Optimized Resource Scheduling using the Meta Heuristic Algorithm in Cloud Computing

G. B. Hima Bindu , K. Ramani and C. Shoba Bindu, *Member, IAENG*

***Abstract*— From the past decade, the utilization of the cloud environment has been increased drastically. Many web applications are relying on the computing environment of the cloud. Therefore, resource scheduling for the client tasks is treated as the major issues challenged by the cloud. This paper concentrated on client QoS requirements and developed the optimized scheduling mechanism. The Optimized scheduling mechanism using ACO algorithm (OSACO) is proposed to reduce the energy, cost and time. The proposed OSACO algorithm performance is compared with the existing algorithms. The simulation results proved the effective optimization compared to the other existing algorithms.**

***Index Terms*— Ant Colony, Computational cost, Energy Consumption, Task scheduling.**

I. Introduction

CLOUD computing offers many services by using the pay-as-you-use policy [1]. The cloud services that are offered to the clients having different components may be collected from different service providers. The cloud services must be offered by using the QoS parameters to satisfy the client's requirements. However, execution of many tasks in single virtual machine leads to degradation of performance which leads to unsatisfactory of clients. Resource scheduling is the major issue faced by the cloud computing [2]. Assigning of tasks to the resources is a complex mechanism which can be solved using the optimization techniques. Heterogeneity, Dispersion are the major problems faced by the resource scheduling which cannot be solved by the traditional scheduling algorithms in cloud. Therefore, it is important to consider all these factors in developing the cloud services and applications.

G. B. Hima Bindu is currently research scholar at JNTUA College of engineering, Ananthapuramu working towards the Ph. D degree in Computer Science & Engineering in the area of Cloud Computing. himabindugbe@gmail.com

Dr. K. Ramani is working as Professor & Head in department of Information Technology, Sree Vidyanikethan Engineering College, Tirupati ramanidileep@yahoo.com

Dr. C. Shoba Bindu, received her Ph. D degree in Computer Science and Engineering from JNT University, Ananthapur. She is currently working as Professor in the Department of Computer Science & Engineering, JNTUA College of Engineering, Ananthapuramu. shobabindhu@gmail.com

Resource scheduling is the mechanism of allocating the proper resource to the particular task to improve the scaling advantage [3]. Less number of resources should be used for task execution with QoS. In cloud computing, reducing the energy consumption is a major task at the time of task execution in the virtual machines (VMs). The energy is consumed due to the utilization of CPU, memory and hard disc. The Green cloud is proposed to reduce the energy consumption and also to improve the utilization of CPU [7-8]. In [9], the authors developed the green computing mechanism for cloud by managing the energy consumption as well as QoS requirements.

In the resource scheduling mechanism, data centers, virtual machines, brokers and cloudlets are utilized for performing scheduling process based on client's requirements [4]. Different Metaheuristic mechanism are developed to solve the scheduling issues in cloud computing such as GA [5] and PSO [6] algorithms. Ant Colony Optimization (ACO) is one of the efficient approaches used for optimization problems. ACO is a bio-inspired approach which is based on the pheromone value of the ants' movement. This paper proposes the multi objective ACO approach which uses the pheromone value as the matching parameter to allocate the resources to the task. The remaining sections are structured as follows: Section 2 deals with the related work regarding the existing scheduling mechanism for cloud. Section 3 deals with architecture of resource scheduling in cloud. Section 4 deals with the Optimized scheduling mechanism using ACO algorithm. Section 5 explains about the result analysis of the proposed model and section 6 concludes the research work.

II. Literature Survey

Resource scheduling is one of the challenging tasks in the cloud computing due to the dynamic nature of resources which are distributed geographically. Most of the research work is concentrated only on the resource scheduling mechanisms based on the client's requirements. In [11], the author proposed the workflow scheduling mechanism to find the best possible solution based on the clients QoS requirements. The authors consider the execution time as the QoS parameter and showed the improvement in response time and they didn't consider the cost as the QoS parameter. In [12], the authors developed the scheduling approach for homogeneous resources by reducing the energy consumption. The research challenge faced by the authors in resource scheduling is on demand provisioning. The applications can change the requirements and it is a typical task to assign the resources to them. The authors proposed the algorithm based on the bids. This will balance the resource cost and also efficiently utilize the servers.

In [13], Ashkan et al proposed the energy efficient scheduling model for cloud. The major functionality of the proposed algorithm is to manage the servers by applying sleep mode to the servers to optimize the energy consumption. In [14], authors proposed prediction and consolidation algorithm for energy efficient computing. This algorithm considers the workload prediction of the resources to schedule the VM. The major advantage of these algorithms is to optimize the energy consumption by reducing the underutilized server. In [15], the authors developed the Heterogeneous framework for finding the execution time of and communication time between the tasks. The task with highest rank is given the top priority for allocating the resources. The limitation of this algorithm is that it is concentrated only on execution time but not considered the cost.

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 [26].

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. The 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 [25]. It involves high time complexity due to the number of reschedulings happen at the time of algorithm execution.

III. Procedure for Resource Scheduling

The process of resource scheduling is composed of four modules. In the first module, the tasks are analysed and grouped based on their resource requirements. In the second module, resource set is identified based on the available resources from the pool. In the third module, the tasks should be mapped with the appropriate resources based on the client specified QoS requirements. In the final module, the scheduling process is initiated for achieving the optimal satisfaction of the clients. The proposed method satisfies the requirements of the resource scheduling based on the clients QoS requirements. Figure 1 explains about the architecture of the proposed model.

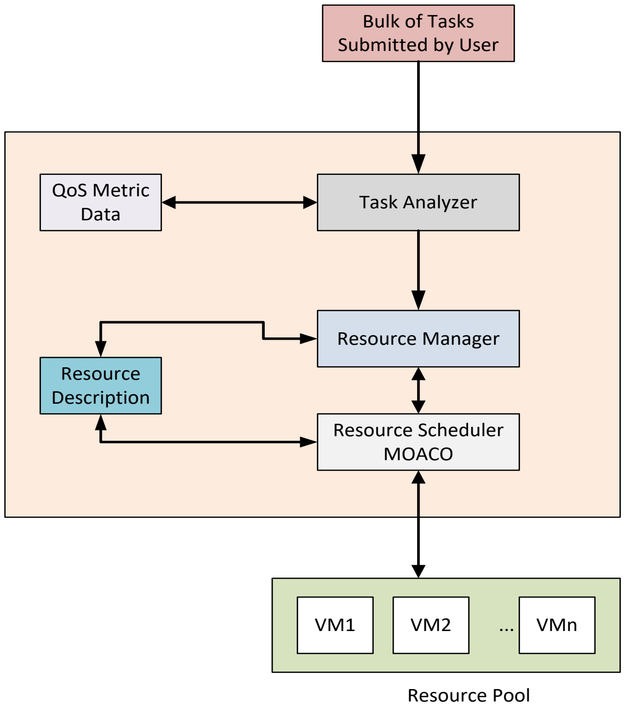

Figure 1: Model for Resource scheduling in Cloud

Figure 1 shows the resource scheduling mechanism in cloud. The resource scheduling mechanism involved with different type of modules which are listed below:

Bulk of Tasks: The application submitted by the user contains bulk of tasks which are forwarded to the task analyser for further processing.

Task Analyser: The major responsibility of task analyser is to analyse the different properties of the task to identify the suitability of the resource in the cloud. The task properties are not unique and they are different based on their QoS requirements. All the tasks submitted to the analyser will analyse the tasks depending on the QoS parameters.

QoS Metric Data: This module computes the cost of the resources based on the task QoS requirements.

Resource Description: The resource description contains the information about memory, CPU, cost, type of the VM.

Resource Manager: This module performs the resource allocation to the tasks for execution. The resource allocation depends on the availability of the resources in the resource pool. If the requested resources are not available in the cloud then it request the user to resubmit the task after some time.

Resource Scheduler: The resource scheduler is responsible for executing the allocated tasks to the resources efficiently. Section 4 describes about the procedure of proposed resource scheduling mechanism.

A. Problem Formulation

Resource scheduling is one of the major challenges faced by the cloud computing. Finding the best possible combination of task and resources based on the client's requirements is a tedious mechanism. The client wants to reduce the cost and whereas the service providers have to decrease the energy consumption and execution time. In the proposed work, we are considering the multi objective function such as reducing the cost, energy consumption and execution time. The problem formulation has been given below.

Let us consider that there is set of independent tasks $\{x_1,x_2\ldots x_m\}$ to be mapped with the set of heterogeneous resources$\{y_1,y_2\ldots y_n\}$.

$$X = \{x_i \mid 1 \le i \le n\}$$
$$Y = \{y_j \mid 1 \le j \le m\} \quad (1)$$

In Eq. 1, X denotes the collection of tasks and 'n' denotes the number of tasks, Y represents the collection of resources and 'm' denotes the number of resources.

B. Objective Function

The main goal of the proposed work is to reduce the cost, execution time and energy consumption. The objective function is formulated in Eq. 2. Further, the task allocation problem with cost, energy consumption and execution time of each resource 'y' is given as follows:

$$f = P\,exec_{cost} + Q\,exec_{time} + R\,energy_{consumption} \quad (2)$$

Where P, Q and R represent the weights of the elements to prioritize in the fitness function.

B.1. Minimization of Execution Cost:

It is defined as the cost incurs to execute the tasks in the cloud which is measured in dollars ($).

$$exec_{cost} = \min(ct(x_i, y_j))\ where\ 1 \le i \le n, 1 \le j \le m, \quad (3)$$

Where $ct(y_j, x_i)$ represents the cost incurred for executing the task 'x_i' on resource 'y_j'.

$$ct(x_i, y_j) = \sum_{x_i \in X} \frac{cl(x_i, y_j)}{(cl_n(x_i) \times X)} \quad (4)$$

Where X is the collection of tasks, $cl(x_i, y_j)$ represents the completion time of the task 'x_i' on resource 'y_j'.

$$cl_n(x_i) = \max_{x_i \in X, y_j \in Y} cl(x_i, y_j) \quad (5)$$

B.2. Minimization of Execution Time:

The $exec_{time}$ is defined as the completion time of the current task $L_x$and also it is represented as expected completion time (ECT)of task 'x_i' on resource 'y_j'.

$$exec_{time} = \min(cl(L_{x_i}))\ where\ x_i \in X \quad (6)$$

Before computing the execution time of the task, the completion time of the resource should be calculated.

$$cl(y_j) = avail\_time(y_j) \pm ECT_{n(x_i)} \quad (7)$$

Where

$$ECT_{n(x_i)} = \max_{x_i \in X, y_j \in Y} ECT(x_i, y_j) \quad (8)$$

B.3. Minimization of Energy Consumption:

$energy_{consumption}$ is derived based on the energy consumption for utilization of resources. Eq. 9 shows the energy consumption of the resources.

$$energy_{consumption} = DC_{EC} + Memory_{EC} + TS_{EC} + Misc_{EC} \quad (9)$$

DC_{EC} denotes the energy consumption of the data center utilization, $Memory_{EC}$ denotes the energy consumption of the memory utilization, TS_{EC} denotes the energy consumption of the all switching devices. $Misc_{EC}$ represents the miscellaneous energy consumption.

Eq. 10 shows the energy consumption at particular time't' for resource 'y'. Here z is the energy consumption of the idle resource, $\max_{EC}$ represents the maximum energy consumption of the resource, 'U' represents the utilization of resource.

$$EC_{t,i}(y) = z \times \max_{EC} + (1 - z) \times \max_{EC} \times U \quad (10)$$

IV. Optimized Scheduling Mechanism Using ACO Algorithm

The Ant Colony Optimization algorithm has the advantage of solving the optimization problems with different combination solution set. The ACO algorithm solves the resource scheduling problem by simulating the ants foraging process. As an initial step, the ants choose the random paths to reach their desired targets. After reaching the targets, the ants calculate the fitness of the path by estimating the pheromone value. Finally, the path with highest fitness value will be chosen as optimal solution to the problem.

A. Fitness Function

Fitness function is used to estimate the quality of the solution. According to the resource scheduling mechanism, this paper considers three objectives such as minimizing the cost, reducing the energy consumption and minimizing the execution time. Eq. 11 shows the fitness function for the proposed ACO approach.

$$Fit(f) = \min(f) \quad (11)$$

B. Pheromone Update

The ants find the optimal path based on the strength of the pheromone. Therefore, it is important to update all possible path pheromone value. The pheromone update mechanism is given in Eq. 12.

$$\gamma(x_i, y_j) = (1 - \alpha) \times \gamma(x_i, y_j) + \Delta\gamma(x_i, y_j) \quad (12)$$

Where $\gamma(x_i, y_j)$ is the pheromone of the scheduling set (x_i, y_j) ,α is the evaporation factor of pheromone, $\Delta\gamma(x_i, y_j)$ represents the incremental value of the pheromone.

$$\Delta\gamma(x_i, y_j) = \begin{cases} \delta(f), & where (x_i, y_j) \in path_w \\ 0, & otherwise \end{cases} \quad (13)$$

Where δ is the constant value. If the three elements in the fitness function are smaller, the pheromone value will be higher.

The implementation procedure of optimization scheduling using ACO algorithm is given in Algorithm 1.

Algorithm 1: Optimization scheduling using ACO algorithm (OSACO)

Input :{x_1,x_2…x_m} -> set of tasks
{y_1,y_2…y_n}->set of resources
I_{max} -> maximum iterations
Output: Scheduling (x_i, y_j)

Step 1:Begin
Step 2: Initialize the searching procedure by distributing the ants over resource y_j;
Step 3: For each ant do
Step 4: For each task x_i do
Step 5: find the next possible path
Step 6: End for
Step 7: calculate the fitness function by using the Eq. 11 for all paths
Step 8: If y_i meets the requirements of the QoS requirements then
Step 9: Assign the y_j to x_i
Step 10: perform pheromone update using the Eq. 12 and 13.
Step 11: End If
Step 12: End For
Step 13: until it reaches the I_{max}
Step 14: End

V. SIMULATION SETUP AND RESULT ANALYSIS

The proposed model is simulated using the efficient cloud simulation tool called as CloudSim tool kit [21]. The resource tasks and resource characteristics are given in Table 1 and Table2. Client application tasks are modeled as independent and resources are modeled as heterogeneous in the simulation environment.

Table 1: Task configuration details

Parameter	Value
File size	[100, 1000]MB
CPU length	[200, 1500] MIPs
Output Size	[40-80]MB
Number of Tasks	[20-100]
Output Size	[40-80]MB

Table 2: Resource configuration details

Parameter	Value
RAM	[1028-4096] MB
Storage	[10-30]GB
CPU Computing capacity	[1860-2660] MIPs
Bandwidth	200 M/s
Cost	[2-5]$/unit time

The performance of the OSACO algorithm is compared with the existing algorithms such as traditional ACO algorithm [22], Genetic algorithm [23] and DVFS-MODPSO algorithm [24]. The constant value for the iteration is taken as 100 for both in proposed and existing algorithm. The simulation environment tested 20 groups of different P,Q,R and γ combinations.

A. Result Analysis

The experiment is performed with different number of tasks (20-100) for QoS verifications.

Case1: Execution Cost with respect to Number of Tasks

Figure 2 shows the execution cost of the tasks with respect to 2 resources. It is observed that the increase in number of tasks leads to the increase in the execution cost. As the tasks increases, OSACO performs better compared to the ACO, GA and DVFS-MODPSO. The performance of the OSACO algorithm is better due to the ability of resource adjustment at runtime. The minimum cost of the proposed OSACO algorithm is 10$ for 20 tasks and maximum cost is 48$ for 100 tasks which is given in Table 3.

Table 3: Comparison of Execution cost with respect to OSACO, ACO, GA, DVFS-MODPSO

Number of Tasks	Algorithms			
	OSACO	ACO	GA	DVFS-MODPSO
20	10	12	14	18
40	18	21	24	28
60	27	29	32	36
80	36	39	43	46
100	48	52	54	59

Case 2: Energy Consumption with respect to Number of Tasks

Within the increase in the number of tasks will automatically increase in the energy consumption. Figure 3 shows the performance of the proposed model with respect to energy consumption. The minimum value of the OSACO is recorded as 5J for 20 tasks and maximum value is recorded as 19J for 100 tasks. The OSACO average energy consumption is recorded as 12J which is shown in Table 4. It is better compared to the ACO, GA and DVFS-MODPSO.

Table 4: Comparison of Energy Consumption with respect to OSACO, ACO, GA, DVFS-MODPSO

Number of Tasks	Algorithms			
	OSACO	ACO	GA	DVFS-MODPSO
20	5	8	12	7
40	9	12	15	10
60	12	18	21	14
80	15	20	25	16
100	19	24	28	21

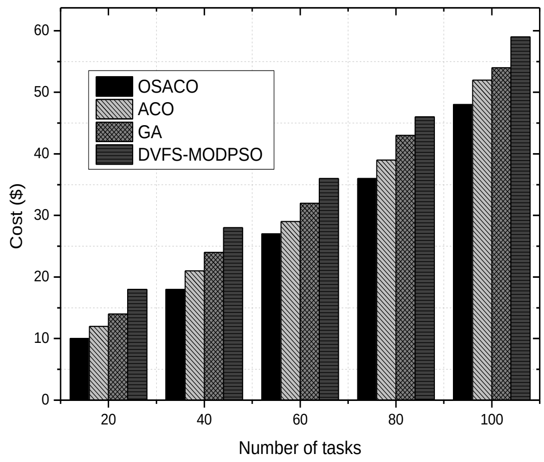

Figure 2: Execution Cost Vs Number of Tasks

Case 3: Execution Time with respect to Number of Tasks

From Figure 4, it is observed that the proposed OSACO algorithm has better performance in terms of execution time compared to the ACO, GA, DVFS-MODPSO algorithms. The execution time recorded for OSACO algorithm is 23% lesser than the ACO, 29% lesser than the GA and 27% lesser than the DVFS-MODPSO algorithms which is shown in Table 5.

Case 4: Execution cost with respect to Number of Resources

Within the increase in the number of resources leads to the increase in the execution cost. Figure 5 shows the execution cost of the resources for 20 tasks. It is identified that the cost of OSACO algorithm recorded as 4$ for 2 VMs and the maximum cost is recorded as 20$ for 10VMs which is given in Table 6. The OSACO algorithm shown minimal cost compared to the ACO, GA and DVFS-MODPSO algorithms.

Case 5: Energy Consumption with respect to Number of Resources

With the increase in the number of resources will automatically increase in the energy consumption. Figure 6 shows the performance of the proposed model with respect to energy consumption. The minimum value of the OSACO is recorded as 5J for 2 resources and maximum value is recorded as 36J for 10 resources. The OSACO average energy consumption is recorded as 20.6J which is shown in Table 7. It is better compared to the ACO, GA and DVFS-MODPSO.

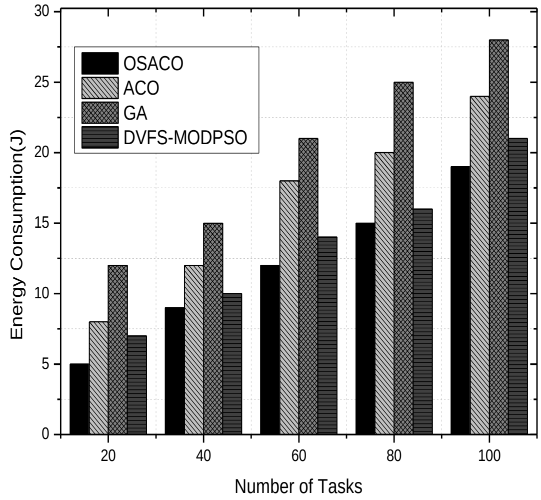

Figure 3: Energy Consumption Vs Number of Tasks

Table 5: Comparison of Execution Time with respect to OSACO, ACO, GA, DVFS-MODPSO

Number of Tasks	Algorithms			
	OSACO	ACO	GA	DVFS-MODPSO
20	15	24	31	29
40	25	38	45	42
60	48	56	61	51
80	54	69	74	71
100	69	72	79	75

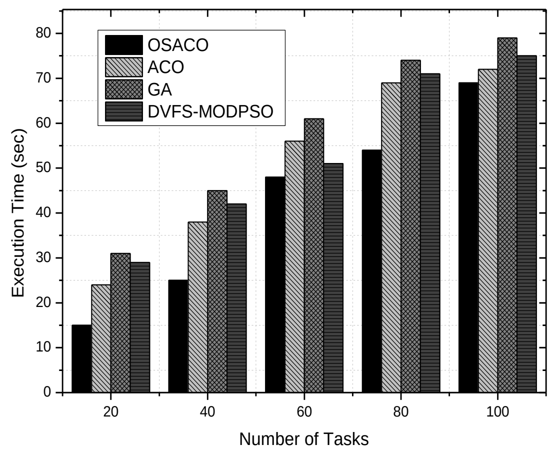

Figure 4: Execution Time Vs Number of Tasks

Table 6: Comparison of Execution Time with respect to Number of Resources

Number of Tasks	Algorithms			
	OSACO	ACO	GA	DVFS-MODPSO
2	4	6	7	9
4	8	10	12	13
6	12	14	15	17
8	16	19	20	24
10	20	23	24	26

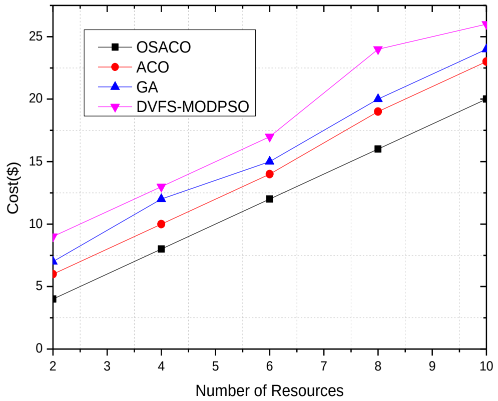

Figure 5: Execution Cost Vs Number of Resources

Table 7: Comparison of Energy Consumption with respect to Number of Resources

Number of Tasks	Algorithms			
	OSACO	ACO	GA	DVFS-MODPSO
2	5	8	11	4
4	12	15	18	8
6	21	24	26	15
8	29	32	35	21
10	36	39	42	24

Case 6: Execution Time with respect to Number of Resources

From Figure 7, it is observed that the proposed OSACO algorithm has better performance in terms of execution time compared to the ACO, GA, DVFS-MODPSO algorithms. The minimum value of the OSACO is recorded as 5 sec for 10 resources and maximum value is recorded as 15 sec for 2 resources which is shown in Table 8.

Table 8: Comparison of Execution time with respect to Number of Resources

Number of Tasks	Algorithms			
	OSACO	ACO	GA	DVFS-MODPSO
2	15	24	31	27
4	12	21	27	25
6	9	16	21	18
8	7	12	18	16
10	5	9	14	13

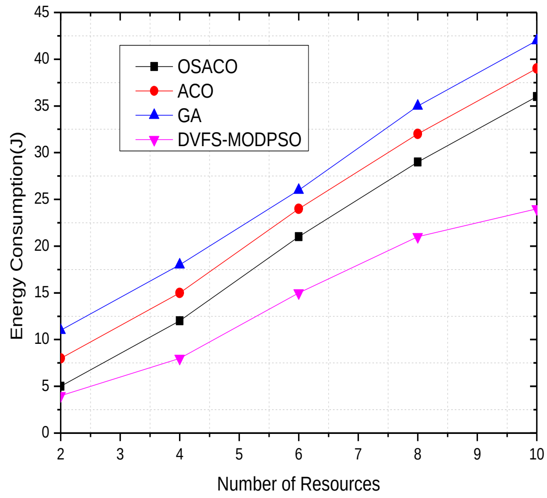

Figure 6: Energy Consumption Vs Number of Resources

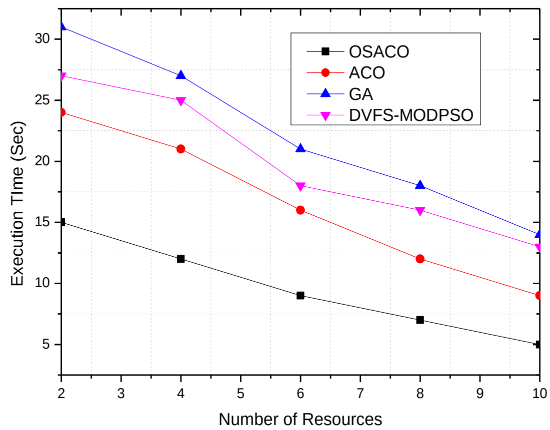

Figure 7: Execution Time Vs Number of Resources

VI. CONCLUSION

This paper proposed the OSACO algorithm to schedule the task to the resources based on the minimization of cost, execution time and energy consumption. Simulation results demonstrate that the OSACO algorithm is efficient in reducing the optimization parameters. The performance of the OSACO algorithm is better due to the ability of resource adjustment at runtime. The cost of the proposed OSACO algorithm is 10$ for 20 tasks and it is more efficient compared to the existing algorithms. The energy consumption of the OSACO is recorded as 5J for 2 resources and maximum value is recorded as 36J for 10 resources. The proposed method has the efficient mechanism to pair the tasks to resources based on the client QoS requirements. In future, we will consider different factors like availability, reliability and fault tolerance in the resources to improve the scheduling mechanism..

REFERENCES

[1] Singh, S., Chana, I., Singh, M.: The journey of QoS based autonomic cloud computing. IT Prof. Mag. 19(2), 42–49 (2017).

[2] Moens, H., Truyen, E., Walraven, S., Joosen, W., Dhoedt, B., De Turck, F.: Cost-effective feature placement of customizable multi-tenant applications in the cloud. J. Netw. Syst. Manag. 22(4), 517–558 (2014).\

[3] Chiang, M.L.: Efficient diagnosis protocol to enhance the reliability of a cloud computing environment. J. Netw. Syst. Manag. 20(4), 579–600 (2012).

[4] Suruchi, S., Kuila, P.: Design of dependable task scheduling algorithm in cloud environment. In: Proceedings of the Third International Symposium on Women in Computing and Informatics, pp. 516–521. ACM, New York (2015).

[5] M. Rahman, X. Li, and H. N. Palit. Hybrid heuristic for schedulingdata analytics workflow applications in hybrid cloud environment. inProceedings of the 25th IEEE International Symposium on Parallel andDistributed, ser. IPDPS Workshops. Anchorage (Alaska) USA, Anchorage(Alaska) USA:966-974, May 2011.

[6] M. A. Rodriguez and R. Buyya. Deadline based resource provisioningand scheduling algorithm for scientific workflows on clouds. IEEETRANSACTIONS ON CLOUD COMPUTING, 2014.

[7] Kliazovich, Dzmitry, Johnatan E. Pecero, Andrei Tchernykh, Pascal Bouvry, Samee U. Khan, and Albert Y. Zomaya. "CA-DAG: Modeling communication-aware applications for scheduling in cloud computing." Journal of Grid Computing 14, no. 1 (2016): 23-39.

[8] Alsughayyir, Aeshah, and Thomas Erlebach. "Energy Aware Scheduling of HPC Tasks in Decentralised Cloud Systems." In Parallel, Distributed, and Network-Based Processing (PDP), 2016 24th Euromicro International Conference on, pp. 617-621. IEEE, 2016.

[9] Li, Yibin, Min Chen, Wenyun Dai, and MeikangQiu. "Energy optimization with dynamic task scheduling mobile cloud computing." IEEE Systems Journal 11, no. 1 (2017): 96-105.

[10] Li, Hao, Hai Zhu, GuohengRen, Hongfeng Wang, Hong Zhang, and Liyong Chen. "Energy-Aware Scheduling of Workflow in Cloud Center with Deadline Constraint." In Computational Intelligence and Security (CIS), 2016 12th International Conference on, pp. 415-418. IEEE, 2016.

[11] Varalakshmi, P., Ramaswamy, A., Balasubramanian, A., Vijaykumar, P.: An optimal workflow basedscheduling and resource allocation in cloud. In: Advances in Computing and Communications,pp. 411–420. Springer, Berlin (2011).

[12] Orgerie, Anne-Cecile, Marcos Dias de Assuncao, and Laurent Lefevre. "A survey on techniques for improving the energy efficiency of large-scale distributed systems." ACM Computing Surveys (CSUR) 46, no. 4 (2014): 47.

[13] AshkanPaya and Dan C. Marinescu:, “Energy-aware Load Balancing and Application Scaling for the Cloud Ecosystem”, IEEE 2015.

[14] MehiarDabbagh, BechirHamdaoui, Mohsen Guizani, AmmarRayes, “Towards Energy-Efficient Cloud Computing: Prediction, Consolidation, and Over commitment”, IEEE 2015.

[15] Topcuoglu, H., Hariri, S., Wu, M.-Y.: Task scheduling algorithms for heterogeneous processors. In:Heterogeneous Computing Workshop (HCW’99). San Juan (1999).

[16] Song, B., Hassan, M.M. and Huh, E.N., 2010, November. A novel heuristic-based task selection and allocation framework in dynamic collaborative cloud service platform. In Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on (pp. 360-367). IEEE.

[17] Tayal, S., 2011. Tasks scheduling optimization for the cloud computing systems. IJAEST-INTERNATIONAL JOURNAL OF ADVANCED ENGINEERING SCIENCES AND TECHNOLOGIES, 1(5), pp.111-115.

[18] Li, J.F., Peng, J., Cao, X. and Li, H.Y., 2011. A task scheduling algorithm based on improved Ant Colony Optimization in cloud computing environment.EnergyProcedia, 13, pp.6833-6840.

[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, 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, July 2005,pp. 140–47.

[21] Calheiros, R.N., Ranjan, R., Beloglazov, A., Rose, C.A.F., Buyya, R.: CloudSim: a toolkit formodeling and simulation of cloud computing environments and evaluation of resource provisioningalgorithms. Soft: Pract. Exper. 41(1), 23–50 (2011).

[22] Vinothina, V., and R. Sridaran. "An Approach for Workflow Scheduling in Cloud Using ACO." In *Big Data Analytics*, pp. 525-531. Springer, Singapore, 2018.

[23] Shishido, Henrique Yoshikazu, JúlioCezarEstrella, Claudio Fabiano Motta Toledo, and Marcio Silva Arantes. "Genetic-based algorithms applied to a workflow scheduling algorithm with security and deadline constraints in clouds." *Computers & Electrical Engineering* 69 (2018): 378-394.

[24] Masdari, Mohammad, FarbodSalehi, MarzieJalali, and MoazamBidaki. "A Survey of PSO-based scheduling algorithms in cloud computing." Journal of Network and Systems Management 25, no. 1 (2017): 122-158.

[25] Xu Liya, Deng Anyuan, Ge Mingzhu, Shi Jiaoli, and Gao Guangyong, "A MADM-based Handover Management in Software-defined 5G Network," Engineering Letters, vol. 27, no.4, pp842-849, 2019.

[26] Jun Yang, Anding Dai, Chao Wei, Jing Zhao, Peifeng Li, Xiaomei Gao, and Mingmei Zhang, "M-Matrix-Based Exponential Synchronization of Delayed Neural Networks with Levy Noise and Markovian Switching," Engineering Letters, vol. 27, no.4, pp855-859, 2019.