**International Journal of Innovative Technology and Exploring Engineering (IJITEE)**
**ISSN: 2278-3075, Volume-8 Issue-2S2 December, 2018**

# Multi objective Ant colony Optimization Algorithm for Resource Allocation in Cloud Computing

**Prasad Devarasetty, Ch. Satyananda Reddy**

***Abstract**: Cloud computing provides the services based on the pay-as-you-use policy. The more utilization of services leads to the utilization of more number of data centres. Therefore, data centres require high energy consumption for computing the tasks. To improve the efficiency of the data centre, resource management using the virtualization technology is the crucial factor. This paper concentrates on the issue of virtual machine placement and also proposes the bio inspired approach for reducing the resource wastage, minimize the energy consumption and communication cost with in the data centre. Ant Colony Optimization (ACO) algorithm is proposed to obtain the solution set for multi-objective problem. The performance of the proposed algorithm is tested with the existing algorithms and it is proved that the proposed algorithm is efficient in terms of energy consumption, communication cost and resource utilization.*

***Key words**: Virtual placement, Cloud, Consolidation, Communication cost, Resources.*

## I. INTRODUCTION

From the past decade, cloud computing [1] is treated as one of the efficient computing platforms for service delivery. Cloud computing have the unlimited resources to serve any type of services [6]. The users can use the resources based on the pay-as-you-use policy. In the recent years, many organizations like Google, Amazon, Yahoo and Amazon moved their services in to the cloud for better serving of customers. The service providers use many servers in data centres to operate the services and this leads to the huge energy consumption. According to the researchers, the energy utilization of the data centre is almost equal to the energy consumed by the twenty five thousand households [2-5].

From the industry point of view, reducing the energy consumption leads to minimizing the cost of data centre. The major approach followed for reducing the energy consumption is to turn off the unwanted server and effective utilization of the allocated resources. Virtual machine placement and resource consolidation are the common mechanisms used for efficient utilization of resources in cloud computing. Virtual machine placement is a mechanism to map the virtual machines to the physical machines [7]. This method involves several issues. One of the issues is NP-hard problem. Resource consolidation is the process of selecting the resources which has to be migrated. Due to the NP-hard nature of the issues, the studies only concentrated on developing the single objective functions and also the studies only developed greedy approaches to solve the issues.

This paper concentrated on developing the multi-objective approach which mainly concentrates reducing the energy consumption, increasing the resource utilization and reducing the communication cost between the network components to the data centre. The rest of the paper is discussed as follows. Section 2 deals with the related work regarding the approaches previously suggested by the researchers to address the issues of energy consumption, reducing the communication cost and increasing the resource utilization. Section 3 deals with the problem statement and formulation for the proposed model. Section 4 explains about the multi objective ACO algorithm for VM placement. Section 5 deals with performance evaluation of the proposed model compared to the existing system in terms of defined objectives. Finally, section 6 concludes the paper.

## II. RELATED WORK

According to the recent studies, the major algorithms follow the greedy approach to find the optimal solution for the issue of VM placement. These algorithms have the less time complexity to solve the issue compared to the Meta-heuristic algorithms. Though, they are relying on the centralized procedure and it is hard to follow the distribution process in the greedy algorithms.

The algorithms like permutation pack, First fit Decreasing (FFD) [3] and Choose pack are the greedy approaches which are used for the VM placement in cloud [8]. According to the experimental results Choose pack performs faster compared to the FFD and permutation pack. Leinberger and Karypis [9] are the authors who proposed Permutation pack and Choose pack greedy algorithms. In [10], the authors developed Best Fit Decreasing Algorithm (BFDA) which considers CPU utilization for VM placement. The parameters considered for the evaluation of the proposed model is VM migrations, energy consumption and SLA violations. In [11], the author studied the utilization of Markov models for identifying the overloaded VMs in the cloud. The assumption made in this model is not practically applicable to the real time workloads [12].

**Revised Manuscript Received on December 08, 2018.**

**Prasad Devarasetty**, Department of Computer Science and Engineering, DVR &Dr HS MIC College of Technology, Vijayawada, Andhra Pradesh, India

**Dr. Ch. Satyananda Reddy**, Department of Computer Science and Systems Engineering,AU College of Engineering, Andhra University, Visakhapatnam, Andhra Pradesh, India

Meta-heuristic algorithms such as Ant colony optimization (ACO) [12-14] and Genetic Algorithm (GA) [15-16] are proposed to find the optimal solution to the issues of VM placement. These algorithms provide optimal solutions to the NP-hard problems. In [12], the authors proposed ACO based on the bin packing algorithm. The objective of this algorithm is to provide less number of bins for all the resources. The performance of the ACO algorithm is compared with the CPLEX and FFD algorithms. This model only concentrates on energy consumption and it is applicable for single objective function. In [13], the authors proposed the ACO algorithm considering the power consumption and resource utilization as the objectives in VM placement. The experimental results of the algorithm are tested with SACO and FFD along with multi objective Genetic Algorithm (MGGA). But the algorithm failed to incorporate the communication cost as the objective to evaluate the algorithm. In [17], the authors proposed the GA for VM placement. The authors considered the objectives as communication cost and power consumption. However, GA has not proved their efficiency in terms of reducing the energy consumption. In [20], the authors developed the mechanism of GAACO approach for scheduling the IoT tasks to the cloud. It is a combination of both GA and ACO algorithm.

## III. PROBLEM STATEMENT AND FORMULATION

### *3.1 Assumptions*

In this paper, we consider that physical machines (PM) doesn't have any virtual machines (VM) before executing the migration algorithm and also the utilization of the data centres is managed by the number of VM's request and server's request. The energy consumption of the VM's is majorly considers the CPU utilization instead of bandwidth and memory [21]. The authors concentrate on CPU utilization rate to minimize the energy consumption.

### *3.2 Formulation of the objective function*

Let us consider that M represents the number of VMs and P represents the number of physical machines PM, R denotes the group of resources required by each VM. $\beta_j$ identifies whether $PM_j$ is active or in idle state. $\delta_{ij}$ represents whether $VM_i$ is assigned to $PM_j$ or not.

### *3.2.1 Minimizing Energy consumption*

The primary objective function of the proposed model is to reduce the energy consumption based on the equation given below.

$$E_j = (E_j^{active} - E_j^{idle}) \times \eta_j^E + E_j^{idle} \tag{1}$$

Where $\eta_j^E$ represents the CPU utilization and it lies between 0 and 1, $E_j^{active}$ and $E_j^{idle}$ represents the average energy consumption values when the $j^{th}$ physical machine is active or in idle states.

The overall energy consumption of the physical machine is calculated as follows:

$$\min \sum_{j=1}^{P} E_j^{PM} = \sum_{j=1}^{P} \left[ \beta_j \times \left\{ (E_j^{active} - E_j^{idle}) \times \sum_{i=1}^{M} (\delta_{ij} \times K_{i,1}^{VM}) + E_j^{idle} \right\} \right] \tag{2}$$

Where K represents the set of resources assigned to the VM.

### *3.2.2 Reduction of resource wastage*

The secondary objective of the proposed model is to reduce the resource wastage. Here, we consider that Y is the resource wastage of each physical machine and $K^{PM}$ is the group of available resources at the physical machine. $K_{i,1}^{VM}$ is the group of resources request by the virtual machine $VM_i$. The objective for the minimization of resource wastage is given in eq. 3.

$$\min \sum_{j=1}^{P} Y_j^{PM} = \sum_{j=1}^{P} \left[ \beta_j \times \left\{ (K_{i,1}^{PM} - (\sum_{i=1}^{M} (\delta_{ij} \times K_{i,1}^{VM}))) \times (\sum_{i=1}^{M} (\delta_{ij} \times K_{i,1}^{VM}))^{-1} \right\} \right] \tag{3}$$

### *3.2.3Minimizing the energy of the communication cost*

The minimization of the communication cost is considered as the third objective of the proposed model. Here, L is considered as the communication overhead between two VMs. The objective function for the communication cost is shown in eq. 4.

$$\min \sum_{g=1}^{N} E_g^{NE} = \sum_{g=1}^{N} \left[ \alpha_g \times [(E_g^{active} - E_g^{idle}) \times \sum_{i=1}^{M} (\delta_{is} \times L_{i,1}^{MM}) + E_g^{idle}] \right] \tag{4}$$

Where g is the number of network elements which ranges from 1 to N, α represents the network element whether it is active or idle state and it is denoted with 1 and 0. Here, K-shortest path algorithm is used to find the network elements in between two virtual machines .

## IV. MULTI OBJECTIVE ANT COLONY OPTIMIZATION (MOACO) ALGORITHM

The proposed algorithm is mainly concentrated on solving the problem which is stated on section 3. An Optimal Solution for the virtual machine assignment problem is considered as the VM placement permutation.

Algorithm 1 explains about the procedure of MOACO algorithm. The working procedure of algorithm 1 is given as follows. In the initial stage, the ACO parameters are initialized and all the pheromone trails are set to $t_0$. At the initial stage, each ant has to receive all the VM requests, and assigns the VMs to available hosts.

### *4.1 Pheromone trail*

The best solution to the problem is depends on the quality of the pheromone trail. In this paper, we are considering the method of assigning VM 'i' to the physical machine 'j'. In the initial stage, the pheromone value is calculated as follows.

$$t_0 = \frac{1}{M[E^{PM}(s_0) + Y^{PM}(s_0) + E^{NE}(s_0)]} \tag{5}$$

Where M represents the number of virtual machines, EPM (S0) represents the energy consumption of the solution S0, YPM(S0) denotes the resource wastage of the solution S0 and the ENE (S0) represents the energy consumption of the networking elements at solution S0.


Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

*4.2 Heuristic Information*

Along with pheromone trail, another important factor needs to be considered in ACO application of selection of good heuristic which will be combined with pheromone trail to produce best solutions. The heuristic information is represented with $H_{i,j}$. This information provides the compatibility of assigning the VM 'i' to physical machine 'j'. In order to access each move of the ant in the ACO algorithm, the heuristic information has to be calculated dynamically by considering the current state of the ant. To improve the efficiency, this algorithm has to calculate the heuristic function to all moves of ants. Let PM be a list of all physical machines. When constructing the solution, every ant starts assigning VMs to list of PMs arranged randomly. The partial assignment of VM 'i' to physical machine 'j' is calculated as follows.

$$H_{i,j,1} = \frac{1}{\varepsilon + \sum_{j=1}^{p} (E_j / E_j^{\max})} \qquad (6)$$

---

Algorithm 1: Multi Objective Ant Colony Optimization Algorithm (MOACO)

---

Input : List of VMs and List of Physical Machines with resource request and the set of parameters.

Output: Pareto set PST

Begin

/*Initialization */

1. Initialize M number of VMs, P number of Physical machines, NA-> Number of Ants,
   I-> Number of iterations, $\rho_l$-> pheromone local update, $\rho_g$-> pheromone global update,
   α-> represents the parameter that usually controls the pheromone trail, $t_0$->intial pheromone value, $q_0$-> uniform random number in {0,1}
2. Initialize Pareto set PST->0
3. Initialize all pheromone values to $t_0$
4. For j=1 to NA do
   a. Sort the list of physical machine PM in random order
   b. Introduce a new physical machine from the list PM
5. For each remaining VM that can be assigned to the current physical machine do
   a. Evaluate the desirable ant movement

$$H_{i,j} = \frac{1}{\varepsilon + \sum_{j=1}^{p} (E_j / E_j^{\max})} + \frac{1}{\varepsilon + \sum_{j=1}^{p} (Y_j^{PM})} + \frac{1}{\varepsilon + \sum_{j=1}^{p} (E_j^{NE} / Max(E_j^{NE}))}$$

   b. Evaluate the probability of the ant movement

$$Pb_{i,j}^{k} = \begin{cases} \dfrac{\alpha \times t_{i,j} + (1-\alpha) \times H_{i,j}}{\sum_{u \in \Omega_k(j)} (\alpha \times t_{i,j} + (1-\alpha) \times H_{i,j})} & i \in \Omega_k(j) \\ 0 & otherwise \end{cases}$$

   End for
6. Identify q
7. If q<=$q_0$ then

8. Identified as a best solution

9. Else

10. Search for new solution

11. End if

12. Apply the local updating rule

$$t_{i,j} = (1-\rho_l)t_{i,j}(t-1) + \rho_l . t_0$$

13. Until all VMs are assigned to the physical machine

14. End for

15. Evaluate the vales of three objectives for each solution in the existing ant population

16. If the solution is non-dominated when compared to other solutions in the existing ant population. This solution is added to the pareto set PST.

17. For each non-dominent solution of PST do

a. $$t_{i,j} = (1-\rho_g)t_{i,j}(t-1) + \frac{\rho_g . \lambda}{E^{PM}(s) + Y^{PM}(s) + E^{NE}(s)}$$

18. End for

19. Until it reaches to the maximum number of iterations

20. Return PST

End

## V. PERFORMANCE EVALUATION

The performance of the proposed algorithm is tested with the CloudSim [2] simulator. The CloudSim simulator allows to moving one VM per event, but in the proposed model, we required global repacking. Here, we simulated the data centre with different configurations of the physical machines. The working module of the CloudSim simulator is composed of three modules: cloud settings, host description and workflow traces. Each PM is configured with 1 CPU core of 2000, 4000 and 6000 MIPS, 256GB of storage and 4GB of RAM. Each VM requires 1GB of storage, 128MB of RAM and 1 CPU core with 500, 750 and 1000 MIPS each. To test the performance of the proposed model, the instructions taken for the application is about 20000 Million instructions. The virtual machines are assigned to the specific servers which are then fully utilized.

*5.1 Simulation Results*

The evaluation of the proposed algorithm is carried with other existing optimization algorithms such as MOGA [18] and DVFS [19]. The three objective functions use for evaluation of the proposed model is resource wastage, energy consumption and communication cost.

Figure 1 shows the differentiation of the energy consumption for different optimization techniques. The DVFS algorithm had highest energy consumption compared to the MOACO and MOGA. But the MOGA and MOACO has the better mechanism in reducing the energy consumption. So, it is proved that the MOACO performs well in minimizing the overall energy consumption.

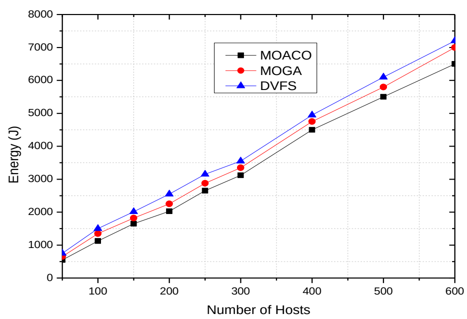


**Figure 1: Energy consumption of the hosts in the physical machines**

Figure 2 shows the ratio of number of available hosts at the time of VM assignment. It is observed that MOACO has utilized minimum number of hosts for VM assignment compared to the MOGA and DVFS. The minimum utilization of the hosts leads to the effective utilization of the resources which ultimately leads to the effective performance increase.

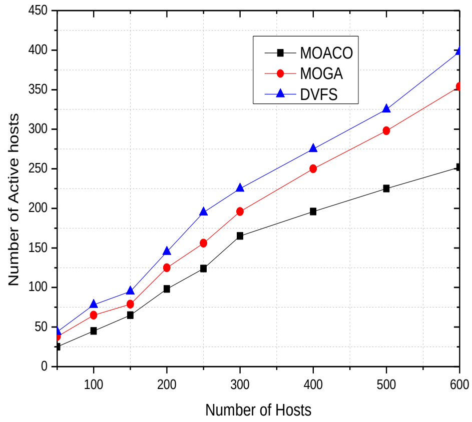


**Figure 2: Number of Active hosts in the physical machines**

Figure 3 shows the utilization of the resources at the time of application execution. The results proved that the MOACO finds the better solution compared to the MOGA and DVFS in all objectives. For instance, for 20 PMs, 30 VMs and 30 tasks, we got 0.6%, 1.66% and 40 % resource wastage for MOACO, MOGA and DVFS. Among 20 servers, the MOACO only utilized 9 servers for completing tasks execution.

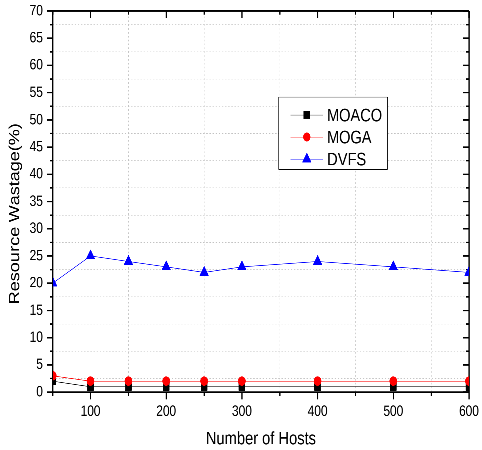


**Figure 3: Resource Wastage Vs Number of Hosts**

Figure 4 shows the communication cost of the MOACO, MOGA and DVFS. In general, the MOACO had better performance in terms of reducing energy consumption and minimizing resource wastage. But compared to the single objective algorithms, the MOACO and MOGA had little performance improvement. However, among the available candidates, the MOACO has the ability to select a solution with the lowest Communication cost.

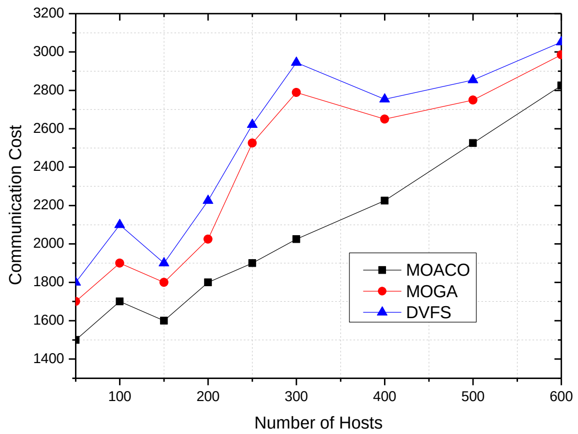


**Figure 4: Communication Cost Vs Number of Hosts**

The performance of the proposed system is given in Table 1. It is proved that the proposed system recorded minimal energy consumption, less number of hosts and reduced resource wastage and less communication cost compared to the MOGA and DVFS algorithms.

**Table 1: Comparison of the MOACO, MOGA and DVFS**

| Algorithms | Energy consumption (J) | No. of Active Hosts | Resource Wastage (%) | Communication Cost |
|---|---|---|---|---|
| MOACO | 6517 | 261 | 0.6 | 2812 |
| MOGA | 7052 | 355 | 1.66 | 3029 |
| DVFS | 7125 | 417 | 40 | 3127 |

## VI. CONCLUSION

This paper proposed the MOACO algorithm which is compared with MOGA and DVFS algorithms with well-known simulation tool called as CloudSim. The performance evaluation demonstrates that MOACO minimizes energy consumption compared to the other algorithms by considering resource utilization along with energy communication cost. On an average 12% of energy was reduced through the MOACO compared to the DVFS. However, DVFS has minimal execution time compared to the MOACO and MOGA. The execution time of the MOACO and MOGA is optimized when they are operated in the large cloud computing environments. In the future, the functionality of the proposed algorithm is extended to more number of data centres.

## REFERENCES:


1. Zhang, Qi, Lu Cheng, and RaoufBoutaba. "Cloud computing: state-of-the-art and research challenges." Journal of internet services and applications 1, no. 1 (2010): 7-18.
2. Calheiros, Rodrigo N., Rajiv Ranjan, Anton Beloglazov, César AF De Rose, and RajkumarBuyya. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." Software: Practice and experience 41, no. 1 (2011): 23-50.
3. J. G. Koomey, "Worldwide electricity used in data centers," Environ. Res. Lett. vol. 3,2008.
4. E. Feller, "Autonomic and Energy-Efficient Management Of Large-Scale Virtualized Data Centers ", UNIVERSITÉ DE RENNES, 2013.
5. M. Stillwell, F. Vivien, and H. Casanova, "Virtual Machine Resource Allocation for Service Hosting on Heterogeneous Distributed Platforms," presented at the Proceedings of the 2012 IEEE 26th International Parallel and Distributed Processing Symposium, 2012.
6. Hayes, B., 2008. Cloud computing. Communications of the ACM, 51(7), pp.9-11.
7. Dikaiakos, Marios D., DimitriosKatsaros, PankajMehra, George Pallis, and Athena Vakali. "Cloud computing: Distributed internet computing for IT and scientific research." IEEE Internet computing 13, no. 5 (2009).
8. E. Coffman Jr, J. Csirik, G. Galambos, S. Martello, and D. Vigo, "Bin Packing Approximation Algorithms: Survey and Classification," in Handbook of Combinatorial Optimization, P. M. Pardalos, D.-Z. Du, and R. L. Graham, Eds., ed: Springer New York, 2013, pp. 455-531.
9. W. Leinberger, G. Karypis, and V. Kumar, "Multi-Capacity Bin Packing Algorithms with Applications to Job Scheduling under Multiple Constraints," presented at the Proceedings of the 1999 International Conference on Parallel Processing, 1999.
10. A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," Future Generation Computer Systems, vol. 28, pp. 755- 768, 5// 2012.
11. A. Beloglazov and R. Buyya, "Managing Overloaded Hosts for Dynamic Consolidation of Virtual Machines in Cloud Data Centers Under Quality of Service Constraints," Parallel and Distributed Systems, IEEE Transactions on, vol. PP, pp. 1-1, 2012.
12. E. Feller, L. Rilling, and C. Morin, "Energy-Aware Ant Colony Based Workload Placement in Clouds," in Grid Computing (GRID), 2011 12th IEEE/ACM International Conference on, 2011, pp. 26-33.
13. Farahnakian, Fahimeh, TapioPahikkala, PasiLiljeberg, JuhaPlosila, and HannuTenhunen. "Utilization prediction aware vm consolidation approach for green cloud computing." In Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on, pp. 381-388. IEEE, 2015.
14. Sigwele, Tshiamo, AtmShafiulAlam, PrashantPillai, and Y. Fun Hu. "Evaluating energy-efficient cloud radio access networks for 5G." In Data Science and Data Intensive Systems (DSDIS), 2015 IEEE International Conference on, pp. 362-367. IEEE, 2015.
15. Wang, Tingting, Zhaobin Liu, Yi Chen, YujieXu, and Xiaoming Dai. "Load balancing task scheduling based on genetic algorithm in cloud computing." In Dependable, Autonomic and Secure Computing (DASC), 2014 IEEE 12th International Conference on, pp. 146-152. IEEE, 2014.
16. Dam, Scintami, GopaMandal, KousikDasgupta, and ParamarthaDutta. "Genetic algorithm and gravitational emulation based hybrid load balancing strategy in cloud computing." In Computer, Communication, Control and Information Technology (C3IT), 2015 Third International Conference on, pp. 1-7. IEEE, 2015.
17. Ramezani, Fahimeh, Jie Lu, JavidTaheri, and FarookhKhadeerHussain. "Evolutionary algorithm-based multi-objective task scheduling optimization model in cloud environments." World Wide Web 18, no. 6 (2015): 1737-1757.
18. Zhang, Miao, Huiqi Li, Li Liu, and RajkumarBuyya. "An adaptive multi-objective evolutionary algorithm for constrained workflow scheduling in Clouds." Distributed and Parallel Databases (2017): 1-30.
19. Wu, Tingming, HaifengGu, Junlong Zhou, Tongquan Wei, Xiao Liu, and Mingsong Chen. "Soft error-aware energy-efficient task scheduling for workflow applications in DVFS-enabled cloud." Journal of Systems Architecture 84 (2018): 12-27.
20. Basu, Sayantani, MarimuthuKaruppiah, K. Selvakumar, Kuan-Ching Li, SK Hafizul Islam, Mohammad Mehedi Hassan, and MdZakirulAlamBhuiyan. "An intelligent/cognitive model of task scheduling for IoT applications in cloud computing environment." Future Generation Computer Systems (2018).
21. Perumal, Boominathan&Karuppiah, Marimuthu&Shenoy, A &Hooda, G & Reddy, S. (2014). An approach towards defense of DDOS attacks in cloud computing using confidence based filtering and hop count filtering techniques. International Journal of Applied Engineering Research. 9. 7177-7190.



*Published By:*
*Blue Eyes Intelligence Engineering*
*& Sciences Publication*