JCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol. 9, Issue.10, October 2021 E-ISSN: 2347-2693

Virtual Machine Placement using Interactive Artificial Bee Colony Algorithm(VMPIABC)

Shubham Kumar[1], Atul Tripathi[2*]

[1,2]Dept. of Computer Science & IT, Mahatma Gandhi Central University, Motihari, India

[*]*Corresponding Author: atultripathi@mgcub.ac.in, Tel.: 9968798126*

DOI: https://doi.org/10.26438/ijcse/v9i10.16 | Available online at: www.ijcseonline.org

Received: 03/Oct/2021, Accepted: 19/Oct/2021, Published: 31/Oct/2021

Abstract— As cloud computing is becoming part of our life day by day, it has attracted research community to tackle the research problems of cloud computing environment. Virtual machine placement is a brewing area for cloud researchers so in the proposed model virtual machine placement problem is modelled as an optimization problem with the objective of resource wastage. As huge resource wastage can affect the cloud service provider so, an virtual machine placement algorithm based on interactive artificial bee colony was proposed. The performance of the proposed method is thoroughly compared with other competing algorithms through exhaustive experiments and results are presented.

Keywords— Virtual Machine, Cloud Computing, Artificial Bee Colony, Resource Wastage, Optimization

I. INTRODUCTION

In today's world, cloud computing is playing a vital role in providing online services. Cloud services are basically of three types Application as Service (SaaS), Platform as Service (PaaS), and Infrastructure as Service (IaaS). Based on the requirements of the individual or organization, the user takes the services from the cloud service provider on a pay and use basis. Users can get these services through an internet connection. Cloud providers use data centers through which they provide services over the internet. Data centers are the house of many servers and storage. Many concepts are used in the cloud, like Grid computing, Parallel computing, virtualization technology, etc. Among all these, server virtualization is the key concept through which the proper utilization of resources is possible in the cloud. The idea of virtual machines enhanced the technology such that on a single server, many users can run their tasks as they run on an independent machine. The user tasks are to be executed on the respective virtual machines. For this, firstly, a set of tasks must be mapped with the set of virtual machines[1]. Based on the requirements of tasks, virtual machines are created, and then these virtual machines are required to be mapped with the available servers in the cloud data center. One server may have many virtual machines, but one VM cannot run on more than one server. VM should be placed on the server which is best suitable for it in terms of proper utilization of resources, task execution time. In figure 1, virtualization technology places some VMs on a single server with the help of Hypervisor, i.e., VMM. In the case of multiple servers, various virtual machines need to be mapped with set of servers based on the resource requirements of tasks. Figure 2 shows the mapping of different VMs on multiple servers. The virtual machine placement problem is an NP-hard problem[2], so there is no optimal solution. Many researchers have tried to place VMs optimally in terms of resource utilization. There are many issues with the placement of virtual machines, like resource utilization, power consumption of data centers, Service-Level Agreement (SLA) violation, and Fault tolerance. An effective VM placement method must consider all of the mentioned features. In this paper, we aim to design a strategy that places VMs with efficient resource utilization.

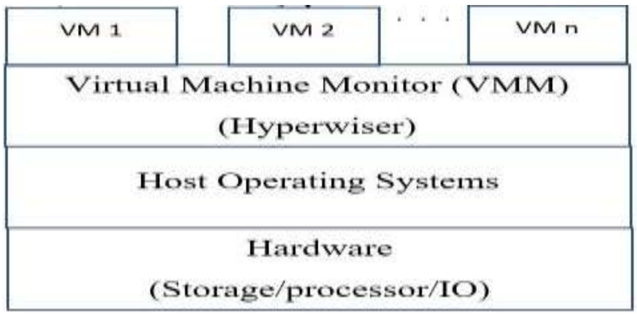

Figure 1: Virtual machines on a single server

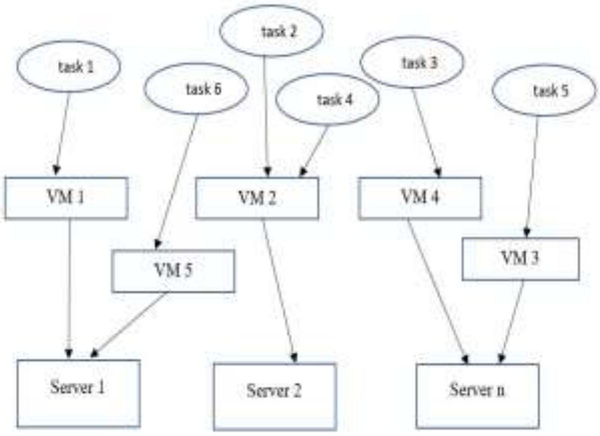

Figure 2: Virtual machine placement on multiple servers

Even though there are various literature available in the research domain, very few of them can utilize the resources efficiently. In the virtual machine placement problem, resource wastage at the cloud data center is a key challenge for both cloud service providers and research communities. If resources of data centers are not efficiently utilized, it will lead to high operational costs to cloud service providers. Therefore, in the proposed work, the virtual machine placement problem is modeled as an optimization problem having resource wastage as an objective. A bio-inspired algorithm, namely Interactive Artificial Bee Colony [3], is applied to place various VMs over the physical server to utilize cloud data center resources efficiently. The manuscript is further organized as follows: Section 2 consist of the literature survey of recent similar problems. Section 3 deals with the background of the proposed VM placement algorithm that consist of formulation, method elaboration etc. Section 4 presents the actual VM placement algorithm which is followed by section 5 which consist of experimental analysis. Finally, section 6 consists of conclusion and future work as well.

II. Related Work

In this section, we have discussed different VM placement strategies that are presented in the literature. Rajkumar Buyya et al.[3] proposed a greedy randomized algorithm for VM placement in large-scale Cloud Data Center Networks (CDCNs). The algorithm's outcome shows that the performance is improved by above 15% regarding the number of activated PMs. Still unable to adequately utilize the resources as the percentage of improvement is significant in terms of wastage of resources. According to Wei Zhang et al.[4] For reliable cloud computing systems, VM placement strategies should consider five factors: SLA violation rate, resource remaining rate, power consumption rate, failure rate, and fault tolerance cost. So, he proposed a heuristic ant colony algorithm to solve the multi-objective optimization model. The fault tolerance issue is the main objective of this algorithm that only considers the single VM failure. Xiong Fu et al.[5] proposed a predicted affinity-based virtual machine placement model. This model explores the relationships between every two virtual machines based on the resource requirements provided by Auto-Regressive Integrated Moving Average (ARIMA) prediction. After putting two VMs on the same host, the proposed model evaluates the volatility of resource utilization. They called this model as affinity model. Based on the highest affinity of the hosts, VMs are placed. Mohamed Ghetas[6] proposed a Multi-objective Monarch Butterfly Algorithm for VM placement in a cloud environment. It is mainly designed to reduce the number of active physical servers and maximize packaging efficiency. Performance is evaluated of the MBO-VM with real workloads and synthetic workloads. No doubt the proposed method is more energy-efficient, but maximizing resource utilization is still the issue. Sasan Gharehpasha et al.[7] proposed a hybrid multi-multiverse optimization algorithm to reduce the power consumption in the data center by minimizing active PMs and also reduces the wastage of resources. The approach combines the hybrid discrete multi-object whale optimization algorithm and multi-verse optimizer with chaotic functions. This outcome is satisfactory as it can minimize the number of active servers in the data center and hence reduce energy consumption. Dabiah Alboaneen et al.[8] presented a metaheuristic method for joint task scheduling and virtual machine placement in the data centers. The proposed model aims to schedule tasks on the VM with the least execution cost and then place the selected VM on the most utilized physical server. The algorithm is compared with glowworm swarm optimization (GSO) and moth-flame glowworm swarm optimization (MFGSO), and it is found that the applied algorithm has higher resources utilization. Binnin Zhang et al.[9] proposed a cluster-based genetic algorithm that uses run-time features to evaluate the preference of VMs on hardware resources. The algorithm usage the K-means approach to cluster the population. The proposed algorithm is compared with the basic genetic algorithm and finds the optimum result. Shalu et al.[10] proposed an artificial neural network-based virtual machine allocation method. The algorithm is capable of finding false allocation of VMs that occurs due to inefficient utilization of resources. Compared with some other techniques, SLA violations are slightly low, but the efficient utilization of resources is still the issue. K.M. Baalamurugan et al.[11] proposed a multi-objective krill herd algorithm for utilized VM placement. The proposed method was compared with First Fit Decreasing (FFD) and Simplified Ant Colony (SACO) algorithms and found more efficient, but the utilization has not matched the mark. A method named Optimistic VM placement using queuing approach was proposed by the author Anitha Ponraj[12] that reduces the processing cost and completion time by more than 90% compared with some traditional methods like FCFS and Priority scheduling. Wenbin Yao et al.[13] proposed a weighted page rank-based method to place VM in the cloud environment. The target is to achieve the proper utilization of resources and minimizing the number of active PMs. First, the possibilities of a PM makes full use of resources under different VMP conditions are measured. Then the algorithm ranks PM and places the Virtual machines based on ranking. The author compared the proposed algorithm with the page-rank VM algorithm, FFD algorithm, and Random algorithm. The experimental result shows that the Weighted page rank-based algorithm is the best among those. A novel adaptive energy-aware Virtual Machine allocation and deployment model AFED-EF has been proposed by authors Zhou Zhou et al.[14]. The algorithm efficiently handles the load fluctuations and gives good performance during VM allocation and placement. The AFED-EF algorithm gives better result in terms of SLA violation rate, and energy consumption than other energy-aware algorithms.

III. Methodology

3.1 Artificial Bee Colony Algorithm

The Artificial Bee Colony algorithm is one of the most powerful heuristic techniques, which was introduced by

Dervis Karaboga[15] in 2007. This iterative algorithm was based on the swarm behavior of honey bees. The colony of bees consists of three kinds of bees, namely employer bees, onlooker bees, and scout bees. In the ABC algorithm, these bees represent the possible solution to the optimization problem. The entire population is divided into two halves. The first half consists of all employer bees, and the second half consists of onlooker bees. In each iteration, the employer bees change source position in the neighborhood to determine a new source. The nectar value of the new source position is considered a fitness function. If the nectar value of the new source position is greater than the previous position, then the employee bee's position is updated to the new source position. When all employee bees had performed this search operation, the nectar value of each source is shared with the onlooker bees through dance. Each onlooker bee selects food as a new food source which will be transformed into scout bees. Note that each food source is mapped with a single bee; therefore, half bees are employee bees in a colony, and half are onlookers. The operating procedure of ABC comprises the following steps:

3.1.1 Initialization

Generate n_e percentage population randomly, which is the ratio of employed bees to the total population. This population will be called as an employed bee. Also, compute the nectar amount of each employed bee.

3.1.2 Onlooker bee movement

The probability of selecting a food source by an onlooker bee is determined by equation (6). Then nectar amount of each onlooker bee is calculated. The roulette wheel selection method is used for the movement of onlooker bees as in equation (7).

$$\chi_i = \frac{\rho(\theta_i)}{\sum_{k=1}^{k=s} \rho(\theta_k)} \quad (6)$$

Position of i^{th} employee bee is represented by θ_i , and total no of scout bee is s and χ_i represents the probability of selecting i^{th} employee bee.

$$\sigma_{ij}(t+1) = \theta_{ij} + \Psi(\theta_{ij}(t) - \theta_{kj}(t)) \quad (7)$$

Here coordinate of i^{th} onlooker bee is represented by σ_i at the total iteration count t, and an employed bee which is selected randomly is represented by θ_k. Ψ is a random number generated between [-1, 1]

3.1.3 Scout bee movement

If, after the predetermined number of steps, further improvement in the nectar value of the food source is not possible, then it is assumed that the food source is abandoned. Then employee bee corresponding to this abandoned food source becomes the scout bee which will move according to equation (8).

$$\theta_{ij} = \theta_{ij\min} + \phi.(\theta_{ij\max} - \theta_{kj\min}) \quad (8)$$

Here ϕ is a random number generated between [0, 1], and the dimension of the solution is represented by j.

3.1.4 Updation of best food source

Position of food source corresponding to best fitness value is memorized along with the fitness value.

3.1.5 Checking the termination condition

All the above steps are repeated till the termination condition is false and the result is noted. Go to step 2 if the termination condition is not satisfied yet.

3.2 The Interactive Artificial Bee Colony Algorithm

In the standard ABC algorithm, the movement of the onlooker bee is only dependent on the relationship between employee bees and is operated based on the roulette wheel selection method. Since this method is based on random selection, therefore exploitation capability of the ABC is quite limited. Thus, an enhancement in ABC, i.e., interactive artificial bee colony[16], [17], was proposed, inspired by the Newtonian law of gravitation force. The universal gravitational force between onlooker bee and employee bee is represented as in equation 9.

$$F_{12} = G\frac{m_1 m_2}{r_{21}^2}\hat{r}_{21} \quad (9)$$

Here F_{12} represents the gravitational force between object_1 and object_2, G represents the universal gravitational constant, m_1 and m_2 are their respective masses. r_{21} represents the distance between object_1 and object_2 and $\hat{r}_{21}$ represents the unit vector as in equation 10.

$$\hat{r}_{21} = \frac{r_2 - r_1}{|r_2 - r_1|} \quad (10)$$

In interactive ABC, equation 9 is remodified as in equation 11.

$$F_{ik_j} = G\frac{\rho(\theta_i)\rho(\theta_k)}{(\theta_{kj} - \theta_{ij})^2} \cdot \frac{\theta_{kj} - \theta_{ij}}{|\theta_{kj} - \theta_{ij}|} \quad (11)$$

And consequently, equation 7 can be modified as

$$\sigma_{ij}(t+1) = \theta_{ij}(t+1) + F_{ik_j}.[\theta_{ij}(t) - \theta_{kj}(t)] \quad (12)$$

IV. PROPOSED VM PLACEMENT ALGORITHM BASED ON IABC

In the proposed VMPIABC algorithm, a colony of bees of swarm Size S_L is generated randomly, and each solution S_h is considered as a vector of d dimension. The search procedure performed by an employed bee, onlooker bee, and scout bee is repeated till *max*. This search procedure will generate a better solution than the previous one in terms of nector value. The nector value is represented by the fitness function, which is carefully designed as in equation 13.

$$fit\ \tau(b)=\sum_{j=1}^{m}\left[y_j\times\frac{\left|\left(R_{c_j}-\sum_{i=1}^{n}(x_{ij}\cdot\rho_{c_i})\right)-\left(R_{m_j}-\sum_{i=1}^{n}(x_{ij}\cdot\rho_{m_i})\right)\right|+\varepsilon}{\sum_{i=1}^{n}(x_{ij}\cdot\rho_{c_i})+\sum_{i=1}^{n}(x_{ij}\cdot\rho_{m_i})}\right]+\sum_{j=1}^{m}\alpha_j\max\left(0,\sum_{i=1}^{n}\rho_{c_i}\cdot x_{ij}-R_{c_j}\cdot y_j\right) \tag{13}$$

Here bee, which is a solution, is represented as b and $\sigma(b)$ represents the fitness value of b, and α_j represents the penalty parameter for j^{th} physical machine when its capacity is full.

Each VM request is selected probabilistically through the roulette wheel selection mechanism using probability Ph of choosing a solution Sh, represented in equation 14.

$$P_h=\frac{fit\ \tau(h)}{\sum_{k=1}^{k=S_L} fit\ \tau(k)} \tag{14}$$

4.1 Solution improvisation
At each iteration, a d dimensional solution is improved using the equation 15

$$New\ S_{hj}=S_{hj}+F_{hi_j}(S_{hj}-S_{ij}) \tag{15}$$

4.2 Solution updation rule
In order to search for new food source, the scout bee uses equation 16 to find the new food source.

$$S_h^j=S_{\min}^j+rand[0,\ 1](S_{\max}^j-S_{\min}^j)$$
$$\text{for } j=1,2,\ldots\ldots\ldots,d \tag{16}$$

4.3 Dynamic virtual machine mapping algorithm based on IABC
In this section VMPIABC is formally discussed.

Algorithm : VMPIABC

Input:

Initialize the set of n VMs and m PMs randomly

Initialize the initial bee colony S_k

Initialize the max and c=1

Steps

Repeat the following steps while(c<=max)

1. *Generate the new population of employee bee New S_k using equation 15.*
2. *Use greedy selection method between S_k and New S_k*
3. *Calculate the probability of selection P_k of S_k using equation 14.*
4. *Generate the new population of onlooker bees based on P_k*
5. *Apply the greedy selection method of onlooker bees*
6. *Replace the abandoned food source by replacing the S_k with new S_k using equation 16*
7. *Memorize the best solution*
8. *C=C+1*
9. *Record the solution*

V. Results and Discussion

The performance of the proposed virtual machine placement algorithm based on interactive artificial bee colony is compared with the other two competing algorithms, which are based on artificial bee colony (VMPABC)[15] and particle swarm optimization (VMPPSO)[18]. The entire model has been simulated using MATLAB, and exhaustive experiments have been performed. For the experiment work size of the bee colony is taken as 50, and a maximum number of count *max* is taken as 100. The data center size ranges between 50-100 PMs, and generated VMs range between 30-80. CPU and Memory requirements of VMs are generated randomly based on uniform distribution, and their arrival rate is as per the poisson process with an average rate of 4 VMs per 100-time units, and each one has an exponentially distributed lifetime with an average of 500-time units. Value of parameter $\alpha_j = 1$ and for VMPPSO, the inertia weight *w* is set to 0.8 and c1=c2= 1.0. The simulation was run for 10000 units and their average results are reported.

5.1 Resource wastage comparison
Resource wastage for three competing algorithms, i.e., VMPIABC, VMPABC, and VMPPSO, is compared at various iterations and their corresponding results are as shown in figure 3. Here, it can be clearly observed that the resource wastage in VMPIABC is slightly less than VMPABC and much lower than the VMPPSO algorithm.

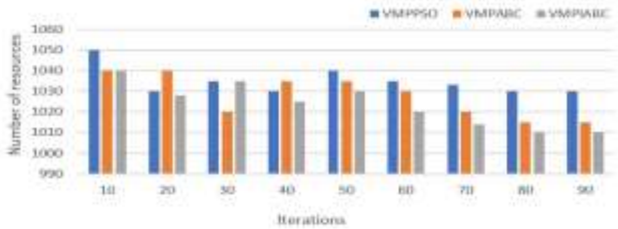

Figure 3: Resource wastage in different iterations

5.2 Acceptance ratio comparison

At different time instances, accept ratios of VMPIABC, VMPABC, and VMPPSO are compared, as shown in figure 4. It is evident from the plot that VMPIABC is performing better as it has a higher acceptance ratio than the other two. It is also evident that VMPABC is better than VMPPSO.

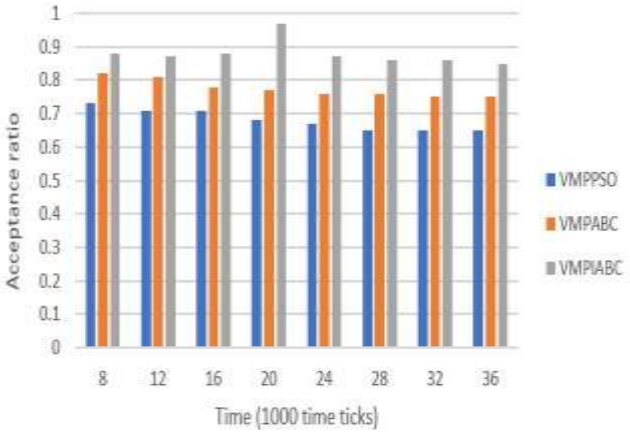

Figure 4: Acceptance ratio

5.3 Placement time comparison

Finally, the placement time of these three virtual machine placement algorithms is compared for 20-80 VMs at different iteration counts and represented in figure 5. It is overserved VMPIABC is taking less time to place the various virtual machines than the rest of the two.

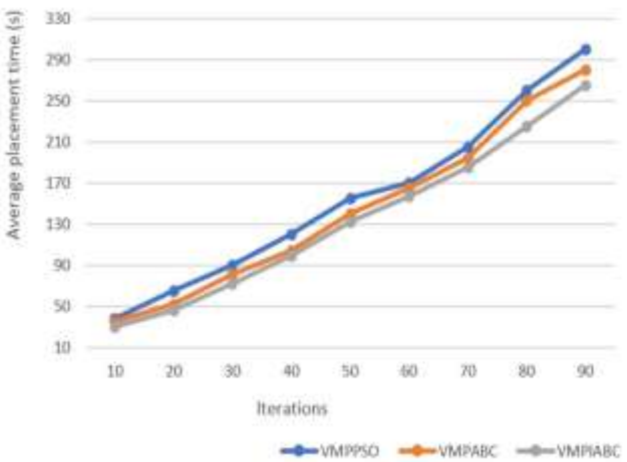

Figure 5: Average placement time at different iterations

VI. Conclusion and Future Scope

In this article, we are trying to solve the VM placement problem with minimum wastage of resources. The Interactive Artificial Bee Colony algorithm is used for this purpose. The algorithm is compared with competing two algorithms, VMPABC and VMPPSO, that are based on the artificial bee colony approach and particle swarm optimization technique, respectively. We have used MATLAB for this comparison. The comparison result clearly shows that our proposed VMPIABC method is efficient in resource utilization, acceptance ratio, and average placement time. Here we used this algorithm for solving the single-objective problem and found the result a little bit better. So, our next target is to use this approach in solving multi-objective VM placement problems where one of the objectives is to solve the Fault tolerance issues.

References

[1] A. Tripathi, "Task Allocation on Cloud Resources using Analytic Network Process," **pp. 971–978, 2015.**

[2] A. Tripathi, I. Pathak, and D. Prakash, "Modified Dragonfly Algorithm for Optimal Virtual Machine Placement in Cloud Computing," **pp. 1316-1342, 2020**

[3] S. Azizi, M. Shojafar, and S. Member, "GRVMP : A Greedy Randomized Algorithm for Virtual Machine Placement in Cloud Data Centers," **pp. 1–12, 2020.**

[4] W. Zhang, X. Chen, and J. Jiang, "A Multi-Objective Optimization Method of Initial Virtual Machine Fault-Tolerant Placement for Star Topological Data Centers of Cloud Systems," **vol. 26, no. 1, pp. 95–111, 2021.**

[5] X. Fu and C. Zhou, "Predicted Affinity Based Virtual Machine Placement in Cloud Computing Environments," *IEEE Trans. Cloud Comput.*, **vol. 8, no. 1, pp. 246–255, 2020.**

[6] M. Ghetas, "A multi-objective Monarch Butterfly Algorithm for virtual machine placement in cloud computing," *Neural Comput. Appl.*, **vol. 33, no. 17, pp. 11011–11025, 2021.**

[7] S. Gharehpasha, M. Masdari, and A. Jafarian, "*Virtual machine placement in cloud data centers using a hybrid multi-verse optimization algorithm*," **vol. 54, no. 3. Springer Netherlands, 2021.**

[8] D. Alboaneen, H. Tianfield, Y. Zhang, and B. Pranggono, "A metaheuristic method for joint task scheduling and virtual machine placement in cloud data centers," *Futur. Gener. Comput. Syst.*, **vol. 115, pp. 201–212, 2021.**

[9] B. Zhang, X. Wang, and H. Wang, "Virtual machine placement strategy using cluster-based genetic algorithm," *Neurocomputing*, **vol. 428, pp. 310–316, 2021.**

[10] Shalu and D. Singh, "Artificial neural network-based virtual machine allocation in cloud computing," **pp. 1-12, 2021.**

[11] K. M. B. S. V. Bhanu, "A multi-objective krill herd algorithm for virtual machine placement in cloud computing," **vol. 76, no. 6, pp. 4525-4542, 2018.**

[12] A. Ponraj, "Optimistic Virtual Machine Placement in Cloud Data Centers using Queuing Approach," **vol. 93, pp. 338-344, 2018.**

[13] W. Yao, Y. Shen, and D. Wang, "A Weighted PageRank-Based Algorithm for Virtual Machine Placement in Cloud Computing," *IEEE Access*, **vol. 7, pp. 176369–176381, 2019.**

[14] Z. Zhou, M. Shojafar, M. Alazab, J. Abawajy, and F. Li, "AFED-EF: An Energy-Efficient VM Allocation Algorithm for IoT Applications in a Cloud Data Center," *IEEE Trans. Green Commun. Netw.*, **vol. 5, no. 2, pp. 658–669, 2021.**

[15] X. Yu, W. Chen, and X. Zhang, "An Artificial Bee Colony Algorithm for Solving Constrained Optimization Problems," *Proc. 2018 2nd IEEE Adv. Inf. Manag. Commun. Electron. Autom. Control Conf. IMCEC 2018*, **pp. 2663–2666, 2018.**

[16] I. Pathak and D. P. Vidyarthi, "An Interactive Artificial Bee Colony based Virtual Network Embedding," **pp. 1-6, 2015.**

[17] P. W. Tsai, M. K. Khan, J. S. Pan, and B. Y. Liao, "Interactive artificial bee colony supported passive continuous authentication system," *IEEE Syst. J.*, **vol. 8, no. 2, pp. 395–405, 2014.**

[18] A. Tripathi, I. Pathak, and D. P. Vidyarthi, "Energy Efficient VM Placement for Effective Resource Utilization using Modified Binary PSO," *Comput. J.*, **vol. 61, no. 6, pp. 832–846, 2018.**

AUTHORS PROFILE

Mr. Shubham Kumar pursed M.Tech from National Institute of Technology Durgapur, India in 2016. He is currently pursuing Ph.D. and currently working as Assistant Professor in Department of Computer Science & IT, MGCU Bihar, India since 2016.

Dr. Atul Tripathi pursed M.Tech and Ph.D. from Jawaharlal Nehru University, India. He is currently working as Assistant Professor in Department of Computer Science & IT, MGCU Bihar, India since 2016

