International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

Teaching-Learning Based Task Scheduling Optimization in Cloud Computing Environments

Ramakrishna Goddu, Kiran Kumar Reddi

Abstract*: *Generating optimal task scheduling plans in cloud environments is a tedious task as it is a np-hard problem. The optimal resource allocation in cloud environments involves more search space and time consuming. Therefore, recent researchers are focused on implementation of artificial intelligence to solve task scheduling problem. In this paper, a new and efficient evolutionary algorithm named teaching-learning based algorithm has been implemented first time to solve the task scheduling problem in cloud environments. The current research work considers the task scheduling problem as a multi-objective optimization problem. The proposed algorithm finds the best solution by minimizing the execution time and response time while maximizing the throughput of all resources to complete the assigned tasks.

Index Terms*: *teaching-learning based optimization, task scheduling, cloud computing, multi-objective optimization, resource allocation and throughput.

I. INTRODUCTION

Cloud computing provides sharing of computing resources and data storage, and allows its users to access information to utilize its services over the internet and central remote servers on demand [1]. The resources are shared between cloud clients through virtualization technique. Virtualization divides single computational resource into multiple independent execution environments, also known as virtual machines [2]. An inefficient virtual machine is nothing but a resource which decrease the performance and increases cost of the system [3]. The main objective of task scheduling is to increase performance and to decrease the task completion time by developing an efficient scheduling algorithm. A typical task scheduling process is represented in Fig.1.

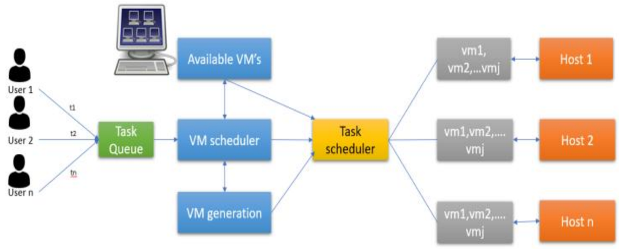

Fig.1 Flow process of task scheduling in cloud environment

Revised Manuscript Received on 30 July 2019.
***** Correspondence Author

Ramakrishna Goddu*, Research scholar, Department of computer science, Krishna University, Machilipatanam, Andhra Pradesh, India

Kiran Kumar Reddi, Assistant Professor, Department of computer science, Krishna University, Machilipatanam, Andhra Pradesh, India

Based on the priority, the task is allocated to the virtual machine and then it is mapped to suitable physical machine. The virtual machine is selected according to the task necessities like CPU power and cost, after that the task scheduler allocates task to the selected virtual machine [3].

Rule based algorithms can be implemented to solve optimal resource allocation problems. Ali et al. [4] introduced grouped task scheduling algorithm(GTS) algorithm for scheduling tasks into services which are QOS driven. This algorithm combines tasks with similar attributes that forms a category. These categories helps to know priority of tasks and then these tasks are scheduled to available resources. Tasks with high value of attributes are firstly scheduled and secondly tasks with less execution time is scheduled first i.e., with low latency. GTS performs optimal scheduling when compared to min-min and TS algorithms. Wu et al. [5] proposed a QOS driven based task scheduling algorithm in cloud computing. This algorithm schedules the tasks based on their priority and the time taken to complete the task, i.e, it allocates the resource which takes less time to execute the given task. Kong et al.[6]described a dynamic task scheduling algorithm with fuzzy prediction in virtualized data centres. Virtualization comes with some provocations such as task scheduling and resource management. This algorithm considers availability and responsiveness to solve task scheduling. A Fuzzy based prediction tool is developed to represent uncertain work load and availability of vogue of virtualized server nodes by using typ-1 and type-2 fuzzy logic systems. This work obtains load balancing and optimal performance. But the rule based algorithms may fail when the cloud data is vague. There are various optimization techniques based on artificial intelligence have been developed to solve task scheduling problem [7]. But, no single technique is globally accepted. Genetic algorithm (GA) is one of the traditional evolutionary algorithm and several researchers have been implemented this algorithm to solve task scheduling problem.

Ge and We [8] proposed genetic algorithm based techniques to solve task scheduling problems in cloud computing. In this research, the entire group of tasks in the job que are examined and then scheduling decision is taken by reducing the make span in order to achieve better load balancing. Jang et al.[9] introduced task scheduling technique with the implementation of genetic algorithm in cloud environments. The algorithm focused on user benefits such as QOS and profits to cloud providers. Whereas the previous developed models concentrated on either or combination of minimum execution time or work load. This work schedules the with preference of user satisfaction by evaluating the GA based fitness function. This process is iterated until an optimal task schedule achieved.

They compared the results with previous works and found it performed well. Kaur and Verma [10] developed a modified genetic algorithm by addressing a fitness to acheive minimum completion time in regard to a single user only. Experiment results showed that the developed algorithm achieves good performance under heavy loads.

The next famous population based algorithms are Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO). Medhat Janmohammadi and Babazade [11] proposed an ant based task scheduling in cloud environments. The researchers concentrated on minimizing the make span by implementing ACO. Finally, the obtained results are compared with previous algorithms and found efficient. Qiang Guo [12] addressed a algorithm to optimize task scheduling problem in cloud computing based on ant colony algorithm. The algorithm initializes the pheromone and is updated along with the developed heuristic function. They performed experiment by comparing results with Min-Min and found the algorithm obtained better make span, less cost and load balance. Zuo et al.[13] introduced a multi-objective optimization scheduling method based on ant colony system. The research used the performance and budget constraint functions to examine the past to give the feedback on the quality. This feature prevents the ACO from local optimal solution and then from the feedback, quality of the solution is obtained.

Particle swarm optimization is a social inspired evolutionary algorithm, which has been implemented for solving various engineering problems [14-16]. Past studies considered particle swarm optimization (PSO) based task scheduling is more efficient for dynamic task scheduling. Juan et al.[17] proposed a PSO based task scheduling algorithm to overcome the problems of cloud computing. They developed cost vector model which measures scheduling schemes cost and solution was developed based on input task and QOS parameters. This method is proven to effective but it is with more complexity. Krishnasamy [18] proposed a hybrid particle swarm optimization task scheduling algorithm which decreases the average operation time and increases the usage of resources and provides the resources according to the user tasks. Alkayal et al.[3] developed an PSO based multi object task scheduling by introducing a new approach of ranking. Here, the tasks are allocated to virtual machines according to the rank, which decreases the waiting time and increase system performance.

A.I.Awad et al. [19] introduced a new mathematical model using load balancing mutation a particle swarm optimization(LBMPSO) for task scheduling to achieve reliability and availability which are cumbersome parameters in cloud computing. This LBMPSO model maintains load balancing while distributing tasks to the resources available and the failure tasks are rescheduled. This feature made the model reliable. RK Jena [20] proposed task scheduling algorithm by the means of multi objective nested particle swarm optimization(MOPSO) which mainly focused on reducing power consumption and processing time. Results shown that when compared to BRS and RSA the MOPSO is efficient with multi objectives. Entisar S. Alkayal et al. [21] developed a model for optimized task scheduling based on particle swarm optimization with multi objective optimization for resource allocation. In this they introduced a new concept of ranking strategy, based on the results obtained from this ranking strategy the scheduling is done that finds the best virtual machine which suits the task. Results shown that this algorithm improves throughput and decreases latency.

One of the famous and recent developed biological inspired algorithm, Artificial Immune System (AIS) has been implemented by several researchers to solve many engineering problems. Shu, Wang and Wang [22] proposed a novel energy efficient resource allocation algorithm based on immune clonal optimization for green cloud computing. The authors considered the cost and energy consumption as parameters to minimize. In immune algorithms performs the clonal selection process to obtain the optimal solution. R .k. Jena [23] addressed an energy efficient algorithm for task scheduling algorithm in cloud computing by using clonal selection algorithm. By taking into consideration, optimization of energy consumption and make span. The clonal operator is an antibody and its affinity is tested to send the new antibodies which have high affinity. The proposed method performs better than scheduling algorithms and random task plans.

However, most of the developed algorithms faced local minima during the implementation. To overcome this problem, researchers are focused on implementation of hybridization of any two of the evolutionary algorithms. To find a solution to the problems of cloud computing in task scheduling, Dordaie & Navimipour [24] proposed a hybrid particle swarm optimization and hill climbing algorithm which was properly scheduled, but it takes more time for task completion. Liu et al.[25] described a task scheduling algorithm with the help of hybridised genetic ant colony systems. The authors focused on finding a solution to slow convergence problem caused by initial pheromone deficiency of ACO and then GA is integrated to find the optimal solution. Laili et al.[26] introduced an energy adaptive immune – genetic algorithm for task scheduling algorithm in cloud manufacturing systems. The developed technique improve the searching diversity which depends on immune strategy and capable of existing crossover and mutation probability. This algorithm maintains the balancing between search diversification and intensification. Mohamed Abd Elaziz et al. [27] proposed a paper on task scheduling based on hybrid moth search algorithm using differential equation (MSDE) to minimize the processing time and increase the throughput.

The moth search algorithm(MSA) was developed by taking into consideration the behaviour of moths to move towards light in nature for global optimization, the DE algorithm is used for local search to improve the capability of MSA.

Even though the hybrid techniques found in solving task scheduling in efficient manner, it takes more search time due to presence of more number of tuning parameters. To overcome this limitation, the current research work concentrated on Teaching-Learning Based Optimization (TLBO). This algorithm has less number of tuning parameters during the implementation. Thereby, it reduces the search time to find out the optimal solution.

II. TEACHING-LEARNING- BASED OPTIMIZATION

There are so many population based heuristic algorithms developed for optimization of engineering problems. Teaching-Learning- Based Optimization (TLBO) is one of the recently developed population based heuristic algorithm which was inspired by the real world behaviour of a teacher and learner [28]. It describes how much the influence of a teacher will be on the learners in a class.

TLBO comprises the following two modes of learning.

(i) Through teacher, can be called as teacher phase

(ii) Through interaction with the other learners, can be called as learner phase.

In this TLBO algorithm group of learners are taken as population and different subjects given to them to learn are taken as design variables(nothing but parameters involved in objective function) and the result obtained from learner is considered as the fitness value for the given optimization problem. The learner with the best fitness value will become teacher. This process continues until it reaches end criteria i.e. either maximum number of iterations reached or optimization of problem achieved. The architecture of the TLBO algorithm and its detailed flow process is shown in Fig.2.

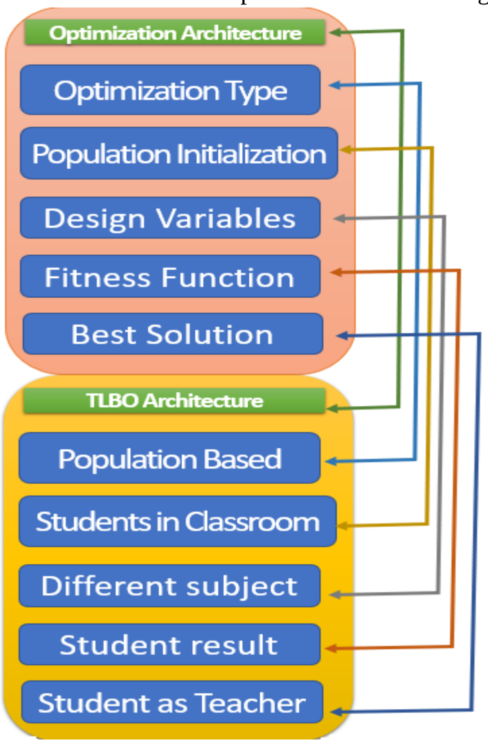

Fig.2 TLBO Architecture

III. PROBLEM STATEMENT AND FITNESS FUNCTION DEVELOPMENT

The objective of this paper, is to decrease the response time and execution time of the tasks while increasing the throughput of the resource by satisfying the QOS demands. The following assumptions are made during the implementation:

(i) Task is executed in a single virtual machine.

(ii) Task is executed on the virtual machine which is functioning.

(iii) The memory required by the total tasks for execution in virtual machines should not exceed the virtual machines extreme memory limit.

(iv) The processing needs of the total tasks toward virtual machines should not exceed the virtual machine extreme processing needs.

Let us consider a task queue contains 'n' tasks which is represented as task set $T_{s} = \{t_1, t_2, \ldots t_n\}$, are to be executed by the processor pool contains 'm' virtual machines represented as Virtual machine set $V_{s} = \{vm_1, vm_2, \ldots vm_m\}$.

A. Important Definitions

Various parameters are to be considered for solving the task scheduling. The details of each parameter are considered in this section.

- *Priority:* The tasks are arranged in a queue with some priority values based on their memory and processing time i.e. tasks are ordered in queue according to their priority values. The task with highest priority is first sent to the corresponding virtual machine for execution. The virtual machines are selected based on the requirements of the tasks.
- *Submission time (subt):* Submission time is defined as the time at which the task enters into the ready queue. The task with less submission time will get the virtual machine first. Generally the submission time starts with '0' for corresponding virtual machine.
- *Burst time (bt):* Burst time is the amount of time required by a task for executing on respective processor. For 'n' number of tasks and 'm' number of virtual machines, each task will have a specified burst time with respect to each individual virtual machine i.e. n*m values will be calculated. We consider burst time is known prior to us or we can calculate the burst time as task length divided by virtual machine speed.
- *Start time (st):* Start time is the time at which the tasks arrived to the virtual machine and ready for execution. Start time can be obtained from the 'gnatt-chart' each of the particular virtual machine. The first task that enters into the VM for execution will have start time zero.
- *Completion time (ct):* Time at which a task completes for its execution in a virtual machine is known as completion time. Completion time is also defined as the time at which a task completes its execution on the virtual machine and takes exit from the system. Completion time of a task can be obtained from 'gnatt chart'.
- *Execution time (ET):* Execution time is the amount of time taken by an individual task for executing completely on corresponding virtual machine. It can also be defined as the difference between task's completion time and start time as represented in equation (1).

$$ET_{(i,j)} = ct_{(i,j)} - st_{(i,j)} \qquad (1)$$

Where 'i' represents the number of tasks varying as $t_1, t_2, \ldots t_n$ 'j' represents the number of virtual machines varying from $vm_1, vm_2, \ldots vm_m$.

- *Turnaround time (tat):* Turn Around time is the total amount of time spent by a task in the system, i.e. the task will be either waiting in the ready queue for getting into the virtual machine or it is executing on the virtual machine.

Turnaround time can also be defined as difference between completion time and submission time as represented in equation (2).

$$tat_{(i,j)} = ct_{(i,j)} - subt_{(i,j)} \quad (2)$$

- *Waiting time (wt):* Waiting time is defined as the time that the task is waiting in the queue looking for its prior task to be executed until its turn comes. Waiting time can be obtained from subtracting burst time from turnaround time as represented in equation (3).

$$wt_{(i,j)} = tat_{(i,j)} - bt_{(i,j)} \quad (3)$$

- *Response time (RT):* The time interval between submission of a task and the first response got to that task is defined as response time as represented in equation (4).

$$RT_{(i,j)} = subt_{(i,j)} - wt_{(i,j)} \quad (4)$$

- *Throughput (TH):* Throughput can be calculated as the tasks that are successfully executed with respect to total execution time as represented in equation (2).

$$TH_{(i,j)} = sucs(N)/T \quad (5)$$

Where, 'sucs(N)' is number of successful tasks

'T' is the total execution time (i.e. total sum of et).

B. Objective Function Formation

The current research work considers the task scheduling problem as multi objective criteria with the following objectives in aspect of resource better utilization:

Objective 1: to reduce the response time, it is given as

$$F \propto \frac{1}{RT_{(i,j)}} \quad (6)$$

Objective 2: is to reduce the execution time, it is given as

$$F \propto \frac{1}{ET_{(i,j)}} \quad (7)$$

Objective 3: is to increase the throughput, it is given as

$$F \propto TH_{(i,j)} \quad (8)$$

Finally, from the equations (6),(7),(8) the fitness equation is given as

$$F = w_1\left(\sum \frac{1}{RT_{(i,j)}}\right) + w_2\left(\sum \frac{1}{ET_{(i,j)}}\right) + w_3\left(\sum TH_{(i,j)}\right) \quad (9)$$

IV. TLBO IMPLEMENTATION

As described in the previous sections, task scheduling requires the optimization as it np-hard problem. This study deals with the implementation of TLBO algorithm to find best resource allocation for the given tasks in the cloud environment.

The detailed flow process of the proposed algorithm is represented in Fig.3.

The algorithm involves the following steps for its implementation:

Step-1: Define the optimization problem which includes-

- Parameter initialization (number of tasks to be executed, define the task priority, number of virtual machines allocated to execute the tasks).
- Identification of Design variables – number of task in a priority que i.e. Ts={t_1, t_2, . . ., t_n} and number of virtual machines allotted for tasks execution i.e. VM={vm_1, vm_2, . . ., vm_n}. various times as described in previous section.

Maximum number of generations-500

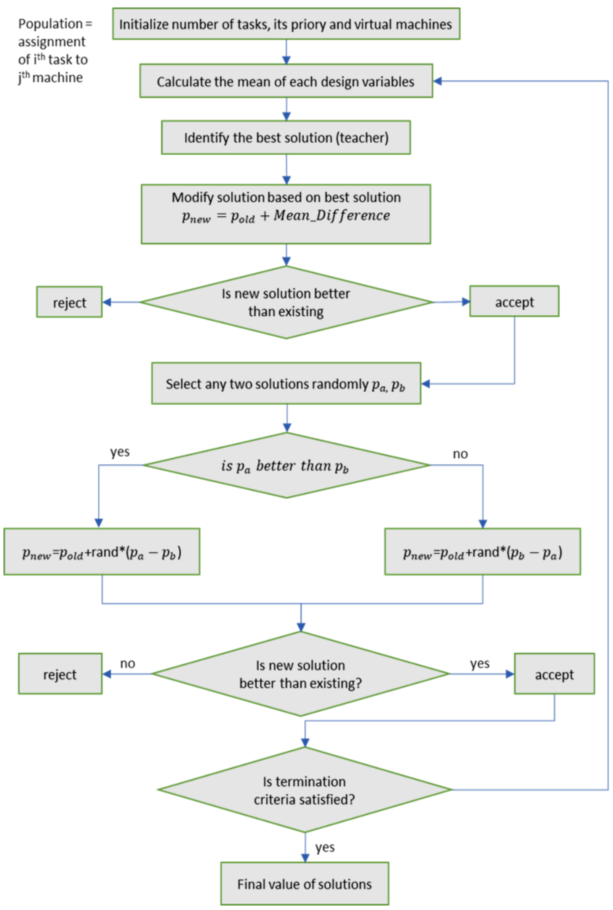

Fig.3 Flow diagram of proposed TLBO algorithm

Step-2: Population generation which includes-

There are so many possible solutions while excitation and also there may be a chance of repeating the same resources allocation for the tasks in a que. In this case the algorithm may be trapped in local minima. To avoid this, only unique sequence of resource allocation are considered for the evaluation. Each unique sequence of resource allocation is termed as an individual and all the number of unique sequence is termed as the population. Table.1 represents the example for unique possible solutions for a four task que with two resources.

Table.1 Population generation in proposed TLBO

Individuals in Population	Queue contains the Task as per the priority			
	t_3	t_2	t_4	t_1
p_1	1	1	1	1
p_2	1	1	1	2
. . .	. . .	. . .	. . .	. . .
p_n	2	2	2	2

Step-3: Teacher phase activation

For each individual in the population and choose a teacher by selecting an individual randomly. Find out the best solution by calculating the mean of the design variables of each individual and treat it as the teacher. Now update the solution as below:

$$p_{new} = p_{old} + Mean\_Difference$$

Retrieval Number B2672078219/19©BEIESP
DOI: 10.35940/ijrteB2672.078219
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Where, p_{new} = new teacher
p_{old} = old teacher
$Mean\_Difference$ = difference of mean of the design variables of new & old teacher

Step-4: Learner phase activation

Select two random population initially, say $p_a \& p_b$.

For i= p_1 to p_n

If fitness (p_a) is better

$$p_{new} = p_{old} + rand * (p_a - p_b)$$

Else

$$p_{new} = p_{old} + rand * (p_b - p_a)$$

End if

End for

Accept current p_{new} as teacher, if it has best fitness value.

Step-5: End criteria

Continue Step-3 & Step-4 until it reaches the maximum number of iterations.

V. RESULTS & DISCUSSION

In order to validate the proposed methodology, the cloud data as illustrated in Table.2 has been considered. The total number of tasks four to eight are considered in the priority que, to be executed in the virtual machines taken from 2 to 4. The weights for the three objectives are assigned with values of '1'.

Table.2 Cloud data for experimentation

Category	Design Variable	Value
Task	Cloudlets range	4-8
	Length (MI)	1000-5000
Virtual Machine	Range	2-4
	Speed (MIPS)	250-1000
	RAM (MB)	256-1024
	Processing cost ($)	0.02-0.2

Cloud environment is created and simulated in MATLAB 2015 software. The program is compiled in a system which has i3-processer with 4-GB RAM and 1-TB hard disk capacity.

In this research study, 3-cases have been considered.

A. Cloud Environment 'Case-1'

This case considers 4-cloudlets in the length range 1000MI-5000MI to be executed in 2-VMs with the speed rang 250MIPS-1000MIPS. On compiling the proposed TLBO algorithm, the best fitness value found with design variables as shown in Fig.4.

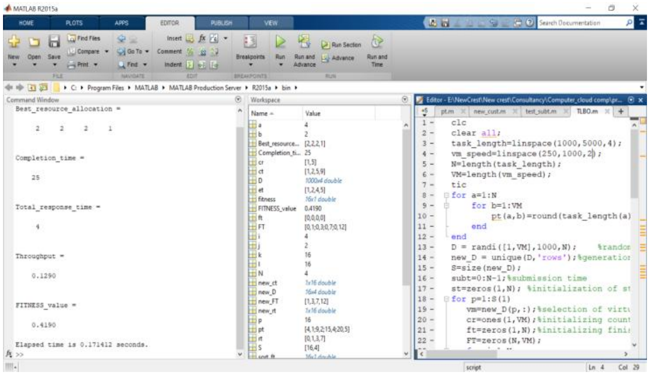

Fig.4 Compilation of TLBO algorithm in MATLAB cloud environment of Case-1

The fitness value variation with the number of TLBO iterations is shown in Fig.5 and the obtained results are tabulated in Table.3.

Table.3 TLBO results for Case-1

Resource allocation to execute the tasks	VM2 → VM2 → VM2 → VM1
Possible number of sequences	16
TLBO iterations to get optimal solution	15
Total completion time during execution of task queue	25
Total response time during execution of task queue	4
Throughput during execution of task queue	0.1290
Optimal Fitness value	0.4190
Elapsed time to compile the program	0.171412 seconds

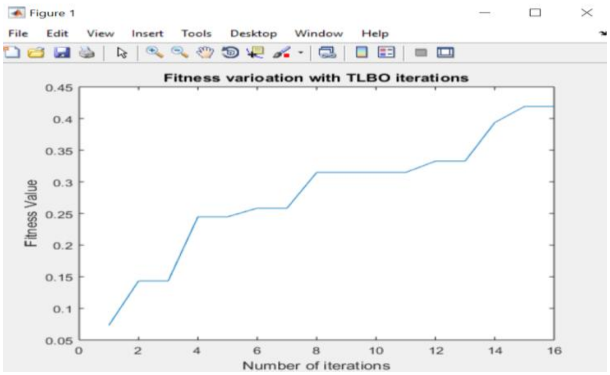

Fig.5 Fitness variation with TLBO iterations of Case-1

B. Cloud Environment 'Case-2'

This case considers 6-cloudlets in the length range 1000MI-5000MI to be executed in 3-VMs with the speed rang 250MIPS-1000MIPS. On compiling the proposed TLBO algorithm, the best fitness value found with design variables as shown in Fig.6.

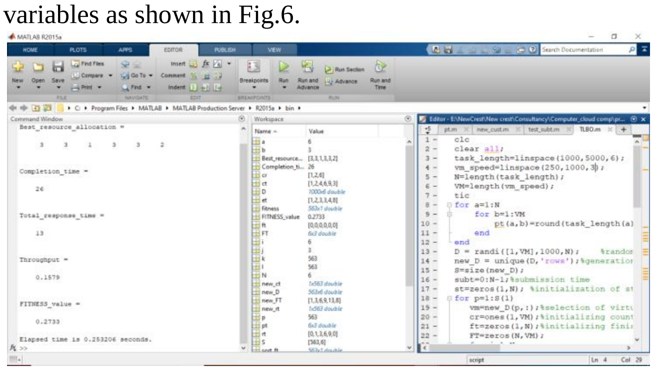

Fig.6 Compilation of TLBO algorithm in MATLAB cloud environment of Case-2

The fitness value variation with the number of TLBO iterations is shown in Fig.7 and the obtained results are tabulated in Table.4.

Table.4 TLBO results for Case-2

Resource allocation to execute the tasks	VM3 → VM3 → VM1 → VM3 → VM3 → VM2
Possible number of sequences	600
TLBO iterations to get optimal solution	513
Total completion time during execution of task queue	26
Total response time during execution of task queue	13
Throughput during execution of task queue	0.1579
Optimal Fitness value	0.2733
Elapsed time to compile the program	0.253206 seconds

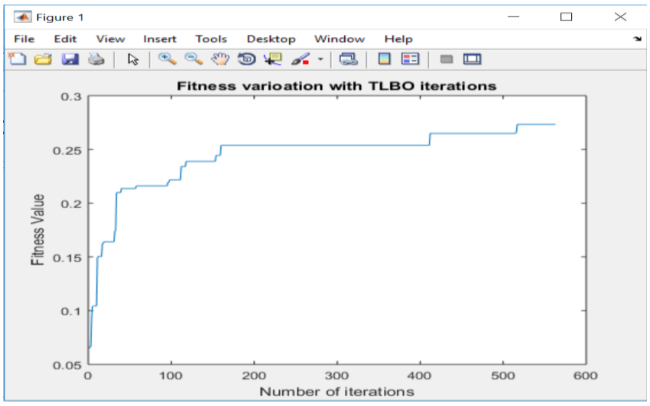

Fig.7 Fitness variation with TLBO iterations of Case-2

C. Cloud Environment 'Case-3'

This case considers 8-cloudlets in the length range 1000MI-5000MI to be executed in 4-VMs with the speed rang 250MIPS-1000MIPS. On compiling the proposed TLBO algorithm, the best fitness value found with design variables as shown in Fig.8.

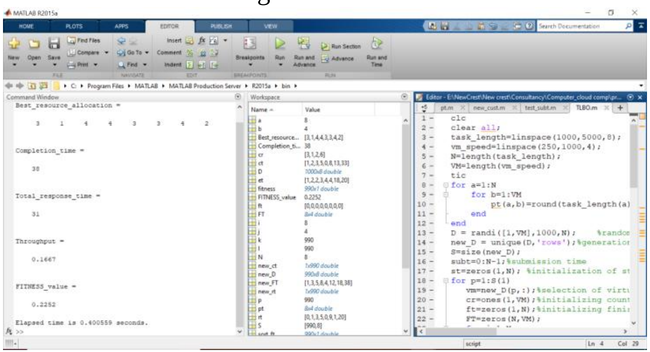

Fig.8 Compilation of TLBO algorithm in MATLAB cloud environment of Case-3

The fitness value variation with the number of TLBO iterations is shown in Fig.9 and the obtained results are tabulated in Table.5.

Table.5 TLBO results for Case-3

Resource allocation to execute the tasks	VM3 → VM1 → VM4 → VM4 → VM3 → VM3 → VM4 → VM2
Possible number of sequences	9500
TLBO iterations to get optimal solution	546
Total completion time during execution of task queue	38
Total response time during execution of task queue	31
Throughput during execution of task que	0.1667
Optimal Fitness value	0.2252
Elapsed time to compile the program	0.400559 seconds

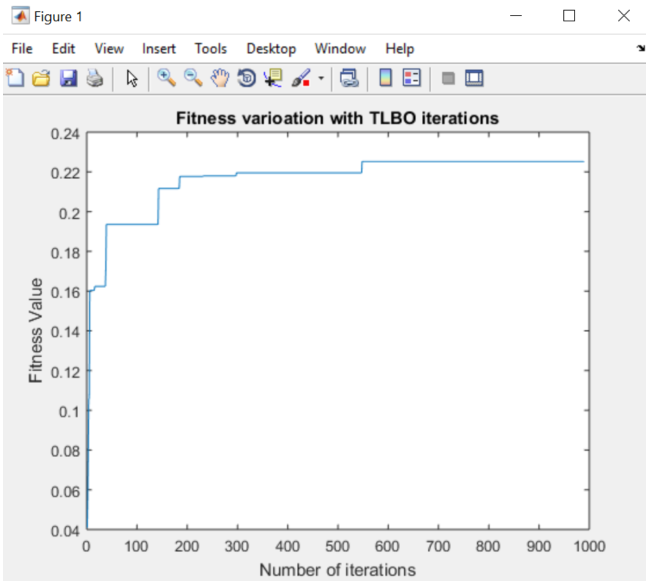

Fig.9 Fitness variation with TLBO iterations of case-3

From the above observations, it is found that the proposed TLBO based algorithm seems efficient when the number task increases with respect to the number of virtual machines

VI. CONCLUSION

In this paper, teaching-learning based optimization has been implemented to the task scheduling in cloud environments for the first time. The proposed algorithm treats the task scheduling as a multi-objective optimization problem. An objective function is introduced in this research work which comprises the minimization of completion time and response time while maximizing the throughput for the tasks execution by the assigned virtual machines. Finally, the proposed technique is validated by presenting the simulation results for these cloud

environments. The results showed that the proposed TLBO algorithm obtains the best solution in less number of iterations even for the cloud environments with huge search space.

REFERENCES

1. G Tilak, S., & Patil, D. (2012). A survey of various scheduling algorithms in cloud environment. International Journal of Engineering Inventions, 1(2), 36-39.

2. Buyya, R., Vecchiola, C., &Selvi, S. T. (2013). Mastering cloud computing: foundations and applications programming. Newnes.
3. Alkayal, E. S., Jennings, N. R., &Abulkhair, M. F. (2016, November). Efficient task scheduling multi-objective particle swarm optimization in cloud computing. In 2016 IEEE 41st Conference on Local Computer Networks Workshops (LCN Workshops) (pp. 17-24). IEEE.
4. Ali, H. G. E. D. H., Saroit, I. A., &Kotb, A. M. (2017). Grouped tasks scheduling algorithm based on QoS in cloud computing network. Egyptian informatics journal, 18(1), 11-19.
5. Wu, X., Deng, M., Zhang, R., Zeng, B., & Zhou, S. (2013). A task scheduling algorithm based on QoS-driven in cloud computing. Procedia Computer Science, 17, 1162-1169.
6. Kong, X., Lin, C., Jiang, Y., Yan, W., & Chu, X. (2011). Efficient dynamic task scheduling in virtualized data centers with fuzzy prediction. Journal of network and Computer Applications, 34(4), 1068-1077.
7. Vinothina, V., Sridaran, R., & Ganapathi, P. (2012). A survey on resource allocation strategies in cloud computing. International Journal of Advanced Computer Science and Applications, 3(6), 97-104.
8. Ge, Y., & Wei, G. (2010, October). GA-based task scheduler for the cloud computing systems. In 2010 International Conference on Web Information Systems and Mining (Vol. 2, pp. 181-186). IEEE.
9. Jang, S. H., Kim, T. Y., Kim, J. K., & Lee, J. S. (2012). The study of genetic algorithm-based task scheduling for cloud computing. International Journal of Control and Automation, 5(4), 157-162.
10. Kaur, S., & Verma, A. (2012). An efficient approach to genetic algorithm for task scheduling in cloud computing environment. International Journal of Information Technology and Computer Science (IJITCS), 4(10), 74.
11. Janmohammadi, P., &Babazade, M. Resource Management in the Cloud Computing Using a Method Based on Ant Colony Optimization. International Academic Journal of Science and Engineering, Vol. 2, No. 6, 2015, pp. 40-54.
12. Guo, Q. (2017, April). Task scheduling based on ant colony optimization in cloud environment. In AIP Conference Proceedings (Vol. 1834, No. 1, p. 040039). AIP Publishing.
13. Zuo, L., Shu, L., Dong, S., Zhu, C., & Hara, T. (2015). A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing. Ieee Access, 3, 2687-2699.
14. Kassarwani, N., Ohri, J., & Singh, A. (2019). Performance analysis of dynamic voltage restorer using improved PSO technique. International Journal of Electronics, 106(2), 212-236.
15. Arumugam, P., Panchapakesan, M., Balraj, S., & Subramanian, R. C. (2019). Reverse Search Strategy Based Optimization Technique to Economic Dispatch Problems with Multiple Fuels. Journal of Electrical Engineering & Technology, 1-7.
16. Ghosh, P., Karmakar, A., Sharma, J., &Phadikar, S. (2019). CS-PSO based Intrusion Detection System in Cloud Environment. In Emerging Technologies in Data Mining and Information Security (pp. 261-269). Springer, Singapore.
17. Juan, W., Fei, L., &Aidong, C. (2012). An Improved PSO based Task Scheduling Algorithm for Cloud Storage System. Advances in Information Sciences and Service Sciences, 4(18), 465-471.
18. Krishnasamy, K. (2013). Task Scheduling Algorithm Based On Hybrid Particle Swarm Optimization In Cloud Computing Environment. Journal of Theoretical & Applied Information Technology, 55(1).
19. Awad, A. I., El-Hefnawy, N. A., &Abdel_kader, H. M. (2015). Enhanced particle swarm optimization for task scheduling in cloud computing environments. Procedia Computer Science, 65, 920-929.
20. Jena, R. K. (2015). Multi objective task scheduling in cloud environment using nested PSO framework. Procedia Computer Science, 57, 1219-1227.
21. Alkayal, E. S., Jennings, N. R., &Abulkhair, M. F. (2016, November). Efficient task scheduling multi-objective particle swarm optimization in cloud computing. In 2016 IEEE 41st Conference on Local Computer Networks Workshops (LCN Workshops) (pp. 17-24). IEEE.
22. Shu, W., Wang, W., & Wang, Y. (2014). A novel energy-efficient resource allocation algorithm based on immune clonal optimization for green cloud computing. EURASIP Journal on Wireless Communications and Networking, 2014(1), 64.
23. Jena, R. K. (2017). Energy efficient task scheduling in cloud environment. Energy Procedia, 141, 222-227.
24. Dordaie, N., &Navimipour, N. J. (2017). A hybrid particle swarm optimization and hill climbing algorithm for task scheduling in the cloud environments. ICT Express.
25. Liu, C. Y., Zou, C. M., & Wu, P. (2014, November). A task scheduling algorithm based on genetic algorithm and ant colony optimization in cloud computing. In 2014 13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science (pp. 68-72). IEEE.
26. Laili, Y., Zhang, L., & Tao, F. (2011, December). Energy adaptive immune genetic algorithm for collaborative design task scheduling in cloud manufacturing system. In 2011 IEEE International Conference on Industrial Engineering and Engineering Management (pp. 1912-1916). IEEE.
27. Elaziz, M. A., Xiong, S., Jayasena, K. P. N., & Li, L. (2019). Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution. Knowledge-Based Systems, 169, 39-52.
28. Rao, R. V., Savsani, V. J., &Vakharia, D. P. (2011). Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Computer-Aided Design, 43(3), 303-315.

AUTHORS PROFILE

Ramakrishna Goddu Pursuing Ph.D from department of Computer Science and Engineering Krishna University, Machilipatnam, Ap, India. M.Tech from JNTU Kakinada, M.C.A from Andhra University Visakhapatanam, B.Sc (Computer Science) from Andhra University Visakhapatanam. Published Five international journal papers Life member to Form for Intellectual academicians and Researchers, Regd.No. 302/2012 under ACT 35 of 2001, with life Membership No. FIAR13032.

Dr Kiran Kumar Reddi, working as a Asst.Professor in the department of Computer Science and Engineering, Krishna University, Machilipatnam, Ap, India. Completed PhD (Computer Science and Engineering) from Acharya Nagarjuna University, India, M.Tech (Computer Science and Engineering) from JNTU, Kakinada, India, MCA (Master of Computer Applications) from Andhra University, India and awarded **twelve** Ph.Ds, published over **Nine** Books, over **Ninety** National and International papers were published and fifteen conference proceedings are presented. Life member in Indian Society for Technical Education Life member in The Institution of Electronics and Telecommunication Engineers, Member in Computer Society of India, Member in International Association of Computer Science and Information Technology, Member in International Association of Engineers.