International Journal of Future Generation Communication and Networking
Vol. 13, No. 3, (2020), pp. 1799 –1808

Time and Cost Aware Meta Task Scheduling Algorithm for Multicloud Environment (TCAMTSA)

Hubert Shanthan B. J.[1], Arockiam L[2], Cecil Donald A[3], Dalvin Vinoth Kumar A[4], Stephen R[5]

[1,3,4,5]*Assistant Professor, Kristu Jayanti College (Autonomous), Bengaluru, India*
[2]*Associate Professor, St. Joseph's College (Autonomous), Tiruchirappalli, India*

Abstract

Multi cloud is an environment which comprised of one or more cloud service providers provide a unanimous service to one or more consumers. This cloud can be utilized for private or public cloud and it is mainly devised to solve vendor lock in problems in the cloud environment. Scheduling is a process of assigning resources and allocate to the user. Scheduling is one of the major problems in the multi cloud environment. Meta task scheduling is suitable for independent tasks and it works based on batch scheduling mechanism. Particle Swarm Optimization (PSO) is meta heuristics-based technique used to solve scheduling problem. Each particle in PSO represents the solution for the problem. The additive weighted function is used as fitness function to solve the scheduling problem. Chaotic inertia weight is used to increase the convergence rate of the optimization. The proposed algorithm uses PSO with chaotic inertia weight optimization mechanism to reduce makespan and cost. This algorithm outperforms the existing simple PSO.

Keywords: *Optimization, Multicloud, Scheduling, Particle Swarm Optimization (PSO), meta heuristic.*

1. Introduction

Multicloud is an emerging paradigm which provides high performance computing services to the user in the form of hardware, software and APIs (Application Programming Interface) [1]. Scheduling plays a significant role in improving the viability, scalability and performance of the multi cloud systems. There are two different types of scheduling algorithms namely dependent and independent(meta). The dependent scheduling is represented in the form of DAG (Direct acyclic graph) whereas independent scheduling is represented in the form of ETC (Expected Time to Compute) matrix [2] [3]. Scheduling is one of the major problems in the multi cloud environment. The solutions obtained from the existing heuristic-based approaches are not sufficient to solve the task scheduling problem. So, there is a need for novel Meta heuristic-based solutions to solve the task scheduling problem in the multi cloud systems. Swarm Intelligence is a popular Meta heuristic based optimized search technique used to solve NP Complete combinatorial optimization problem [4][5]. Particle Swarm Optimization (PSO) algorithm is a population-based Meta heuristic algorithm works under Swarm Intelligence [6]. The possible solutions in the PSO are represented in the form of particles. The movement and speed of each particle is determined by the fitness value and velocity. To avoid pre-mature convergence of optimization algorithms, the inertia weight is used along with velocity of PSO to obtain a feasible solution. Personal best value is determined by the best solution of the individual particle [7]. The best solution of entire individual particle termed as Global best. In all iterations, a particle moves towards the global optimal solution based on the experience gained from the previous personal best and global best [8][9]. The PSO combines local search and global search methods to balance the exploration and exploitation in the multi-dimensional search space [10].

ISSN: 2233-7857 IJFGCN

This paper comprised of five sections namely, the first section consists of introduction and basic terminology, second section comprised of existing literary works, third section contains the methodology of the proposed algorithm with explanation, four section contains results and discussions and finally five section has conclusion.

2. Related Works

Thamarai et al., [11] proposed a fitness function was designed to minimize both execution time and the computation cost of VM resources. The deadline was also added along with the fitness function. The premature convergence rate of this algorithm was extremely high. So, this approach failed to reach global optimal solution.

Jemina et al., [12] enhanced the PSO based task scheduling algorithm to reduce makespan and to increase the resource utilization rate of the server. The local best and global best values were computed based on the position and the velocity of the particle. The fitness function values were used to determine the speed and movement of the particle. The convergence rate of the algorithm was high when it was compared with the standard PSO. So, this algorithm attained the global optimum solution with minimum time. The cost parameter was not considered in this approach.

Begom et al., [13] proffered a Pareto optimal based PSO algorithm for the cloud systems. The algorithm was designed to solve the bi objective optimization problem. The primary aim of this algorithm was to reduce the makespan and execution cost of the VM resources in the cloud. The weighted sum approach was used to convert multi objective optimization problem into single objective. Integer PSO was a small variant of the simple PSO and it was used in this algorithm.

Zhang et al., [14] suggested multi task scheduling algorithm based on the self-adaptive inertia PSO. The self-adaptive inertia weight was adopted to adjust the convergence rate of the optimum solution. The disruption and chaos operator were the two operators used in this algorithm. The disruption operator was applied to prevent the loss in the population diversity. To prevent the optimum solution stuck in the local search, the chaos operators were applied.

Netjinda et al., [15] proposed a PSO scheme-based task scheduling algorithm for the cloud. The PSO was designed to optimize the cost for provisioned VMs resources. The positions of each particle were updated frequently until it would reach near the optimal solution. The cost was computed based on the parameters such as instance purchased on-demand and reserved, per-hour cost of on-demand and reserved instances.

Kaur et al., [16] proposed a task scheduling algorithm which was combined the PSO and SA (Simulated Annealing) to increase the resource utilization of the VMs. The goal of this algorithm was to increase the rate of the convergence speed of the optimum solution. The PSO was used for the faster search in the feasible optimal solution in the multi-dimensional space. The SA was used to avoid jump from local optimal solution to global optimum solution. The N initial solutions were selected randomly and the midpoint of the present position and the target position were found in the first step.

Medhat et al., [17] proposed task-based scheduling algorithm based on ant colony optimization technique. The random optimized search approaches were used for the virtual machine allocation in the cloud. The probabilistic transition rules were used to update the pheromone value of each ant. The constraint satisfaction method was used for the scheduling process. The memory was also used for the storage of the allocated tasks in the available virtual machines.

Narendran et al., [18] highlighted an improvised ACO algorithm for cloud systems. It was suitable for IaaS service model and used to reduce the degree of imbalance among the VMs in the cloud. The Multi objective optimization task scheduling strategies were used in

ISSN: 2233-7857 IJFGCN

the algorithm. The decisions for the task-VM allocation were made in this algorithm based on the probability of deposition of the pheromone of the ant. The pheromone values were computed based on the processing speed of the VM. The pheromone values were changed dynamically based on the user requests.

3. Methodology

The algorithm is more viable for the static independent task-based scheduling in multi cloud computing environment. The flow diagram of TCAMTSA (Time and Cost Aware Meta Task Scheduling Algorithm) is shown in the Figure 1.

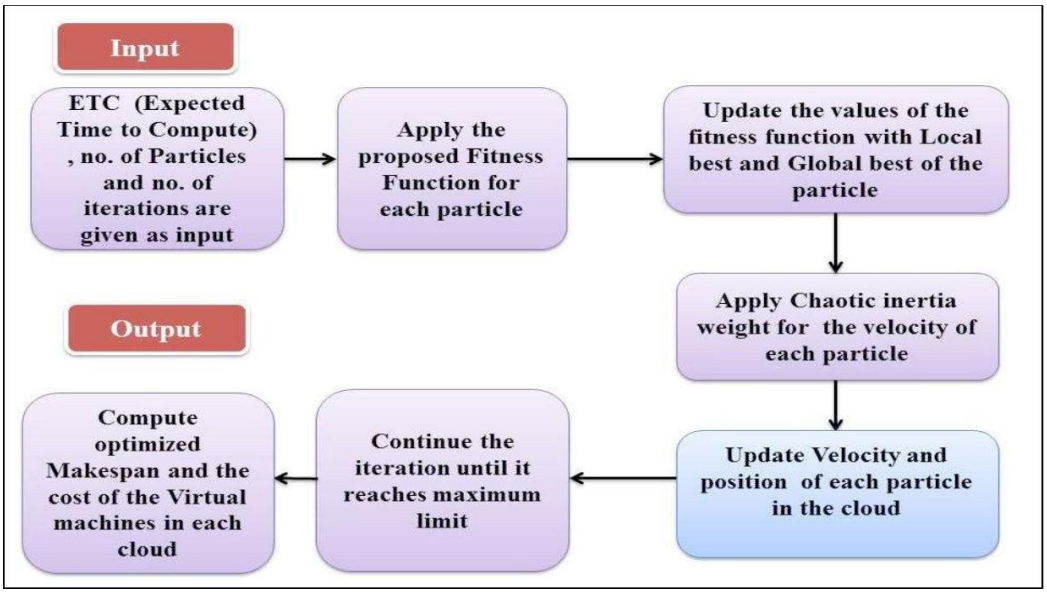

Figure 1. Methodology of TCAMTSA Algorithm for Multicloud

The ETC (Expected Time to Compute) matrix, total number of particles and the number of iterations is given as the input in this algorithm. The algorithm works based on the particle swarm optimization technique. A group of particles are used to represent the solution. The fitness function is designed to reduce the maximum completion time and the execution cost of the VMs. The proposed fitness function is designed to solve Multi Objective Optimization Problem in PSO. The Weighted Sum approach is used along with the fitness function to solve the problem. The approach is used to convert multi objective optimization problem into single solution with the help of the weights that are assigned to the function.

Every particle has two components namely velocity and position. The velocity is used to change the searching direction of the particle. The current and the previous positions of each particle are stored for entire iteration. The local best value is computed based on the personal best position and it also keeps track on the minimum fitness value obtained in each iteration. The global best value is computed based on the overall best position and it stores the minimum fitness value in the entire population. The mechanism of the proposed TCAMTSA works until it reaches the maximum number of iterations.

PSO is a population-based Meta heuristic technique. It is well suitable for continuous and combinatorial optimization problem. The task scheduling is a combinatorial optimization problem. PSO strategy is well suitable to solve the problem. The fitness function of the PSO requires basic mathematical and logical function and it is simple and easy to implement. The existing GA and Evolutionary programing techniques are difficult to implement. The Time complexity is high when it is compared to PSO. The PSO has larger searching ability and it is more efficient for computation of larger numbers. It can

ISSN: 2233-7857 IJFGCN

handle objective function with stochastic nature. The procedure for TCAMTSA are elaborated as follows:

3.1. Generation of Initial Conditions of Each particle

The initial condition of every particle is generated randomly based on the specified constraint. Each particle represents the solution for task scheduling problem.

3.2. Evaluation of Each Particle

Each particle X_k is evaluated using the fitness function of the problem to minimize makespan and the execution cost is given in the (1).

$$f(X_k) = (\theta - 1) \times f(x_i) + \theta \times f(x_j) \quad \ldots (1)$$

where, θ denotes the relative weight or preference of one objective over the other and it has range [0, 1]. When θ = 0, the optimization problem minimizes the makespan and if the θ = 1, it minimizes the cost. $f(x_i)$ denotes the makespan value and $f(x_j)$ represents the execution cost of the VMs in the multi cloud.

$$f(x_i) = \min(Mksp(M)) \quad \ldots (2)$$

Subject to $M \in F_r$

where, M is used to search and schedule the task-VM pair that minimizes the makespan and F_r is a feasible region in the objective space.

$$Mksp(M) = \max(CTm_{ij}) \times y_{ij} \quad \ldots (3)$$

where CT_{ij} represents the completion time of the tasks, y_{ij} represents the decision variable and also denotes that the task is assigned to the VMs.

$$CTm_{ij} = \sum_{i=1}^{m} \sum_{j=1}^{n} ETC_{ij} + rt_j \quad \ldots (4)$$

where, $i < 0 < m$, $j < 0 < n$ is used to represents the set of tasks, j denotes the set of VMs in the clouds, ETC_{ij} denotes the ETC matrix, m denotes the independent tasks and n denotes the virtual machines and rt_j denotes the ready time of the j virtual machines.

The tasks are scheduled to the VMs which are subject to the following constraints.

$$m \leq n \quad \ldots (5)$$

$$y_{ij} = \begin{cases} 1, & if\ task\ is\ scheduled\ to\ VM_{ij} \\ 0, & otherwise \end{cases} \quad \ldots (6)$$

where, y_{ij} represents the decision variable, m tasks and n VMs resources.

$$f(x_j) = \min(CsT(Cs)) \quad \ldots (7)$$

Subject to $C_s \in F_r$

$$CsT(Cs) = \sum_{j=1}^{n} ETC \times cost(T(VM_j) \times y_{ij}) \quad \ldots (8)$$

where, $CsT(C_s))$ is used to compute the execution cost of Task-VM pair in each cloud, C_s is used to search the task-VM pair with minimum cost in each cloud.

3.3. Modification of Each Particle

The position for each particle is modified according to the following equation (9).

$$pt_i(t) = vl_i(t) + pt_i(t-1) \quad \ldots (9)$$

where,

ISSN: 2233-7857 IJFGCN

$$vl_i(t) = w(t).vl_i(t-1) + cst_1 rd_1(pt_{Lbest,i} - pt_i(t-1)) + cst_2 rd_2(pt_{Gvbest} - pt_i(t-1)) \quad \ldots (10)$$

where,

$vl(t)$ is the velocity of particle i at iteration t

$w(t)$ is the inertia weight and it is expressed in (10)

cst_1, cst_2 are the acceleration factors which are taken 0.12 and 1.2 respectively

rd_1, rd_2 are the random numbers between 0 and 1

$pt_{lbest,i}$ is the local best of particle i

pt_{Gvbest} is the global best of the group of particles in solution space

$pt_i(t)$ is the position of particle i at iteration t

$pt_i(t-1)$ is the previous position of particle in the iteration

Inertia weight is a strategy used to balance the exploration and exploitation process in the PSO. Inertia weight is introduced to increase the convergence rate of the optimal solution. The velocity of each particle is controlled by the inertia weight. The chaotic random inertia strategy is used in the PSO to get the better global optimal solution. Chaotic random inertia weight strategy is used to increase the accuracy and to avoid pre mature convergence of the optimal solution. This strategy enables the global search ability in the PSO.

The chaotic inertia weight for PSO is expressed in the equation (11)

$$\omega = (\omega 1 - \omega 2) \times \frac{MaxIter - iter}{MaxIteration} + \omega 2 \times Z \quad \ldots (11)$$

where, ω=0.9, ω2=0.4, Z is the random number generated for each iteration ∈ [0,1].

4. Algorithm of TCAMTSA

Input: ETC Matrix, Num_Par, CsT
Process: Chaotic Inertia Weight strategy is used to obtain optimal solution
Output: Best particle with minimum makespan and Cost

1. **BEGIN**
2. Initialization $Pos\_Par_i = 0$, $Par\_Vel_i = 0$ *// Initialization stage//*
3. While Num_Iter = 500 *// Termination Criteria //*
4. {
5. While Num_Tk ≠ 0 && Num_VM ≠ 0 do // *Total Number of tasks and VM* //
6. {
7. While Num_Par = *0 // Total Number of Particles//*
8. {
9. for 1 to Num_Par
10. {
11. $f(A_i(t)) = Min(Max(CT(ETC_{ij})) + Min(CST)$; *// Fitness Function//*
12. *Compute the position of the particle based on the equation 5*
13. *Compute the velocity of the particle based on the equation 6*
14. $CInW = 0.9 - 0.4 \times 500 - Itr_{cur} / Itr_{max} + 0.9 \times 0.5$; *// Inertia Weight//*
15. If $F(A_i(t)) < F(PL_{best})$ *// Performance of the individual //*
16. {
17. $F(PL_{best}) = F(A_i(t))$;
18. $PL_{best} = A_i(t)$;
19. }
20. If $F(A_i(t)) < F(PG_{best})$ *// Performance of the particle//*

ISSN: 2233-7857 IJFGCN

```
21.             {
22.                 F(PG_best) = F(A_i(t));
23.                 PG_best = A_i(t);
24.             }
25.             Update the position of each particle and the velocity of each particle
26.             Move each particle position
27.             }
28.         }
29.     }
30. }
31. END
```

The ETC represents the Expected Time to compute matrix [19], Num_Par symbolizes Number of particles and CsT represents the execution cost of VMs in the multi cloud. The step 2 indicates the initialization stage of the algorithm. The Pos_Par_i denotes the position of the particle. The Par_vel_i represents the velocity of the particle. In the initialization stage, the position and the velocity of the particle are randomly initialized within the feasible space. Each particle represents an initial task scheduling solution and each solution represents the task to VM resource mapping. In the step 3, the maximum numbers of iterations are taken as termination criteria of the algorithm. The steps (4-5) check whether the total number of tasks and the VMs resources are empty or not.

The steps (6-10) checks whether the total number of particles are empty or not and iteration phase begins from this step. The fitness function is used in the step 11 and it is applied for each particle in the entire population. The position and the velocity of each particle are computed in the steps 12 and 13.

The chaotic inertia weight is also applied to update the velocity of the particle and it is used in the step 14. The steps 15-19 clearly denote to compute the personal best position of the particle. These steps are used for the local search of the particle.

The comparison is made between the performance of each individual and the personal best position of the particle. The steps (20-24) are used to compute the global best.

5. Results and Discussions

The TCAMTSA is tested using CloudSim [20] as the simulation tool. An initial random population is generated. The algorithm is executed for 400 iterations with 2 clouds. A task set of 10, 20, 50, 100 and 400 with four virtual machines are considered.

Table 1: Comparative Analysis of Makespan values with existing algorithms

Number of Tasks	Min-Min (Milliseconds)	Max-Min (Milliseconds)	PSO (Milliseconds)	TCAMTSA (Milliseconds)
10	32.68	29.84	27.34	26.12
20	62.45	56.45	49.89	47.22
50	98.45	95.12	89.45	87.12
100	134.56	128.78	120.56	118.56
400	545.41	512.69	507.39	500.12

The Table 1 depicts the comparative analysis of makespan values with the existing algorithms such Min-Min, Max-Min and simple PSO. The results show that the proposed TCAMTSA outperforms the existing algorithms.

ISSN: 2233-7857 IJFGCN

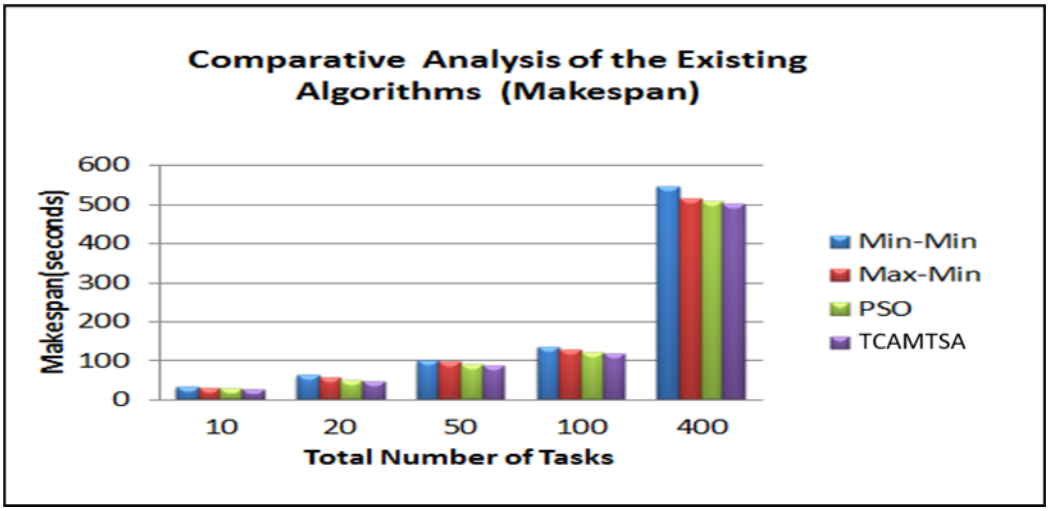

Figure 2. Performance (makespan) of TCAMTSA with existing algorithms

The figure 2 clearly depicts the graphical representation of the makespan values which are tabulated in the table 1. The X axis denotes the makespan in terms of milliseconds and Y axis represents the total number of tasks.

Table 2: Cloud Resource Utilization Ratio

Number of Tasks	Min-Min (%)	Max-Min (%)	PSO (%)	TCAMTSA (%)
10	78.23	79.01	80.12	81.58
20	79.05	81.25	81.98	82.98
50	80.52	78.89	82.25	83.54
100	81.02	81.45	83.12	83.98
400	82.45	82.92	83.78	84.02

The Table 2 depicts the values of cloud resource utilization for the existing min-min, max-min, PSO and the TCAMTSA for the tasks with different ranges 10, 20, 50, 100 and 400. The proposed TCAMTSA shows the increased cloud resource utilization when it is compared with the existing algorithms.

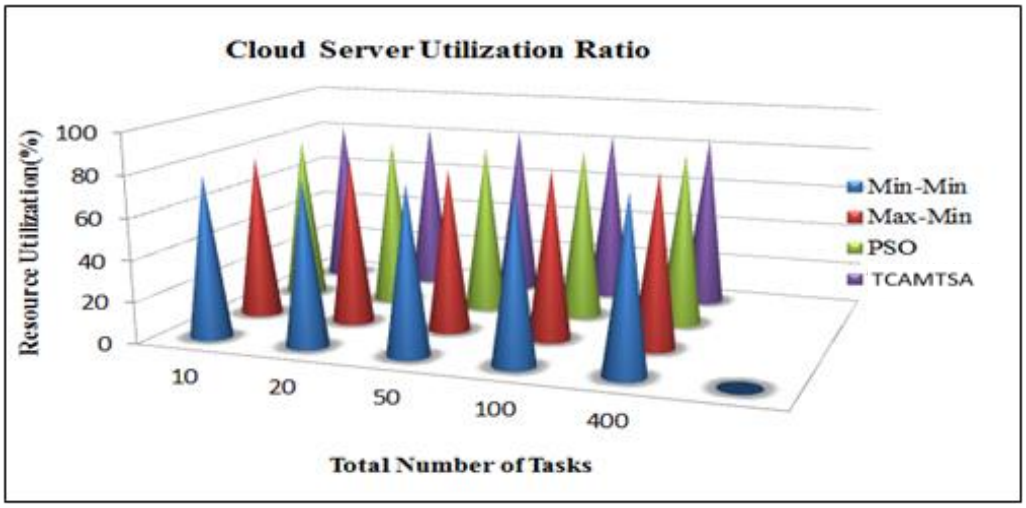

Figure 3. Cloud Resource Utilization Ratio of Proposed TCAMTSA

The Figure 3 clearly depicts the graphical representation of the proposed algorithms with the existing algorithms. The results clearly show that the proposed TCAMTSA algorithm

ISSN: 2233-7857 IJFGCN

increases the cloud resource utilization ratio when it is compared with the existing algorithms.

Table 3: Execution Cost for the existing and proposed TCAMTSA

Number of Tasks	Min-Min (Dollars)	Max-Min (Dollars)	PSO (Dollars)	TCAMTSA (Dollars)
10	245.67	198.89	195.45	190.16
20	300.12	265.56	220.15	200.54
50	412.56	320.77	289.77	250.58
100	578.96	498.38	455.12	400.12
400	985.12	790.89	678.23	600.12

The Table 3 shows the tabulated results of the execution cost of the virtual machines for the existing min-min, max-min, PSO and the proposed TCAMTSA algorithm. The simulation results prove that the proposed TCAMTSA algorithm reduces the execution cost when compared with the existing conventional min-min algorithm.

Table 3 considers execution cost of VM values in the multi cloud environment with four virtual machines and two clouds for the tasks with different ranges of 10, 20, 50, 100 and 400.

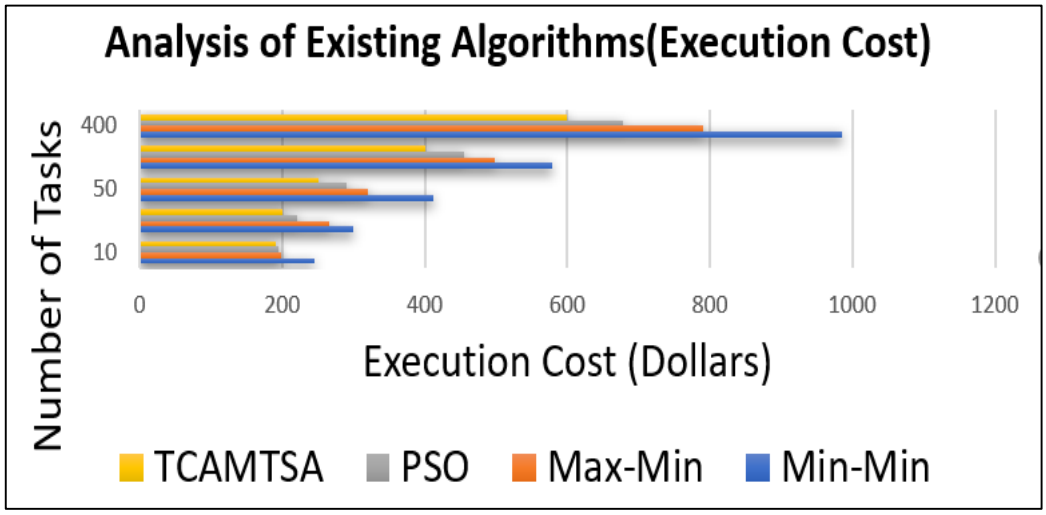

Figure 4. Comparison of Existing Algorithms (Execution Cost)

The Figure 4 clearly depicts the comparative analysis of the execution cost of the existing algorithms with the proposed TCAMTSA algorithms. The X axis denotes the total number of tasks and Y axis denotes the execution cost of VMs in terms of dollars. The results prove that the TCAMTSA requires less cost when it is compared with the existing Min-Min, Max-min and PSO algorithms.

The normalized values of all iterations and the convergence characteristics are shown in the figure 5. The proposed TCMTSA converges at 152th iteration, whereas the simple PSO converges only at 200th iteration. The convergence rate of the proposed TCAMTSA algorithm is faster when compared with existing simple PSO.

ISSN: 2233-7857 IJFGCN

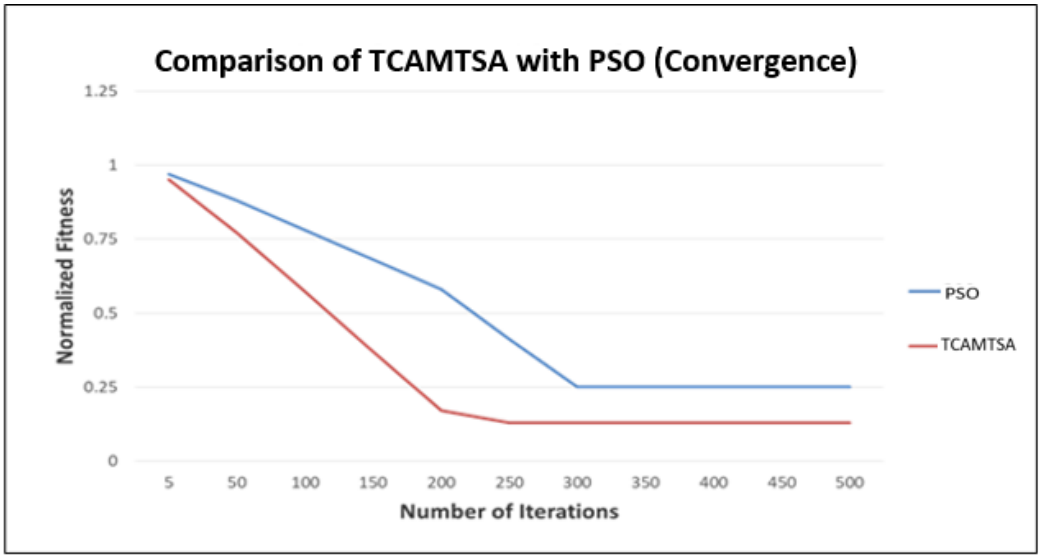

Figure 5. Comparison of Convergence Rate with the Existing Algorithm

The convergence rate is computed when the algorithm reaches the global optimum solution.

6. Conclusion

Time and Cost Aware Meta Task Scheduling Algorithm (TCMTSA) algorithm has been proposed for the independent tasks (i.e. Meta) in the Multicloud environment. The proposed algorithm minimizes time, cost and increases the cloud server resource utilization. There is a need for effective solution for this multi objective optimization problem. The simple PSO is suitable for the single objective optimization problems. The TCAMTSA is proposed to design a multi objective optimized solution for this problem. The fitness function is designed based on the additive weighted method and Pareto optimal solution for the multi objective optimization problem. The Simple Additive Weighted method is efficient method and it requires less time when compared with the existing methods. This method is used for multiple attribute decision making situations. So, this method is more suitable for multi cloud environment. The simple PSO uses linear weight. The convergence rate in the existing approach is relatively low. The chaotic inertia weight is used in the algorithm to increase the rate of convergence of the optimum solution. The parameters considered are makespan, cost and resource utilization. The proposed TCAMTSA algorithm achieves better resource utilization and minimizes makespan and cost compared to the existing algorithm. In future this algorithm can be modified for dynamic environment with other optimization methods.

References

[1] Singh Shyam and Wagle, "Cloud Service Optimization Method for Multi-Cloud Brokering", IEEE Transactions on Cloud Computing, **(2017)**, pp.1-13.
[2] Gartner, **(2020)**, "Multicloud Definition https://www.gartner.com/doc/3892385/ Multi cloud definition, accessed online 3.03.2020.
[3] Mohammed Ahsan Ullah, "Design and Implementation of a framework for Multi-Cloud Service Broker"**,** PhD. Thesis, Ryerson University, Canada, **(2015)**.
[4] Panda Gupta and Jana, "Allocation-Aware Task Scheduling for Heterogeneous Multi-Cloud Systems", Information System Frontiers, Springer Journal, **(2017),** pp.1-17.
[5] Hubert Shanthan B.J. and Arockiam L., "Cost-Based Meta-Task Scheduling Algorithm for MultiCloud Computing (CBMTSA)", In: Ranganathan G., Chen J., Rocha Á. (eds) Inventive Communication and Computational Technologies. Lecture Notes in Networks and Systems, vol 89. Springer, **(2020)**.

ISSN: 2233-7857 IJFGCN

[6] BJ. Hubert Shanthan, Dr. L. Arockiam, "Rate aware Meta task Scheduling Algorithm for multi cloud computing (RAMTSA)", Journal of Physics: Conference Series, No.1142**,** doi:10.1088/1742-6596/1142/1/012001, **(2018)**, pp. 1-10.

[7] BJ. Hubert Shanthan, Dr. L. Arockiam, "Spell Aware Meta Task Scheduling Algorithm (SAMTSA) for Multicloud", International Journal of Scientific Research in Computer Science Applications and Management Studies, Vol. 7, No. 4, **(2018)**, pp. 1-7.

[8] BJ. Hubert Shanthan, Dr. L. Arockiam, "LAHUBMAX –Priority Based Meta Task Scheduling Algorithm in multicloud", International Journal of Computer Sciences and Engineering (IJCSE), Vol. 6, No. 10, ISSN. 2347-2693, **(2018)**, pp.650-655.

[9] BJ. Hubert Shanthan, A. Dalvin Vinoth Kumar, Er. Karthigai Priya Govindarajan, Dr. L. Arockiam, "Scheduling for Internet of Things Applications on Cloud: A Review", Imperial Journal of Interdisciplinary Research (IJIR) Vol. 3, No. 1, **(2017)**, pp. 12-18.

[10] BJ. Hubert Shanthan, L. Arockiam, A. Cecil Donald, A. Dalvin Vinoth Kumar and R. Stephen, "Priority Intensed Meta Task Scheduling Algorithm for Multi Cloud Environment (PIMTSA)", Journal of Physics: Conference Series, No 1427, doi: 10.1088/1742-6596/1427/1/012007, **(2020)**, pp. 1-8.

[11] Thamari Somasundaram and Govindarajan, "CLOUDRB: A Framework for Scheduling and Managing High-Performance Computing (HPC) Applications in Science Cloud", Future Generation Computer Systems, Elsevier, Vol. 34, No. 12, (**2012)**, pp. 47-65.

[12] Jemina Priyadarsini and Arockiam, "An Improved Particle Swarm Optimization Algorithm for Meta Task Scheduling in Cloud Environment", International Journal of Science and Technology (IJCST), Vol. 3, No. 4, **(2015)**, pp. 108-112.

[13] Beegom and Rajasree, "A Particle Swarm Optimization Based Pareto Optimal Task Scheduling in Cloud Computing", International Conference in Swarm Intelligence, Springer, **(2014)**, pp. 1-20.

[14] Zhang, Zuo and Tan, "Self-adaptive Learning PSO Based Deadline Constrained Task Scheduling for Hybrid IaaS Cloud", IEEE Transactions. Automation. Science Engineering., Vol. 11, No. 2, **(2014)**, pp. 564–573.

[15] Netjinda, Achalakul and Sirinaovakul, "Cloud Provisioning for Workflow Application with Deadline using Discrete PSO", ECTI Trans. Computer Inf. Technol, Vol. 7, No. 4, **(2014)**, pp. 43-51.

[16] Kaur and Sharma, "Optimized Utilization of Resources Using Improved Particle Swarm Optimization Based Task Scheduling Algorithms in Cloud Computing", International Journal of Advance Engineering, Vol. 4, No. 1, **(2014)**, pp. 110–115.

[17] Shanthan, B. H., & Arockiam, L., "Resource Based Load Balanced Min-Min Algorithm (RBLMM) for Static Meta Task Scheduling in Cloud", In International conference on advances in computer science and technology. Int J Eng Technol Spec, **(2018)**, pp. 1-8.

[18] Narendran Reddy and Kumar, "Modified Ant Colony Optimization Algorithm for Task Scheduling in Cloud Computing Systems", International Smart Intelligent Computing and Applications, Springer, **(2019)**, pp. 357-365.

[19] Kokilavani Thangavel and George Amalarethinam, "EMGEN – A Tool to Create ETC matrix with Memory Characteristics for Meta Task Scheduling in Grid Environment", International Journal of Computer Science and Applications, Vol. 2, No. 3, **(2013)**, pp.1-8.

[20] Buyya, James Broberg and Andrzej Goscinski, "Cloud Computing Principles and Paradigms", Wiley and Sons Pub Pvt, **(2013)**, pp. 12-55.

ISSN: 2233-7857 IJFGCN