ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 8, Issue 3, 2019

Proposing an innovative method for optimizing virtual machine placement in the cloud through the resolution of an optimization problem

Rajesh Kumar E[1]

[1,2] Department of CSE, Koneru Lakshmaiah Education Foundation,

Vaddeswaram, AP, India. rajthalopo@gmail.com,

Abstract

Efficient virtual machine placement is a pivotal challenge in Cloud Computing with far-reaching implications, encompassing economic considerations. The primary goal is to judiciously assign a specified virtual machine to its optimal Physical Machine. Various research-oriented methodologies have been scrutinized for scheduling virtual machines onto suitable Physical machines. This paper introduces a novel approach, leveraging the discrete Gravitational Search Algorithm, to adeptly place virtual machines on designated hosts. This method incorporates optimization and relinking strategies employing two components, departing from conventional approaches. Experimental results demonstrate better performance compared to existing algorithms.

Keywords: Discrete Gravitational Search Algorithm, Virtual Machine Placement, Cloud Computing.

1. Introduction

Cloud computing (CC) has become a popular technical advancement in recent years due to its major benefits in terms of robustness and reliability [1]. Because of the enormous rise of cloud computing with respect to storage efficiency, many applications are migrating to the cloud for reliability and safety purpose. When effective scheduling is required in modern cloud centers, virtual machine (VM) placement to its appropriate Physical Machine (PM) is a crucial challenge to be resolved [2]. A good VM placement in the right PM will make the cloud more cost-effective in terms of energy, latency, and bandwidth. Many researchers have proposed different approaches to address VM placement difficulties.

Author Li et al. [8] have defined the challenges that arise in online VM placement, the author offers the EAGLE algorithm, which is based on a multi-dimensional space model. The

ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 8, Issue 3, 2019

major goal is to reduce operating PMs and minimizing the energy usage. The aim is to find a deal between balancing by utilizing multi-dimensional resources. During the deployment of the VMs, the number of PMs is kept to a minimum for every timeslot. The space out is divided into three different phases related to a one-dimensional resource. The suggested approach evaluates the posterior resource utilization status for every viable PM when a new job appears in the VM's placement. later, the EAGLE selects a perfect suited PM and performs the basic operations. Excessive resource fragments are being boycotted as part of the start-up process for a new PM by [9] Liu et al. The author suggested VM model's multi-objective placement has the potential to reduce the number of traffic-based communications. It allows for proper multi-dimensional resource consumption balancing within a data center. The proposed approach aids in the improvement of the NS-GGA evolutionary based multi goal. It delivers a design to improve the dominated sorting of NSGA-II into Genetic Algorithms in a faster method.

With multi-objective placement issue, Wang et al. [10] suggest a new strategy. The author proposed a scheme method that is simple since it employs average value inequality and positional limitations. It also includes a local heuristic method and an upgraded grouping genetic algorithm (GGA) for construction.To adapt operators to a local genetic heuristic model using a genetic selectivity technique. Elitism method is incorporated to improve placement discovery. Yang et al. [11] proposed a technique called “virtual machine placement with traffic configuration algorithm (VPTCA)”. It is characterized as a planning strategy based on an energy-efficient data center network that gathers data to properly deploy virtual machines. It also provides information on how to effectively place virtual machines using less energy consumption. The proposed scheme's theme is to reduce traffic to maintain good communication allocated to the physical machine.The authors Jamali et al. [12] sought to improve GGA by offering an efficient encoding technique. Using the proposed crossover model, it generates novel results. The difficulty of managing various sets of virtual machines, as well as power consumption and energy loss, is minimized. The Particle Swarm Optimization (PSO) model is responsible for producing quick speed convergence. PSO-related model for allocating transferred VMs was proposed by author Dashti et al. [13]. The proposed method may dynamically centralize the under-loaded hosts in crowded hosts, resulting in higher power savings. The integrated PSO algorithm was used to produce a flawless mapping of one VM to each dimension of the particle's position. The term "position value" refers to the host plus as well as the cost of migration. The framework's goal is to save money while maintaining QoS in the private cloud through energy

ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 8, Issue 3, 2019

conservation. Wang et al. [14] primarily focused on data center energy usage, and the model is described in terms of energy consumption. The initial way to update the particle's position is characterized as the readjustment or redefining of parameters to lower the amount of energy with local fitness. The second option for lowering energy use is to use a two-dimensional (2D) particle encoding scheme.

Gao et al. [15] developed an ant colony algorithm that achieved non-dominated outcomes, demonstrated a proper and effective method of increasing resource usage and power consumption. To distinguish a VM, this approach employs 2D. As a server node, it includes CPU and memory consumption. The proposed approach is designed to prevent memory and CPU bottlenecks. It accepts an upper constraint on resource implementation with a threshold value of a single server to reduce full use of CPU processing. By considering multi-resource restrictions, Dong et al. [16] proposed an Ant Colony Optimization-based (ACO-based) VM positioning and placement solution. These PM limitations aid in improving system performance by optimizing overall system traffic. This method uses VM Placement to help improve scalability by reducing total traffic. It adopts multiple virtual machines in a row, allowing the traffic layout between them to be adjusted. The distribution of network traffic is equalized by lowering the network's Maximum Connection Utilization (MLU). This procedure is followed to avoid congested hotspot areas. To boost the performance, the ACO is combined with a 2-opt local search method. This results in huge data calculations, high time complexity and gradual convergence.

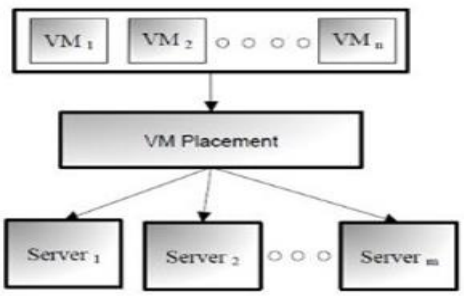

Figure 1: VM Placement

2. Problem definition

The main objective is to use the lower number of physical machines that are possible to

ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 8, Issue 3, 2019

schedule the specified number of virtual machines (VMs). for example, as shown in figure 1, $VM_1, VM_2 \ldots\ldots\ldots VM_n$which takes 25%, 40%, 30%, 35%, and 20% of Physical Machine resources, respectively.

In terms of CPU processing, all PMs will have a maximum threshold capacity to hold any number of machines. Let us assume that each PM's maximum threshold is 70 percent. Now the challenge is to assign each VM to a PM so that the total number of PMs is kept to a minimum and the maximum 70 percent utilization constraint is not violated. The VM Positioning is formulated as in Eq 1.

$$Minimize\{PM_N | N = \{VM_1, VM_2, \ldots\ldots, VM_n\}\} \quad (1)$$

3.Methodologies

3.1. Ant colony Optimization Model (ACO)

Author [15] proposed an ACO to solve the VM placement challenge, that had achieved some uncontrolled outcome to increase the power consumption and resource utilization.The topic of VM Positioning around a server node pool is modelled as vector packing with multidimensional issues like resource consumption. To two-dimensional scheme is used to define server node and VM, processor and memory utilization. Server CPU utilization is computed as twice the amount of single VM utilization for two VMs on same server. A threshold value is fixed as an upper limit for the resource consumption with single server to avoid more memory and CPU utilization. The main concept is significant concert degradation occurs due to maximum usage of CPU on nodes that are migrating by VM migration technology.

3.2. First Fit Decreasing (FFD)

Based on the order of piece that is suitable can be fixed in the bin, in same manner the PM can considered based on this approach.

3.4. Gravitational Search Model Search components

It is important to understand how the algorithm operates to identify the principal search components of GSA. Suppose an agent population is initialized in a space solution of a problem. On applying GSA in a top-down manner to solve the optimization problem, it seeks “to move” the agents position in the solution space of the problem. Therefore, operator is the main GSA component to achieve the search for optimization problem in the solution space. From Eq. (10), the operator used achieve two different parameters namely the length of

ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 8, Issue 3, 2019

movement of i-th agent and its position, and finally it returns a solution as new position to solution space. Every agent movement can be expressed in speed. From Eq. (9), two types of movement lengths are measured for the movement duration of the ith agent itself they are Dependent Movement Length (DML) and Independent Movement Length (IML). Notice that IML is the duration of action that is obtained without any information about the current iteration position other than its own for each agent. For the ith agent, this form of movement length depends only on its previous motion length (or previous velocity), and in reality is a fraction of the ith agent's previous action length.

On the other hand, DML is the length of motion that is obtained by considering the location of all Lbest set members for each agent. Of the Eq. (7) displays the i^{th} agent DML calculation in which all elements of Lbest attempt to influence it. Eq. based. (7), for the i^{th} agent itself the measurement of DML depends the position of the j^{th} agent acts as a sub-component. $(j \in Lbest)$ the space between the agents of i^{th} and j^{th}, the gravity mass of every agent of j-th (i.e., Qj(x)) and gravitational coefficient (Gc(x)) value.

3.5 Proposed Discrete Gravitational Search Algorithm

The proposed Gravitational Search Approach (GSA) is a metaheuristic population based on a stochastic algorithm for solving continuous nonlinear problems [3]. Because VM placement is a combinatorial optimization problem, it is necessary to convert continuous GSA to discrete GSA to solve it. For solving combinatorial optimization issues, Dowlatshahi et al. suggested discrete GSA [4]. After calculating the present fitness using Eq. 2 and Eq. 3, the gravitational mass (GM) of each agent is measured to solve VM placement.

$$P_i(x) = \frac{fitt_i(x) - worstt(x)}{bestt(x) - worstt(x)} \quad (2)$$

$$Q_i(x) = \frac{P_i(x)}{\sum_{j=1}^{s} P_j(x)} \quad (3)$$

Here, $Q_i(x) and fit_i(x)$ indicate GM and fitness for agent i with time x. and $worst(x) and best(x)$ is used for minimization task as shown in Eq. 4 and Eq. 5.

$$best(x) = \underset{j \in \{1,2 \ldots s\}}{Max} fit_j(x) \quad (4)$$

$$worst(x) = \underset{j \in \{1,2 \ldots s\}}{Max} fit_j(x) \quad (5)$$

The overall force acting on the i^{th} agent starting from the set of L agents that are heavier (called the Lbest set) can be computed using Eq. (6) to calculate the acceleration

ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 8, Issue 3, 2019

of the i[th] agent and then the overall force can be split by this agent's gravitational mass, i.e. $Q_i(x)$ with the use of Eq. 7.

$$\varphi_i{}^u(x) = \sum_{j \in Lbest j \neq i} rand_j Gc(x) \frac{Q_j(x) Q_i(x)}{T_{ij}(x)+\varepsilon} (y_j{}^u(x) - y_i{}^u(x)) \quad (6)$$

$$\alpha_i{}^u(x) = \frac{\varphi_i{}^u(x)}{Q_i(x)} = \sum_{j \in Lbest j \neq i} rand_j Gc(x) \frac{Q_j(x)}{T_{ij}(x)+\varepsilon} (y_j{}^u(x) - y_i{}^u(x)) \quad (7)$$

Here, $rand_j$is random numbers that are distributed uniformly in interval (0,1), $T_{ij}(x)$ is the Euclidean distance among agent i and j, $\varepsilon$is used for division by zero escape order among the agents. Lbest denotes set of L agents with best value with L indicating time sequence with respect to $L_{initial}$at begin of the model. *Gc(x)* indicate gravitational coefficient (GC) at with $Gc_{initial}$as initial value and finally ends with Gc_{end} by Eq. 8.

$$Gc(x) = Gc(Gc_{initial}, Gc_{end}, x) \quad (8)$$

Then the i-th agent's next velocity is computed as a fraction of its current velocity applied by Eq. 9 to its acceleration, and an i-th agent's next location can be determined using Eq. (10).

$$D_i{}^u(x+1) = rand \times D_i{}^u(x) + \alpha_i{}^u(x) \quad (9)$$

$$y_i{}^u(x+1) = y_i{}^u(x) + \alpha_i{}^u(x+1) \quad (10)$$

The algorithm for proposed DGSA is shown in algorithm below as Algorithm 1.

Algorithm 1: Proposed Discrete GSA for resolving VM Placement

Input: consider total no. of Virtual Machine and Physical Machine and resource constraints for every

Virtual Machine (RAM & CPU).

Initial: population is Initialized by generating functions of random permutation statistics..

Output: Best Solution

Fitt function calculation is done for all PM used

For every population gravitational mass is computed.

While till stop criteria satisfy **do**

ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 8, Issue 3, 2019

Update Gc,L, Lbest

Acceleration of all agent is calculated by using Eq. 7

Velocity of all agent is calculated by using Eq. 9

Every agent position is updated using Eq. 10

Calculate fitness for every agent.

GM calculation for all agent.

End While

In this paper, As discussed in section 3.4, based on the components of gravitational search model, it is discussed separately with proposed DGSA.

3.5.1. DGSA Independent movement operator (IMO)

The i^{th} agent location is retrieved by IMO and input by IML in the original GSA, and the agent moves with respect to solution space based on the IML input value. In a broad sense, the IMO hypothesis exemplifies an agent's confidence in its own early movement. One of the most prominent operators that may be adjusted or replaced by employing IMO operators is a modified version of the local search algorithm. The local search algorithm is a computer search method that begins with an initial solution and then progresses to a neighboring solution that improves the problem's objective function. Many approaches are followed for better neighbor selection based on (1) “Best selection strategy for enhancement in which the best neighbor is chosen” (i.e. “the neighbor who increases the target function the most), (2) “Initial selection strategy for improvement, which consists of selecting the initial neighbor that is improved than the existing solution for improvement” and, (3) “Strategy of random selection in which a random selection is extended to those Neighbors that increase in the existing solution”. The current solution enhancement procedure will continue until every candidate neighbours are worser compared to current solution. The procedure of proposed Local search Model (IMO) is as follows.

Algorithm 2: Proposed Local search Model (IMO)

ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 8, Issue 3, 2019

Input: consider x as the initial outcome, IML as movement length that is maximum, Iterations= 0;

Repeat

Produce neighbors (candidate) to x:

If no better neighbor

End of Algorithm 2:

Assigning a best neighbor of x to x;

Iterations++.

Upto the iterations are equal to IML.

3.5.2. DGSA Dependent Movement operator (DMO)

The DMO consists of two parameters in original GSA, the DML of i^{th} agent and its current position, returns a new position to the space solution as output. Based on Eq. (7) The i^{th} agent's DML is an acceleration vector which is determined by the number of accelerations with which the i^{th} agent acts on all members of Lbest. In other words, all members of Lbest are trying to attract it to compute the DML for the i^{th} agent (Eq. (7)). Defining the DMO in multi-dimensional continuous spaces is simple: moving the i^{th} agent in all dimensions of space towards all members of Lbest. Thus, by performing the IMO and DMO components the VM can be easily placed with respect to its neighbors.

Let i^{th} agent relate to beginning solution y, let the j^{th} agent relate to target solution z, and finally the i^{th} agent's DML corresponds to K_{ij} against the j^{th} agent. The value of K_{ij} is computed in the proposed DGSA by the following equation:

$$\varphi_{ij}(x) = rand \times Gc(x) \times \frac{Q_j(x)Q_i(x)}{NT_{ij}(x)} \times T_{ij}(x) \qquad (11)$$

$$\alpha_i{}^u(x) = l_{ij} = \left[\frac{\varphi_i{}^u(x)}{Q_i(x)}\right] = \left[rand_j Gc(x) \frac{Q_j(x)}{T_{ij}(x)+\varepsilon} \times T_{ij}(x)\right] \qquad (12)$$

Where, $Q_i(x)$ and $Q_j(x)$ is gravitational mass i and j for agent, *rand* is is uniformly distributed random number with interval (0,1). $T_{ij}(x)$ indicates the distance among agent i and j, and by using the Xor operator to convert i to j. $NT_i(x)$ indicates a normalization value of $T_{ij}(x)$ with intervals (0.5,1), $Gc(x)$ is gravitational coefficient with initial value (0,1) and vary with time, such that $G_{Start} \geq G_{End}$.

ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 8, Issue 3, 2019

4. Results and discussion

4.1. Experimental Setup

The suggested model's performance was tested in MATLAB v9.1, with a system specification of an Intel Core i5, 10th Gen Processor running at 3.2 GHz and 8GB RAM.

Reference Value	Distribution value
$\overline{C_{CPU}} = \overline{C_{Mem}} = 25\%$	[0,50%]
$\overline{C_{CPU}} = \overline{C_{Mem}} = 50\%$	[0,90%]

Table 1. Simulation setup values

For comparison of proposed results, Ant Colony Optimization Algorithm (ACO) [5], Least Loaded (LL) [7] and First Fit Decreasing (FFD) [6]. RAM and CPU resources are produced randomly, and the particulars are shown in Table 1.

Correlation	LB	Algorithm	PM	PM/LB
-0.759	*55*	*FFD*	*71*	*1.29*
		LL	*66*	*1.2*
		ACO	*61*	*1.2*
		DGSA	*56*	*1.1*
-0.366	*54.8*	*FFD*	*68*	*1.24*
		LL	*62*	*1.13*
		ACO	*59*	*1.07*
		DGSA	*55*	*1.00*
-0.60	*54.2*	*FFD*	*65*	*1.99*
		LL	*60*	*1.107*
		ACO	*58*	*1.070*
		DGSA	*55*	*1.014*
0.38	*54*	*FFD*	*63*	*1.666*
		LL	*59*	*1.092*
		ACO	*57*	*1.055*
		DGSA	*55*	*0.98*
0.792	*53.9*	*FFD*	*64*	*1.187*
		LL	*59*	*1.094*

ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 8, Issue 3, 2019

		ACO	*56*	*1.038*
		DGSA	*54*	*1.001*

Table 2. Result-100 instance with $\overline{C_{CPU}} = \overline{C_{Mem}} = 25\%$

4.2. Performance Analysis

Evaluations were made of the proposed model under the state of the world as described in the section above. For all the three referenced values on five separate correlation variables, the output of the algorithms is taken as per the metrics. Tabulated and depicted in graph format are the results as shown in Table 1, figure 1, table 2 and figure 2.

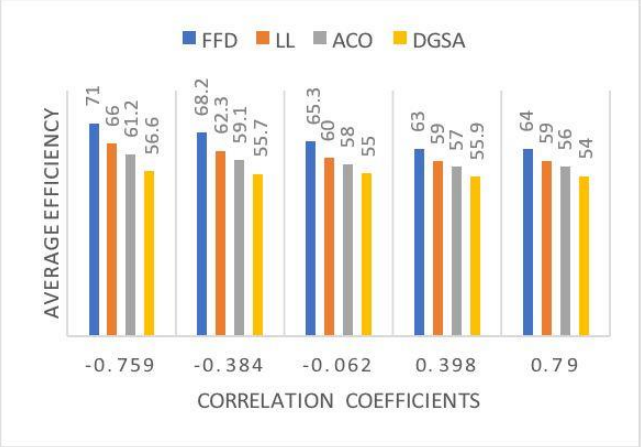

Figure 2. Result-100 instance with $\overline{C_{CPU}} = \overline{C_{Mem}} = 25\%$

The results of our DGSA, as well as other current algorithms for VM placement, are shown in Table 2. Figure 2 is a graphical representation of the results in the table. The number of PMs used to accommodate the specified VMs was minimized using Table 2.

Correlation	LB	Algorithm	PM	PM/LB
-0.764	*74.6*	*FFD*	*85*	*1.13*
		LL	*87*	*1.66*
		ACO	*79*	*1.05*
		DGSA	*76*	*1.02*
-0.352	*74.8*	*FFD*	*84*	*1.12*
		LL	*88*	*1.17*
		ACO	*77*	*1.03*

ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 8, Issue 3, 2019

		DGSA	*75*	*1.00*
-0.58	*74.8*	*FFD*	*83*	*1.11*
		LL	*89*	*1.19*
		ACO	*77*	*1.03*
		DGSA	*76*	*1.01*
0.37	*74.7*	*FFD*	*82*	*1.09*
		LL	*87*	*1.12*
		ACO	*76*	*1.01*
		DGSA	*75*	*1.00*
0.78	*74.7*	*FFD*	*81*	*1.08*
		LL	*87*	*1.16*
		ACO	*76*	*1.01*
		DGSA	*75*	*1.00*

Table 3. Result-100 instance with $\overline{C_{CPU}} = \overline{C_{Mem}} = 50\%$

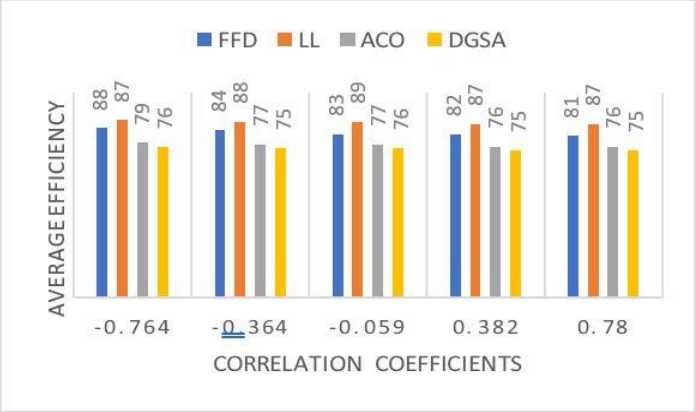

Figure 3. Result-100 instance with $\overline{C_{CPU}} = \overline{C_{Mem}} = 50\%$

The results of our DGSA, as well as other current algorithms for VM placement, are shown in Table 3. Figure 3 is a graphical representation of the results in the table. The number of PMs used to accommodate the specified VMs was reduced using Table 3.

5. Conclusion

In a cloud computing system, VM placement is one of the most prevalent elements to be minimized from a commercial standpoint. The proposed Discrete GSA model is used in this paper to successfully schedule VM to appropriate PMs with a smaller number of PMs. To test

ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 8, Issue 3, 2019

the used approach on VM placement, it was implemented in MATLAB, and the results were compared to existing techniques in the literature that solved VM placement. When comparing the suggested model's results to those of existing models, the proposed method demonstrates a considerable improvement in terms of performance. This research can help future analysis that could be improved by incorporating multi-objectives for a more effective solution to VM placement.

References

[1] Randles, M., Lamb, D., Odat, E., &Taleb-Bendiab,A. (2011). Distributed redundancy and robustness in complex systems. Journal of Computer and System Sciences, 77(2), 293-304.

[2] Hyser, C., McKee, B., Gardner, R., & Watson, B.J. (2007). Autonomic virtual machine placement in the data center. Hewlett Packard Laboratories, Tech. Rep. HPL-2007-189, 189.

[3] Rashedi, E., Nezamabadi-Pour, H., &Saryazdi, S. Rashedi, E., Nezamabadi-Pour, H., &Saryazdi, S.(2009). GSA: a gravitational search algorithm.Information sciences, 179(13), 2232-2248.

[4] Dowlatshahi, M. B., Nezamabadi-Pour, H., &Mashinchi, M. (2014). A discrete gravitational search algorithm for solving combinatorial optimization problems. Information Sciences, 258, 94-107.

[5] Feller, Eugen, Louis Rilling, and Christine Morin. "Energy-aware ant colony based workload placement in clouds." Proceedings of the 2011 IEEE/ACM 12th International Conference on Grid Computing. IEEE Computer Society, 2011.

[6] Lodi, Andrea, Silvano Martello, and Daniele Vigo. "Recent advances on two-dimensional bin packing problems." Discrete Applied Mathematics 123.1 (2002): 379-396.

[7] Ajiro, Yasuhiro, and Atsuhiro Tanaka. "Improving packing algorithms for server consolidation." Int. CMG Conference. Vol. 253. 2007.

[8] Li, X., Qian, Z., Lu, S., & Wu, J. (2013). Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in adata center. Mathematical and Computer Modelling, 58(5), 1222-1235.

[9] Liu, C., Shen, C., Li, S., & Wang, S. (2014, June). A new evolutionary multi-objective

ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 8, Issue 3, 2019

algorithm to virtual machine placement in virtualized data center. In Software Engineering and Service Science (ICSESS), 2014 5th IEEE International Conference on (pp. 272-275). IEEE.

[10] Wang, S., Gu, H., & Wu, G. (2013, July). A new approach to multi-objective virtual machine placement in virtualized data center. In Networking, Architecture and Storage (NAS), 2013 IEEE Eighth International Conference on (pp. 331-335). IEEE.

[11] Yang, T., Lee, Y. C., &Zomaya, A. Y. (2014, December). Energy-efficient data center networks planning with virtual machine placement and traffic configuration. In Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on (pp. 284-291). IEEE.

[12] Jamali, S., &Malektaji, S. (2014, October). Improving grouping genetic algorithm for virtual machine placement in cloud data centers. In Computer and Knowledge Engineering (ICCKE), 2014 4th International eConference on (pp. 328- 333). IEEE.

[13] Dashti, S. E., &Rahmani, A. M. (2016). Dynamic VMs placement for energy efficiency by PSO in cloud computing. Journal of Experimental & Theoretical Artificial Intelligence, 28(1-2), 97-112.

[14] Wang, S. H., Huang, P. P. W., Wen, C. H. P., & Wang, L. C. (2014, February). EQVMP: Energy- efficient and QoS-aware virtual machine placement for software defined datacenter networks. In Information Networking (ICOIN), 2014 International Conference on (pp. 220-225). IEEE.

[15] Gao, Y., Guan, H., Qi, Z., Hou, Y., & Liu, L. (2013). A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. Journal of Computer and System Sciences, 79(8), 1230-1242.

[16] DONG, J. K., WANG, H. B., LI, Y. Y., & CHENG,S. D. (2014). Virtual machine placement optimizing to improve network performance in cloud data centers. The Journal of China Universities of Posts and Telecommunications, 21(3), 62-70.