a framework for scheduling and managing high-performance computing (HPC) applications in science cloud." *Future Gener Comput Syst*, Vol. 34, 47–65, 2014.
8. F. Abazari, M. Analoui, H. Takabi, S. Fu, "MOWS: multi-objective workflow scheduling in cloud computing based on heuristic algorithm." *Simul Modell Pract Theory*, 1–19, 2018.
9. M. Abdullahi, M. AsriNgadi, S. M. Abdulhamid, "Symbiotic organism search optimization based task scheduling in cloud computing environment." *Future Gener Comput Syst,* Vol. 56,640–650, 2016.
10. F. Juarez, J. Ejarque, R. M. Badia, "Dynamic energy-aware scheduling for parallel task-based application in cloud computing." *Future Gener Comput Syst,* Vol. 78, 257–271, 2018.
11. G. Soni, M. Kalra, "A novel approach for load balancing in cloud data center." *In: Advance Computing Conference (IACC)*, 1-8, 2014.
12. G. Patel,R. Mehta,U. Bhoi, "Enhanced load balanced min-min algorithm for static meta task scheduling in cloud computing." *Proc Comput Sci*, Vol. 57, 545–553, 2015.
13. U. Bhoi, P. N. Ramanuj, "Enhanced max–min task scheduling algorithm in cloud computing." *Int J Appl Innov Eng Manag*, Vol. 2, 259–264, 2013.
14. W. Tan, Y. Sun, L. X. Li, G. Lu, T. Wang, "A trust service-oriented scheduling model for workflow applications in cloud computing," *IEEE Systems Journal*, Vol.8, Issue. 3, 868–878, 2013.
15. W.-N. Chen, J. Zhang, "An ant colony optimization approach to a grid workflow scheduling problem with various qos requirements," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,* Vol. 39, issue. 1, 29–43, 2009.
16. B. Song, M. M. Hassan, and E. N. Huh, "A novel heuristic-based task selection and allocation framework in dynamic collaborative cloud service platform." *IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom)*, 360-367, 2010.
17. S. Tayal, Tasks scheduling optimization for the cloud computing systems. *IJAEST-International Journal of Advanced Engineering Sciences and Technologies,* Vol.1, issue 5, 111-115, 2011.
18. J.F. Li, J. Peng, X. Cao, and H. Y. Li, "A task scheduling algorithm based on improved Ant Colony Optimization in cloud computing environment." *EnergyProcedia*, Vol. 13, 6833-6840, 2011.
19. T. A. L. Genez, L. F. Bittencourt, and E. R. M. Madeira, "Workflow Scheduling for SaaS/PaaS Cloud Providers Considering Two SLA Levels," *IEEE/IFIP NOMS*, 906-912, Apr. 2012.
20. J. Yu, R. Buyya, and C. K. Tham, "Cost-based Scheduling of Scientific Workflow Applications on Utility Grids," *Int'l. Conf. e-Science and Grid Computing*, 140–47, 2005.
21. D. Chaudhary,B. Kumar, "Cloudy GSA for load scheduling in cloud computing." *Appl Soft Comput,* Vol. 71, 861–871, 2018.
22. L. Ismail, A. Fardoun, "EATS: energy-aware tasks scheduling in cloud computing systems," *Proc Comput Sci*, Vol. 83, 870–877, 2016.
23. RK Jena, "Energy efficient task scheduling in cloud environment," *Energy Proc,* Vol. 141, 222–227, 2017.
24. K Li, "Scheduling parallel tasks with energy and time constraints on multiple many core processors in a cloud computing environment," Future *Gener Comput Syst,* Vol. 82, 591–605, 2018.
25. J. Delaram, O. F. Valilai, "A mathematical model for task scheduling in cloud manufacturing systems focusing on global logistics," *Proc Manuf* , Vol. 17, 387–394, 2018.
26. R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, R. Buyya. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." *Software: Practice and experience*. Vol. 41, No. 1, 23-50, 2011.
27. V. Lakra, D. K. Yadav. "Multi-objective tasks scheduling algorithm for cloud computing throughput optimization." *Procedia Computer Science*. Vol. 48, 107-13, 2015.
28. G. B. Bindu, K. Ramani, and C. Shoba Bindu. "Optimized Resource Scheduling using the Meta Heuristic Algorithm in Cloud Computing." *IAENG International Journal of Computer Science,* Vol. 47, issue 3, 360-366 2020.