Deadline Constrained Workflow Scheduling Optimization by Initial Seeding with ANT Colony Optimization

Neel Sinha Vishesh Srivastav Waquar Ahmad

ABSTRACT

Cloud computing is a new model of service provisioning in distributed systems. It encourages researchers to investigate its benefits and drawbacks on executing scientific applications such as workflows. One of the most challenging problems in workflow scheduling in cloud environment is its quality of service, which minimizes the cost of computation of workflows. In this paper, we use the Predicted Earliest finish time (PEFT) for initial seeding to Ant Colony optimization technique (ACO). As we know ACO is a very powerful technique appropriate for optimization.. The increasing complexity of the workflow applications is forcing researchers to explore hybrid approaches to solve the workflow scheduling problem. In this paper we proposed PEFT with ACO algorithm which reduces the initialization complexity and converge ACO algorithm.

Keywords

PEFT, Workflow, QOS, Cost

1. INTRODUCTION

Cloud computing technology is a versatile platform for providing various services related to hardware, software and for developing new applications. The cloud service provider is responsible for providing these services in a seamless manner [1]. The shared pool of resources (processing power, memory, storage, network etc.) is allocated in an optimized way to ensure maximum resource utilization with minimum makespantime[5].

A cloud datacenter is a collection heterogeneous resources in which both static and dynamic information is kept[2]. Static information refers to information related to total available datacenter storage, memory, processor cycles, memory/storage allocated to VMs. Such information remains constant throughout the cloud datacenter [4]. Whereas, dynamic information includes load allocated to datacenter physical machines, number of tasks/threads executing at a particular instance, tasks running states, number of CPU cycles utilized for particular task, status of tasks in execution etc.[7]. This dynamic information is updated at regular intervals in real time so that the dynamic user requests can be handled in minimum response time. So, both static and dynamic information is used while scheduling tasks of an application workflow [5].

Workflow tasks requests received in cloud datacenters vary in terms of communication and computation. Some requests demand more computation than communication while others are more communication oriented. The parameter computation-to- communication ratio is used to determine whether workflow tasks are computation intensive or communication intensive [10].

The process of allocating resources to the user application requests is called **Resource (Cloud) Provisioning**. The cloud resource manager (service provider) optimally deploys user requests to available resources. Resource provisioning further involves two phases:

1. VM Provisioning: Cloud computing is based on Virtualization Technology through which a single computational resource can be shared among different requests. It is a core technology through which multiple virtual machines (VMs) can be allocated to a single physical machine. The hosting of VMs on physical machines (servers) in cloud datacenters is called VM Provisioning [6].
2. Application (task) Scheduling: The mapping of user requests (tasks/ application requests) onto VMs while ensuring best resource utilization with minimum make span, cost and energy utilization.

1.1 Workflow Scheduling

Cloud computing concerns with business, scientific and experimental applications, which are complex and they involve large amount of data. Some examples of such applications are Montage, Cyber shake, weather forecasting, etc. These applications comprise of tasks with dependencies and are named as workflows [13]. Workflow management and scheduling require efficient and effective techniques called workflow management system (WMS) responsible for organizing, managing and optimizing huge amount of distributed data [9].

Workflow scheduling is an NP-hard problem which further becomes more complex and challenging in heterogeneous environment as cloud computing. Further, Resource Provisioning affects the overall performance of workflow [15]. To propose a novel technique for workflow management, optimization, scheduling and to tackle with the challenges faced during these phases is an emerging research area. Due to increased complexity of workflow scheduling, hybrid optimization techniques are being explored by researchers. Heuristics are important for generating polynomial time optimal solution to NP-hard task scheduling problem in cloud environment [18]. Researchers have proposed various heuristics to minimize the task completion time (make span).

In the proposed research, PEFT algorithm with Ant Colony Optimization (ACO) has been used to achieve an optimal schedule with load balancing to avoid overutilization and underutilization of physical machines by virtual Machines. In this paper, a comprehensive model for cloud application workflow scheduling with load balancing has been proposed and implemented for cloud provider and end user perspective. For cloud provider point of view, the proposed model achieves high resource utilization with time and energy efficiency while meeting the end user requirements. On the other hand, the end user is ensured high availability of resources and reduced cost.

The proposed model is based on priority based earlier finish time (PEFT) heuristic for generating initial seed (schedule) for the workflow. Further, to reduce the make span and cost of executing tasks on VMs and improve the overall performance of the cloud datacenter, load balancing of tasks across VMs onto physical machines has been done for better management of resources and to ensure high resource availability when resources are efficiently utilized. Load balancing is based on he Ant Colony Optimization technique to identify the underutilized resources.

2. PROBLEM STATEMENT

Many large e-scientific and e-business applications are modeled by workflows these days. Workflows are represented as a Directed Acyclic Graph (DAG) with a tuple G= (T,E) , where T is the set of tasks and E is the set of edges showing the precedence constraints between tasks.

A successor task cannot be started until all of predecessor tasks have finished. Workflow scheduling is an automation of mapping every task of workflow onto a suitable. Resource to meet the user's QoS requirements without violating task dependencies. Cloud act as an ideal platform for their execution because of its scalability, on demand access and pay per usage model. Due to large number of tasks and VMs, workflow scheduling is a significant issue in cloud computing.

The problem here can be defined as:

Mapping of workflow tasks to available resources in cloud aiming to minimize cost while keeping execution time below the given deadline that is minimize ECsubject to ET $\leq$ D where EC is the total execution cost, ET is the total execution time and D is the deadline of the workflow**.**

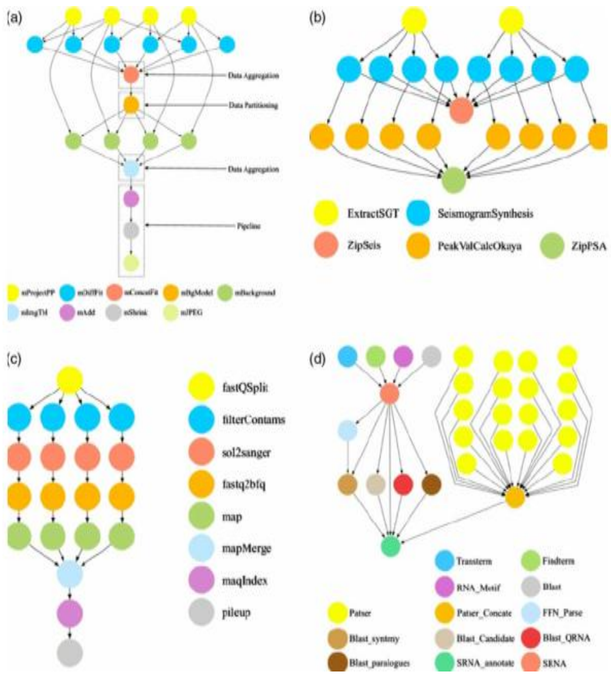

Figure 1: Scientific WorkFlows

3. PROPOSED WORK

In this research work, we are presenting an algorithm which is a hybrid of PEFT and ACO (Ant colony Optimization) for scheduling workflows on the cloud. The algorithm tries to minimize the execution cost while maintaining the deadline constraint.

The proposed work is roughly divided into two main steps:

1. To generate a high quality seed for the ACO using the PEFT heuristic.
2. To obtain an optimized schedule by ACO in such a way that it will have the minimal cost and will finish execution before the workflow deadline.

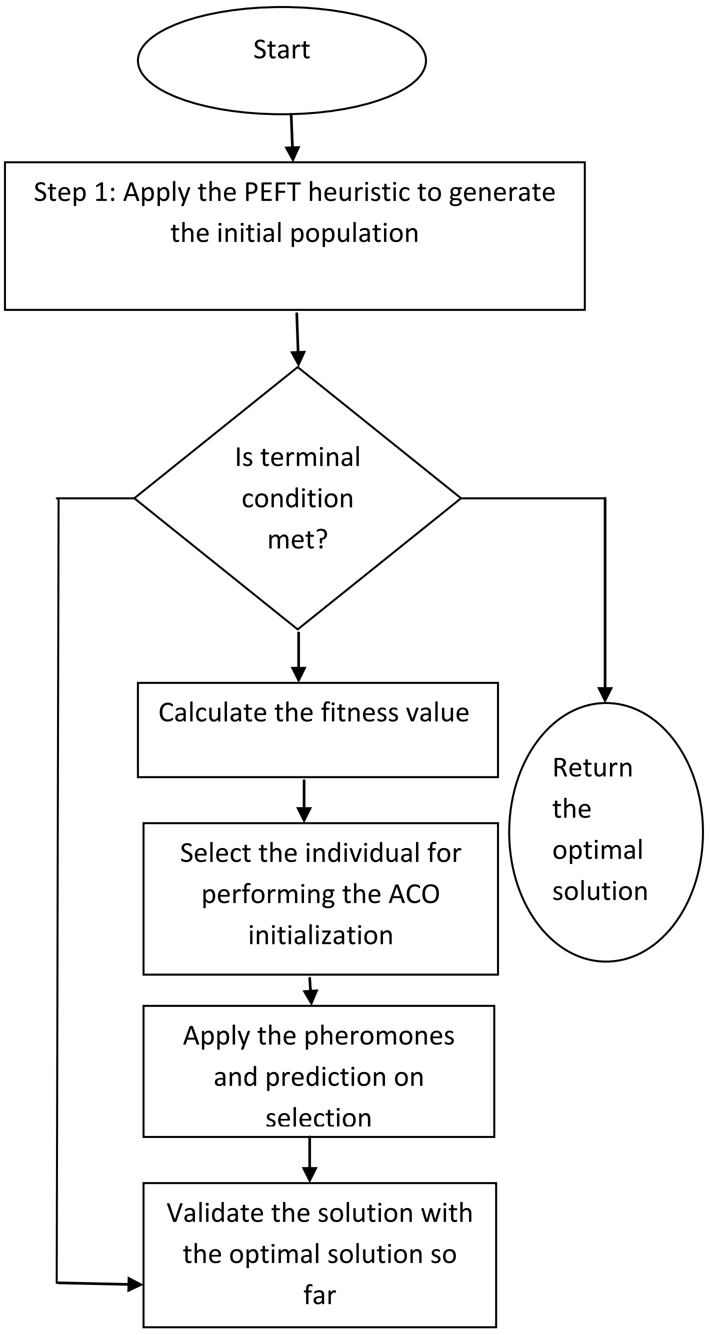
Start
Step 1: Apply the PEFT heuristic to generate the initial population
Is terminal condition met?
Calculate the fitness value
Select the individual for performing the ACO initialization
Apply the pheromones and prediction on selection
Validate the solution with the optimal solution so far
Return the optimal solution

Figure 2: Flow Chart of Proposed work of ACO Initialization and Optimization

Workflow Diagram 2

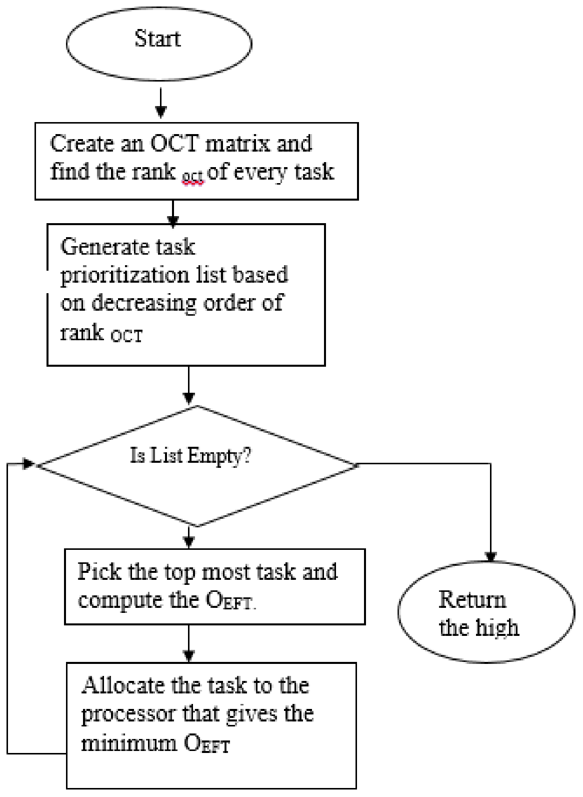
Start
Create an OCT matrix and find the rank oct of every task
Generate task prioritization list based on decreasing order of rank OCT
Is List Empty?
Pick the top most task and compute the OEFT.
Return the high
Allocate the task to the processor that gives the minimum OEFT

Figure 3: Steps of PEFT algorithm which call in fig2

Steps of the proposed methodology:

Step 1: Generate a good seed using the PEFT algorithm.

Step 1.1 Compute the OCT table. The OCT is in the form of a matrix whose rows are the tasks and columns are the VMs. The OCT value is calculated according to the equation below recursively using the backward approach. It will give the cost to execute all the children tasks of a current task until it reaches the last point.

where =0 if p w = p k (1)

Step 1.2: Find cumulative OCT of every node and cumulative OCT defines the rank of every node or task(rank oct) as in Eq 3 . Tasks are ordered in the list on the basis of decreasing order of the rank OCT,

rankoct (t i) = (2)

Step 1.3: Repeat steps 1.4 and 1.5 if the list is not empty, else return the best schedule in terms of make span.

Step 1.4: Optimistic EFT is calculated according to the below equation to allocate a task for the processor using the insertion based policy.

$$O\,EFT(t_i, p_j) = EFT(t_i, p_j) + OCT(t_i, p_j) \qquad (3)$$

Step 1.5: Task is assigned to the processor which will give the minimum O EFT.

Step 2: If the termination condition is met, return the optimal solution else repeat steps 3 to step 6.

Step 3: A high quality schedule thus generated, is seeded it into the Ant Colony Optimization Algorithm. A specific number of ants are used to optimize the resource utilization for the task sequence generated by PEFT.

Step 4: Each ant builds its solution by selecting the next resources based on the current pheromone value for that resource and task.

Step 5: After each ant builds its solution, it is compared to the optimal solution so far. If the optimal solution is updated if required.

ALGORITHM 1

Algorithm: Seeding based on Priority based effective finish time (PEFT)

Input: workflow with 'n' dependent tasks to be scheduled on 'm' processors

CC(ti,tj)	average Communication cost between two successive tasks ti and tj with dependency, CC(ti,tj)=0 if both ti and tj are scheduled on the same processor

EFT(ti,pk) Effective finish time of task ti on processor pk

$Rank_{OCT(ti,pk)}$ Priorities assigned on the basis of number of tasks allocated to VM

t_{entry} and t_{exit} are the first and the last tasks of the workflow

Store tasks ready for execution in 't_Schedule' and initializet_schedule = t_{entry}

Output: Priority based schedule of tasks for execution on Virtual machines(VMs).

```
1    For i = 1 to n
2    Compute optimal cost of executing each task on particular processor
3    $OCT(t_i,p_k)$ =maximum [$OCT(t_j,p_l) + ET(t_j,p_l) + CC(t_i, t_j)$], $CC(t_i, t_j)=0$ if $t_i$ and $t_j$are scheduled on same VM
                         $t_j \in$ child $t_i$
4    Prioritize tasks on the basis of $OCT(t_i,p_k)$, such that
     $Rank_{OCT(ti,pk)} = \sum_{k=1}^{m} \frac{OCT\,(ti,pk)}{m}$
5.   Add tasks ready for execution in t_Schedule according to their Priority ($Rank_{OCT(ti,pk)}$) from highest to lowest
6    End For
7    Repeat
8    Remove the first task $t_i$ with highest rank from t_Schedule
9    for i=1 to n, compute
10       $EFT(t_i,p_k)=EST(t_i,p_k) + ET(t_i,p_k) + CC(t_i, t_j)$
11       $O_{EFT}(ti,p_k)= EFT(ti,pk) + OCT(t_i,p_k)$
12       End for
13   Assign $t_i$ to $p_k$withminimum$O_{EFT}$
14   Add child tasks of $t_i$ to t_Schedule sorted by $Rank_{OCT}$
15   Until (t_schedule != NULL)
```

ALGORITHM 2

Algorithm 2: Ant Colony Optimization based on PEFT seeding

Input: N_p Ant Population (No. of ants refers to the number of schedules generated for assigning tasks to VMs)

T_{ij}	Intensity of pheromone on edge (i,j)
ΔT_{ij}	Amount of pheromone laid by kth ant on edge (i,j)
α	parameter controlling the influence of Tij
μij	Refer to desirability of edge i, j
β	parameter controlling the influence of μij
ρ	pheromone evaporation rate
i_{max}	maximum number of iterations, i=1

Output: Identification of Underutilized nodes with optimal solution based on minimization of Cost and Makespan.

```
1    Repeat
2    For k=1 to $N_p$
3    Assign each ant '$A_k$' resource(node) $r_i$ based on PEFT seeding and save $r_i$in schedule of $A_k$
     / / For ant next to ant $A_k$
4      For m=k+1 to $N_p$
5    Choose the next resource $r_j$ for each $A_m$ with probability: $P_{ij}(t) = \frac{(Tij)^{\alpha}(\mu ij)^{\beta}}{\sum (Tij)^{\alpha}(\mu ij)^{\beta}}$
6    Store the selected resource $r_j$ for each $A_m$ in schedule of $A_m$
7    End For
         // Update Optimal solution
8    Compute the resource utilization $R_k$ by each ant $A_k$(Virtual Machine).
9    Compute $\Delta T_{ij}^k$laid by ant $A_k$ on edge (i,j) as $\Delta T_{ij}^k$ =$\{1/R_k$ if $A_k$ moves on edge i, j|0 otherwise$\}$
10   Update the intensity of pheromone as $T_{ij} = (1-\rho)T_{ij}(t) + \sum_{i=1}^{Np} \Delta T_{ij}^k$
11   Store minimum resource utilization $R_{k\_min}$ as optimal solution.
12     End for
13   Until (i>$i_{max}$)
14   Identify the underutilized nodes with $R_{k\_min}$ less than minimum threshold
15   Perform VM Migration of underutilized nodes
16   Analyze and compare the algorithm on the basis of time and cost parameters
```

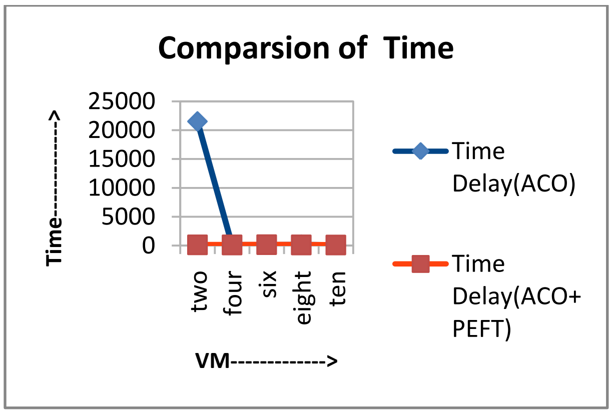

Figure 4: Comparison of Time Delay for Genome (ACO+PEFT)

4. RESULT ANALYSIS

In this section we analyze the proposed approach on two scientific workflows. Analyze the proposed approach and existing approach on parameter energy, cost and time. Units of parameters are JL (joule) for energy, used resource in $ for cost and sec for time delay.

Table 1: Genome Workflow comparison for ACO and PEFT with

VM	two	four	six	eight	ten
Cost (ACO)	112.42	89.86	58.86	109.46	47.79
Energy(ACO)	168.63	24435.39	31241.35	28347.37	30818.51
Time Delay (ACO)	21526.72	121.965	88.29	164.19	71.685
Cost (ACO+PEFT)	108.98	103.16	128.08	102.24	92.88
Energy(ACO+PEFT)	23532.19	24986.84	31130.23	30872.16	24268.1
Time Delay(ACO+PEFT)	136.225	128.95	160.1	127.8	116.1

For the Genome Workflow the comparison for time delay, energy consumption and cost of the resources have been shown in figure 4,5 and 6 respectively. Figure 4 shows that the time delay using ACO+PEFT is significantly less for 2 VM's. For more VM's the time delay is almost the same. Figure 5 states that as the number of VM's increase, the energy consumption by resources using ACO+PEFT decreases significantly as compared to ACO.Time delay using 2VM's is lesser for ACO+PEFT. Overall, for the Genome workflow there is a significant decrease in energy utilization and time delay(makespan).

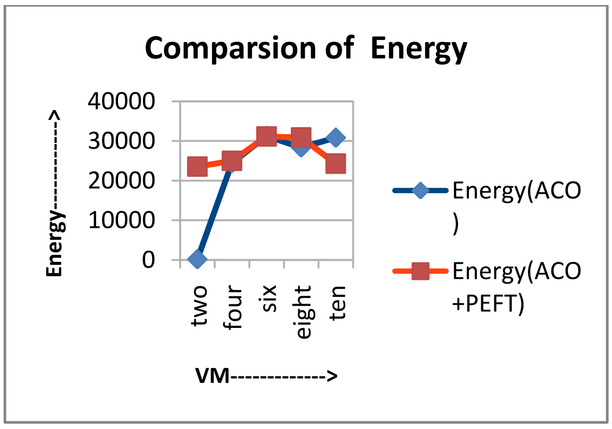

Figure 5: Comparison of Energy for Genome (ACO+PEFT)

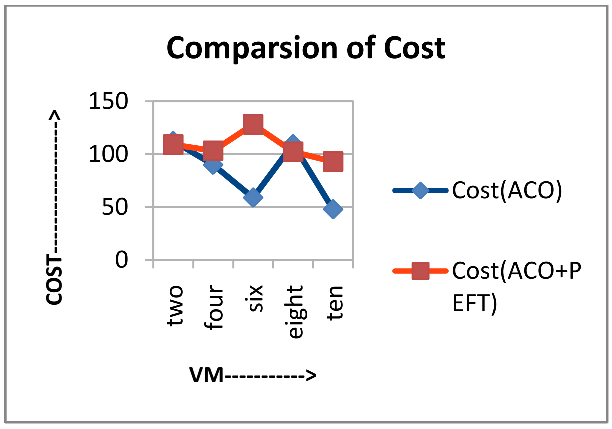

Figure 6: Comparison of Costfor Genome (ACO+PEFT)

Table 2: CybershakeWorkflow comparison for ACO and PEFT and ACO

VM	two	four	six	eight	ten
Cost (ACO)	0.45	3.45	0.32	3.23	2.45
Cost(ACO+PEFT)	0.29	2.49	0.13	2.15	1.73
Energy (ACO)	112	400	102	378.45	345.34
Energy (ACO+PEFT)	100.2756	351.88	99.3966	357.92	336.3398
Time Delay(ACO)	0.567	4.34	0.578	3.567	2.976
Time Delay(ACO+PEFT)	0.435	3.735	0.465	3.225	2.595

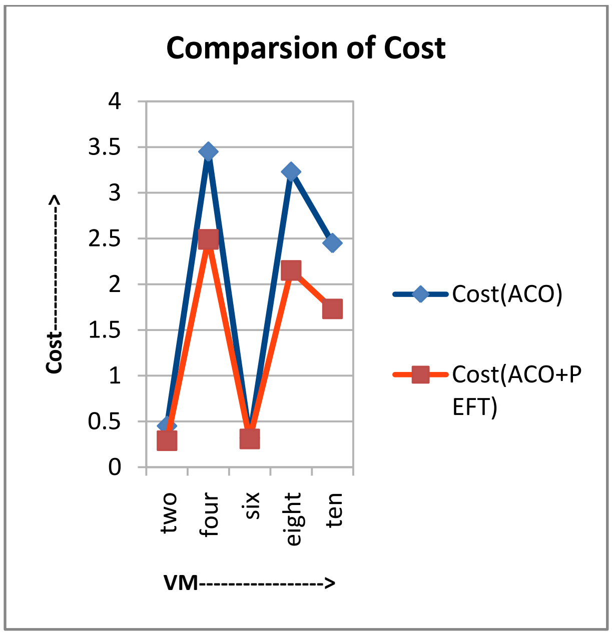

Figure 7: Comparison of Cost for Cybershake (ACO+PEFT)

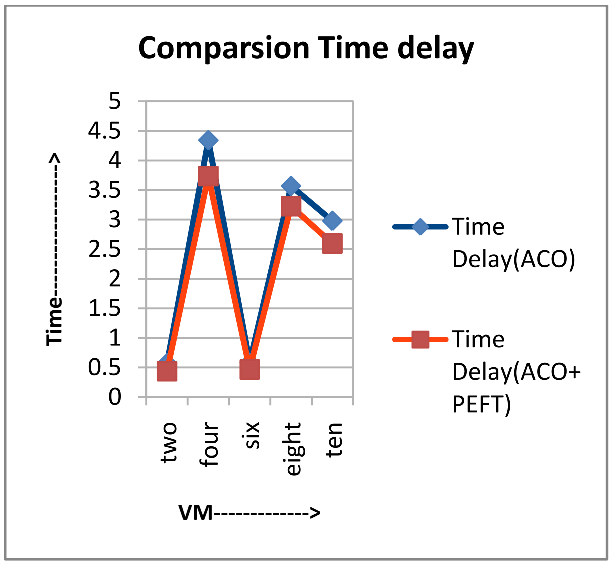

Figure 8: Comparison of Time Delay for Cybershake(ACO+PEFT))

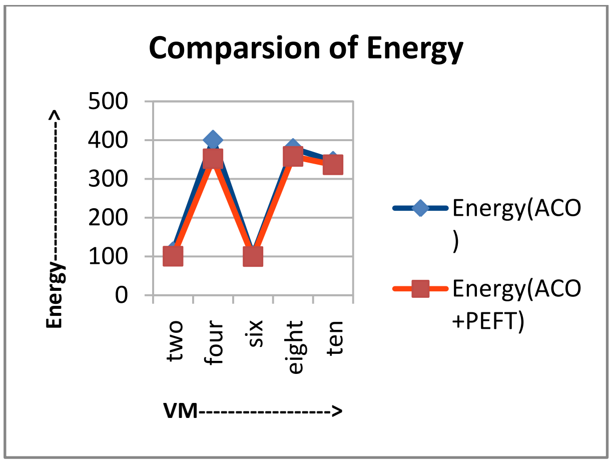

Figure 9: Comparison of Energyfor Cybershake (ACO+PEFT)

For the cybershake workflow, the parameters for cost, time delay and energy consumption have been recorded in figure 7, 8 and 9 respectively. Figure 7 clearly shows that the cost of resources is lesser for our approach. Figure 8 shows that time delay(makespan) is slightly lesser using ACO+PEFT. Energy Consumption by the resources is displayed in figure 9 and found to be almost equal using ACO and ACO+PEFT. For the cybershake workflow, it can be concluded that the cost of utilization of resources is significantly lesser with a slight decrease in makespan.

5. CONCLUSION

In this paper we extend the previous algorithms for deadline-constrained workflow scheduling in utility cloud, to design new algorithms, which are called PEFT with ACO,for the IaaS Cloud environment. The new algorithm reduces the complexity of ACO initialization and speeds up its convergence to the ideal solution. In our experiment we use two work flows: genome with more than 10,000 tasks and cyber shake with more than 1000 tasks, the Genome workflow significant decrease for makespan and energy utilization by resources, while the cybershake workflow showed good decrease in cost of utilization of resources along with slight decrease in makespan.

6. REFERENCES

[1] Q. Zhang, L. Cheng and R. Boutaba, &quot;Cloud computing: state-of- theart and research challenges&quot;,J. Internet Services and Applications, vol. 1, no. 1, 2010

[2] R. Ranjan, L. Zhao, X. Wu, A. Liu, A. Quiroz and M. Parashar, &amp;ldquo,Peer-to- Peer Cloud Provisioning: Service Discovery and Load-Balancing,&amp;rdquo, Cloud Computing, Computer Communications and Networks, N. Antonopoulos and L. Gillam, eds., pp. 195-217, Springer, 2010.

[3] G. Aceto, A. Botta, W. de Donato and A. Pescapè, &quot;Cloud monitoring: A survey&quot;, Computer Networks, vol. 57, no. 9, pp. 2093-2115, 2013.

[4] K. Nuaimi, &quot;A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms&quot;, in Network Cloud Computing and Applications (NCCA), 2012, 2016.

[5] I. FisterJr, X.-S. Yang, I. Fister, J. Brest, and D. Fister.A brief review of nature-inspired algorithms for optimization.Elektrotehniskivestnik, 80(3):116-122, 2013.

[6] E. García-Gonzalo and J. Fernández-Martínez, &quot;A Brief Historical Review of Particle Swarm Optimization (PSO)&quot;, j bioinformintelli control, vol. 1, no. 1, pp. 3-16, 2012.

[7] E. Duman, M. Uysal and A. Alkaya, &quot;Migrating Birds Optimization: A new metaheuristic approach and its performance on quadratic assignment problem&quot;, Information Sciences, vol. 217, pp. 65-77, 2012.

[8] Jianyong Sun, J. Garibaldi and C. Hodgman, &quot;Parameter Estimation Using Metaheuristics in Systems Biology: A Comprehensive Review&quot;, IEEE/ACM Trans. Comput. Biol. and Bioinf., vol. 9, no. 1, pp. 185-202, 2012.

[9] J. Liu, X. Luo, X. Zhang, F. Zhang and B. Li, &quot;Job Scheduling Model for Cloud Computing Based on Multi-Objective Genetic Algorithm&quot;, IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 3, January 2013.

[10] A. Soni, G. Vishwakarma and Y. Kumar Jain, &quot;A Bee Colony based Multi-Objective Load Balancing Technique for Cloud Computing Environment&quot;, International Journal of Computer Applications, vol. 114, no. 4, pp. 19-25, 2015.

[11] M. Abdullah and M.Othman, &quot;Cost-based Multi-QoS Job Scheduling Using Divisible Load Theory in Cloud Computing&quot;, Procedia Computer Science, vol. 18, pp. 928-935, 2013.

[12] D. Santos, A. de Sousa and F. Alvelos, &quot;A hybrid column generation with GRASP and path relinking for the network load balancing problem&quot;, Computers &amp; Operations Research, vol. 40, no. 12, pp. 3147-3158, 2013.

[13] Y. Jiang, Z. Shao, Y. Guo, H. Zhang and K. Niu, &quot;DRSCRO: A Metaheuristic Algorithm for Task Scheduling on Heterogeneous Systems&quot;, Mathematical Problems in Engineering, vol. 2015, pp. 1-20, 2015.

[14] F. Zhang, J. Cao, K. Li, S. Khan and K. Hwang, &quot;Multi-objective scheduling of many tasks in cloud platforms&quot;, Future Generation Computer Systems, vol. 37, pp. 309-320, 2014.

[15] A. Beloglazov, J. Abawajy and R. Buyya, &quot;Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing&quot;, Future Generation Computer Systems, vol. 28, no. 5, pp. 755-768, 2012.

[16] T. Keskinturk, M. Yildirim and M. Barut, &quot;An ant colony optimization algorithm for load balancing in parallel machines with sequence-dependent setup times&quot;, Computers &amp; Operations Research, vol. 39, no. 6, pp. 1225-1235, 2012.

[17] M. Yakhchi, S. Ghafari and S. Yakhchi, &quot;Proposing a load balancing method based on Cuckoo Optimization Algorithm nergyanagement in cloud computing infrastructures&quot;, in 6th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO), Istanbul, 2015.

[18] J. Adhikari and S. Patil, &quot;Double threshold energy aware load balancing in cloud computing&quot;,in Computing, Communications and Networking Technologies (ICCCNT),2013 Fourth International Conference, Tiruchengode, 2013, pp. 1 - 6.

[19] J. Doyle, R. Shorten and D. O&#39;Mahony, &quot;Stratus: Load Balancing the Cloud for Carbon Emissions Control&quot;, IEEE Transactions on Cloud Computing, vol. 1, no. 1, pp. 1-13, 2013.

[20] S. Ahmad, C. Liew, E. Munir, T. Ang and S. Khan, &quot;A hybrid genetic algorithm for optimization of scheduling workflow applications in heterogeneous computing systems&quot;, Journal of Parallel and Distributed Computing, vol. 87, pp. 80-90, 2016.