

Resource-Efficient VM Placement in the Cloud Environment Using Improved Particle Swarm Optimization

Bhagyalakshmi Magotra, Central University of Jammu, India*

Deepti Malhotra, Central University of Jammu, India

ABSTRACT

Fundamentally, a strategy considering the effective utilization of resources results in the better energy efficiency of the system. The aroused interest of users in cloud computing has led to an increased power consumption making the network operation costly. The frequent requests from the users asking for computing resources can lead to instability in the load of the computing system. To perform the load balancing in the host, migration of the virtual machines from the overloaded and underloaded hosts needs to be done, which is considered an important facet concerning energy consumption. The proposed particle swarm optimization-based resource-aware VM placement (RAPSO_VMP) scheme aims to place the migrated virtual machines. RAPSO_VMP takes into consideration multiple resources like CPU, storage, and memory while trying to optimize the overall resource utilization of the system. According to the simulation analysis, the proposed RAPSO_VMP scheme shows an improvement of 5.51% in energy consumption, reduced the number of migrations by 9.12%, and the number of hosts shutdowns 22.74%.

KEYWORDS

Binary Particle Swarm Optimization, Bio-Inspired, Cloud Computing, Energy Efficiency, Euclidean Distance, Meta-Heuristic, Resource Utilization, VM Placement

INTRODUCTION

There is a great pace escalation in the introduction of newer technologies in the modern era of computing. One such technology is cloud computing which has been able to attract both, the IT community and the research society in the last decade. The extensive power, high speed processors and the enormous increase in data storage capability have raised the interests of many researchers and motivated them to share the resources on the network. This has led to the emergence of cloud computing. In cloud computing, the resources are provided to multiple users on the on demand and sharing basis. Various services provide by Cloud computing include Infrastructure as a Service (IaaS), Software as a Service (SaaS) and Platform as a Service (PaaS) (Beloglazov et al., 2012). In order to fulfil the demands of the end users, various resources are provided to them in the form of storage, processing and network. This demand has increased exponentially in past years and thus has given rise to the concept of virtualization. Virtualization enables a physical machine host multiple virtual machines (VMs) each having its own operating system so as to optimally

DOI: 10.4018/IJAMC.298312

*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

utilise the available resources. The concept of Virtualization enables the end users and the service providers to have an efficient utilization of cloud resources with optimum usage and least cost. This virtualization technique is responsible for effectively handling the increasing need of the users in terms of needed resources in Cloud Data Centres (CDC). Various objectives like balancing of load, energy management, consolidation, the sharing of users among multiple users, making the system fault free, can be achieved with the help of virtualization (Noshay et al., 2018). However, the aroused interest of the users in cloud computing has led to a tremendous growth of demand for various cloud resources, making energy consumption a critical issue. The energy demand in hyper scale data centers has increased from 31.11 terawatt hours in 2015 to 76.23 in 2020 and is expected to reach 86.58 by the end of 2021 (*Energy demand data centers globally by type 2021*, 2021). It is predicted that 90% of organizations will have personal data on IT systems they don't own or control and the information technology (IT) sector will consume up to 13% of global electricity by 2030 which is at present 7% (Gartner Inc., 2013). COVID-19 has also resulted in the acceleration of digital businesses, depending upon ruling technologies like cloud computing. According to the recent report by Gartner, the spending on remote working during pandemic will increase by 4.9% in the year 2021 (Costello & Rimol, 2021). This will, in turn, increase the energy consumption in the data centres due to increased workload. One of the main reasons of energy consumption in data centres is the inefficient resource utilization. With lower resource utilization, the energy efficiency of the system will also be low. Also, the number of active hosts will increase, leading to more cooling employments. According to a study by Srikantaiah, optimal resource utilization leads to minimum energy consumption (Srikantaiah et al., 2008). Dynamic VM consolidation has proved to be one of the magical solutions for reducing energy consumption by improving utilization of resources. VM consolidation is best achieved with the help of live VM migration. To minimize the number of the active servers and to save the energy consumption, migration of VMs from overloaded/underloaded servers takes place. Therefore, the process of consolidation comprises of a) finding overloaded/ underloaded servers b) selecting a VM from the server to be migrated c) finding a suitable for the migration of the selected VM, also called the VM Placement. Various researches have been done to carry out the process of consolidation. VM placement has attracted many academic researchers since it is considered an important issue for efficient VM consolidation. Since VM placement is an NP-hard problem (Békési et al., 2000), finding a deterministic solution to this problem is quite difficult. Various heuristic and meta-heuristic VM placement techniques, proposed by different researchers, to solve the problem are discussed in the latter part of the background work. Though it is easy and quick to implement the heuristic techniques, they may fall into local optima. Metaheuristic techniques have been proved to be able to find near optimal solutions to such NP-hard problems (Donyagard Vahed et al., 2019). They have been used in several kinds of researches (Bangyal, Ahmad, et al., 2019) (Pervaiz et al., 2021) to show their importance. In this paper, a Particle Swarm Optimization based VM Placement technique, RAPSO_VMP (Resource Aware Particle Swarm Optimization), has been proposed. The robustness of the PSO algorithm towards the control parameters makes its implementation easy with a high convergence rate. The proposed scheme considers multiple resources like CPU, storage, and memory while trying to optimize many factors like energy efficiency, power consumption, and the overall resource utilization of the system. The contributions of the proposed scheme are as follows:

1. VM placement has been modelled as an optimization problem to improve the energy efficiency of the system.
2. CPU, storage and memory utilizations have been considered to formulate the objective function.
3. Binary Particle Swarm Optimization with mirrored S shaped transfer function has been used to find a near optimal solution.
4. To avoid falling into local optima, the transfer function is further modified as time varying.
5. Simulations have been carried out in CloudSim.
6. Results show the superiority of the proposed scheme over already existing LRMMT scheme.

BACKGROUND

VM Placement

VM placement can be considered as a bin packing NP hard problem whose main objective is to accommodate the migrated VMs into minimum number of physical machines. The physical machines are treated as bins that can house the VMs with the aim of optimizing the usage of the available resources such that the system works efficiently and effectively. The VM placement problem can be categorised into two categories. One is the initial placement problem while another is the runtime placement that occurs during the consolidation process.

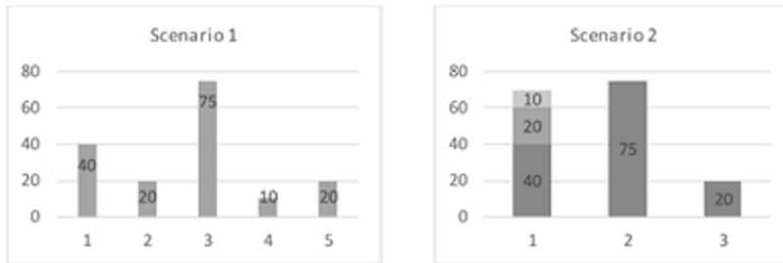
The first problem occurs when the VMs are to be allocated to the hosts for the first time. Assuming there are n VMs that need to be created. Instead of placing these n VMs in n hosts out of m available hosts where, $n < m$, it is beneficial to accommodate these VMs into lesser number of hosts such that lesser number of hosts remain in active mode, while fulfilling the criteria mentioned in equation 1.

$$\left(\sum_{i=1}^k CPU_i < CPU_j \right) \wedge \left(\sum_{i=1}^k RAM_i < RAM_j \right) \wedge \left(\sum_{i=1}^k BW_i < BW_j \right) \quad (1)$$

Where, CPU_j , RAM_j , BW_j = resource capacity of Physical machine j k = number of VMs currently allocated to the Physical machine j .

An example of VM placement is shown in Figure 1. The objective is to place 5 VMs machines with resource requirements of 40%, 20%, 75%, 10% and 20% respectively in a manner that optimizes the performance of a data center.

Figure 1. Example of VM Placement



In Scenario1 each VM is placed on a single host. This keeps 5 hosts in active mode. Whereas, in Scenario 2, instead of placing the VM, one on each host, they have been placed in a way that reduces the number of active hosts and therefore, the energy consumption. Scenario 1 makes the system extremely inefficient as the resources of the hosts remain underutilized.

The second type of VM placement occurs when the VMs are to be migrated during the process of consolidation.

$$A = \{VM_1, VM_2, VM_3, \dots, VM_n\}$$

$$B = \{PM_1, PM_2, PM_3, \dots, PM_m\}$$

Where, A = Set of n VMs to be migrated. B = Set of m available physical machines.

The objective is to place the migrated VMs to the respective physical machine according to equation 2.

$$\left(\sum_{i=1}^k CPU_i + CPU_{mig} < CPU_j \right) \wedge \left(\sum_{i=1}^k RAM_i + RAM_{mig} < RAM_j \right) \wedge \left(\sum_{i=1}^k BW_i + BW_{mig} < BW_j \right) \quad (2)$$

Where, CPU_{mig} , RAM_{mig} , BW_{mig} = resources required by the VM that is to be migrated.

$$\Upsilon_{ab} = \begin{cases} 1, & \text{if } VM a \in A \text{ is hosted on } b \in B \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Where, Υ_{ab} = VM-Host mapping variable.

The value of the variable is equal to 1 if VM “a” is allocated to host “b”, and zero otherwise. Also, each VM “a” can be allocated to a single host only such that

$$\sum_{b=1}^m \Upsilon_{ab} = 1 \quad \forall a \in A \text{ and } b \in B \quad (4)$$

Where, the $\sum_{b=1}^m \Upsilon_{ab}$ = summation of all Υ_{ab} considering the mapping between the VMs and the hosts.

The proposed scheme in the current research focusses on the second type of placement where VMs are migrated as a part of the consolidation process.

Particle Swarm Optimization(PSO)

PSO is a meta heuristic optimization technique designed for continuous search space by Eberhart and Kennedy in 1995 (Kennedy & Eberhart, 1995). The technique is based on Swarm intelligent and tries to mimic the behaviour or flock of birds and fishing school. They live in groups and interact with each other that helps them in achieving some tasks. Initially each one in the group is scattered and search for the food. The members of the group communicate with one another and update their direction according to the group member who is nearest to the food. Similarly, in PSO, the system is initialized with a population of certain particles each having a position and a velocity. The position of any particle i in D dimensions is represented as $X_i = x_{i1}, x_{i2}, x_{i3}, \dots, x_{id}, \dots, x_{iD}$, $x_{id} \in R$, each having a velocity $V_i = v_{i1}, v_{i2}, v_{i3}, \dots, v_{id}, \dots, v_{iD}$, $v_{id} \in R$. An iterative process is carried out in order to update these positions and velocities. The velocities of the particles are updated based on the interaction between them and previous information (equation 5). Whereas, positions change in accordance to the velocity. (equation 6) Each particle memorises its best solution it has attained so far, called the pbest (personal best). And also the best position achieved by whole

of the swarm, known as the gbest (global best). The outcome of the iterative process is a near optimal solution.

$$V_{t+1} = \left[(w \times v_t) + r_1 c_1 (pBestPosition - X_t) + r_2 c_2 (gBestPosition - X_t) \right] \quad (5)$$

$$X_{t+1} = X_t + V_{t+1} \quad (6)$$

Where, V_{t+1} = velocity of the particle, X_{t+1} = next position of the particle, w = inertia factor in the range (0,1), c_1 , c_2 = learning coefficients called the cognitive and social factors respectively, and r_1 and r_2 = random numbers generated under uniform distribution in the range (0,1).

Binary Particle Swarm Optimization (BPSO)

Developed in 1997, BPSO (Kennedy & Eberhart, 1997) is a modified version of PSO for discrete search space. In this model, the position of each particle is restricted by only two values (0 and 1). The value of the position is defined by a probability based on the velocity of the particle given by

$$S(V_{t+1}) = \frac{1}{1 + e^{(-V_{t+1})}} \quad (7)$$

$$X_{t+1} = \begin{cases} 0 & \text{if } random < S(V_{t+1}) \\ 1 & \text{if } random > S(V_{t+1}) \end{cases} \quad (8)$$

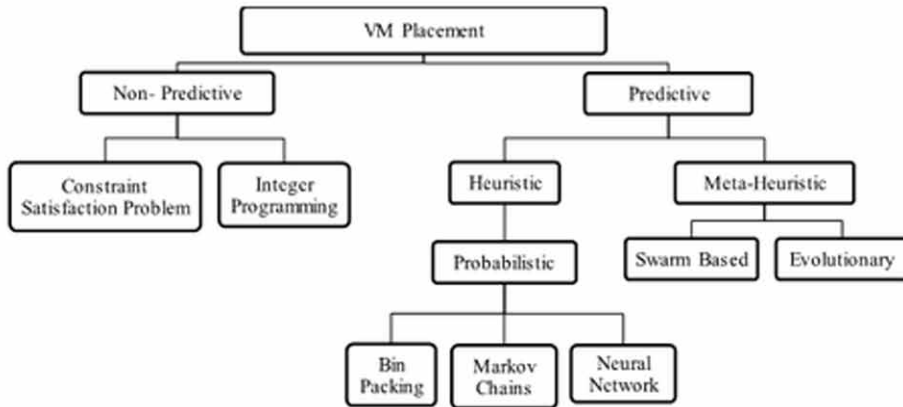
Where $S(V_{t+1})$ = velocity dependent sigmoid function, X_{t+1} = next position of the particle

The next position of the particle is computed based on the value of a sigmoid function with velocity as its parameter (equation 7). A random number is generated and compared with the value of the function. The position is updated according to equation 8. If the value of the random number generated comes out to be less than the value of the evaluated sigmoid function, then the next position of the particle is set as 0. Otherwise the position of the particle is set as 1.

Existent VM placement Techniques

Various researchers have contributed to the area of VM placement in the literature. Few have proposed forecasting techniques, some used greedy methods while many others used meta heuristics. The taxonomy of different VM placement techniques is represented in Figure 2.

Figure 2. VM placement taxonomy



The researchers (Beloglazov et al., 2012)(Huang & Tsang, 2012)(Farahnakian et al., 2016)(Hieu et al., 2017)(Zhou et al., 2016)(Rajabzadeh & Toroghi, 2017)(Wang & Tianfield, 2018a)(Mapetu et al., 2020) considered the problem of VM placement as the bin packing problem. (Beloglazov et al. (Beloglazov et al., 2012) proposed a VM placement scheme that chooses the host with a minimal rise in the level of power consumption after the placement of the virtual machine. Huang & Tsang (Huang & Tsang, 2012) have modelled the virtual machine consolidation as the minimum cost optimization problem aims to place the virtual machines on a given group of hosts while reducing the service level agreement violations. Farahnakian et al. (Farahnakian et al., 2016) proposed a scheme that checks the future load condition along with the current CPU utilization of the host while placing the VMs. Mapetu et al. (Mapetu et al., 2020) used Pearson Correlation Coefficient to find correlation between resources and the selected the host based upon the correlation coefficient. Many others modelled the problem based on constraint satisfaction (Dupont et al., 2012)(L. Zhang et al., 2013) (Tchana et al., 2016) while few have focussed on linear programming (Tseng et al., 2015)(Zeng et al., 2015)(Huang & Tsang, 2016). For improving the placement of VMs on the servers, (Hammer et al., 2017) has used a forecasting technique. The technique predicts the resource consumption using a Hidden Markov Model (HMM). The time variant version of HMM has remarkably improved the prediction capabilities of the system. (Hieu et al., 2017) has also proposed a prediction model for the consolidation of VMs. The estimated values of utilizations along with the current usage are integrated into VM placement process to improve the performance of the data centre. The workload and the number of migrations have been considerably reduced using the proposed scheme. (Rajabzadeh & Toroghi, 2017) has also used markov chain integrated with simulated annealing to optimize the energy efficiency during the VM placement process. A greedy method has been proposed for VM placement during the consolidation process in (Wang & Tianfield, 2018b). The technique selects the host with minimum available MIPS after allocating the VM and places the VM on it. In (Chen et al., 2018), the authors have used Neural Networks to predict the future demands of the resources and depending upon this, correlation aware placement algorithms has been proposed while improving the resource utilization. Authors (Rahmanian et al., 2018) have used ensemble prediction along with learning automata and correlation coefficients to improve the energy efficiency in the allocation of hosts to VMs. Many optimization algorithms have also been proposed by several researchers to chair the VMs onto different hosts. One such technique has been developed by the authors (Satpathy et al., 2018) using crow search. The authors have tried to maintain a balance between resource wastage and power consumption during the process of VM Placement. A gravitational search based optimization

algorithm has been put forward by the authors in (F. A. W. Zhang, 2019) to reduce power wastage in order to improve the efficiency of VM placement. A VM placement policy, PS-ABS, has been proposed by authors in (Xu et al., 2013) where the selection process is carried out using Artificial Bee Colony(ABC). In order to obtain a uniformity in the initialization process, the authors have introduced binary search along with ABC. The incorporation of binary search reduces the amount of overhead incurred. The proposed scheme also employs Bayes theorem along with Boltzmann selection that helps in achieving the objectives of less consumption of power and less VM migration failures making the system more reliable. Tripathi et. al. (Tripathi et al., 2018) have modelled the problem of VM placement as an optimization problem with the objective of reducing the resource wastage. The authors solved the problem with the help of Dragonfly algorithm. Few evolutionary based VM placement techniques (Tang & Pan, 2015)(Kaaouache & Bouamama, 2015)(Sharma & Saini, 2019) have also been proposed in literature. Tang and Pan(Tang & Pan, 2015) have proposed a hybrid GA-based virtual machine placement technique that considers the energy consumed by communication network along with the incorporated in the proposed scheme. Kaaouache & Bouamama(Kaaouache & Bouamama, 2015) have focussed on the infeasible solution problem arising with one-dimensional bin packing problem. To solve this, the authors have put forward a solution combining GA with Best Fit Decreasing. Sharma & Saini(Sharma & Saini, 2019) have used NSGA to deal with the VM placement aimed at improving the performance of the network while minimizing the energy consumption and the number of migrations. The main idea is to sort the individuals on the basis of the levels of their dominations. Several other authors have used Ant Colony (Xiao et al., 2019) (Liu et al., 2017) (Liu et al., 2018) (Alharbi et al., 2019), Bio Geography based Optimization (Zheng et al., 2016)(Teyeb et al., 2017)(Li et al., 2017)(Pahlevan et al., 2016) and Particle Swarm Optimization (Tripathi et al., 2018) (Yan et al., 2018)(Ibrahim et al., 2020) for accomplishing the task of VM Placement. Tripathi et al. (Tripathi et al., 2018) have used binary PSO to optimize VM placement by taking energy usage and utilization of resources into consideration. Yan et al. (Yan et al., 2018) have concentrated on memory and CPU utilization as parameters of the objective function in discrete PSO. OpenStack has been used to evaluate the capabilities of the proposed algorithm and has been compared with the native VM scheduler of OpenStack. The initialization of the particle is an important step in the execution of PSO. Many researchers (Bangyal, Hameed, et al., 2021)(Bangyal, Nisar, et al., 2021)(Bangyal, Rauf, et al., 2019) have focussed on this step of particle initialization to optimize the convergence rate of the optimization algorithm. (Xiao et al., 2019) and (Alharbi et al., 2019) defined the subject as a combinatorial optimization problem and (Liu et al., 2017) also worked on VM placement as a multi objective problem, (Xiao et al., 2019) (Liu et al., 2017) aimed at reducing the number of migrations and consumed power. The authors (Ibrahim et al., 2020) have tried to minimizing the number of active servers, while trying to reduce resource wastage and power consumption simultaneously.

Most of the researchers have focussed on optimizing the power consumption model and the resource wastage. However, very few have focussed on resource utilization as a means to improve energy efficiency. The proposed scheme essentially tries to make the system energy efficient by monitoring the resource utilization of the system with the help of the swarm based metaheuristic technique.

Problem Statement

The aim of the VM placement process is to focus on minimizing the number of physical machines (PMs) that are active at any given instant of time by making each node work under the maximum capacity while maintaining the constraints of threshold. Various system resources contained by a machine include CPU, memory, bandwidth, and disk. The resource utilization should be neither below the threshold nor should it exceed it. There exists a linear relationship between resource utilization and energy consumption (Fan et al., 2007). In order to save energy, the total energy consumption of the data centre should be minimized. According to the study of Srikantaiah (Srikantaiah et al., 2008), the energy efficiency of a physical node can be defined in terms of the Euclidean distances

of the resource utilization from their optimal utilizations. Smaller the Euclidean distance, better is the energy efficiency.

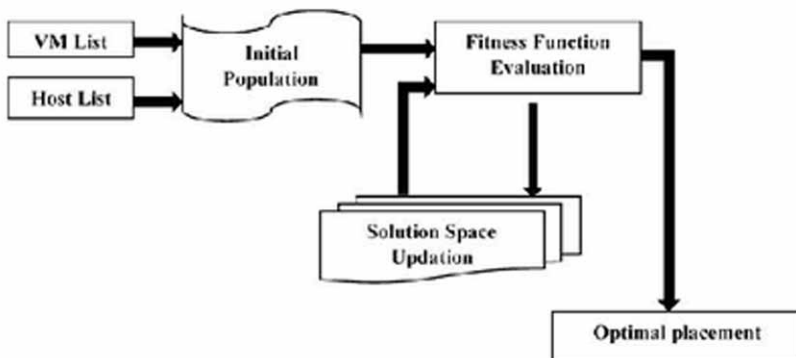
PSO-based on Resource Aware VM placement (RAPSO_VMP)

RAPSO_VMP focusses on finding a suitable host for VM Placement by finding a near optimal solution using PSO. It receives as input the list of n VMs that need to be migrated and the list of m available hosts. The list of available hosts excludes the overloaded hosts, so as to avoid the selected VMs getting placed on the overloaded hosts.

Block Diagram of RAPSO_VMP

Once the overloaded servers are detected, some VMs need to be migrated and placed to other servers so as to make the system keep working under proper conditions. Placement of the VMs to appropriate servers is an important task. Thus, the proposed work suggests a technique based on BPSO with the intention to find a near optimal solution for VM Placement. The flow of execution of the proposed scheme is shown in Figure 3.

Figure 3. Block diagram of RAPSO_VMP



Description of each component given in the block diagram:

VM List: This component comprises of the list of n VMs that are to be migrated.

Host List: A list of all m available hosts is contained in this component.

Initial Mapping: The input to this block is the list of n VMs and m available hosts. Mapping has been done by this block to generate the initial population. In RAPSO_VMP, the population has been initialized using greedy method.

RAPSO_VMP Initial Mapping Example

The initial mapping produced by RAPSO_VMP comprises of particles, where each particle has few dimensions. The number of dimensions is equal to the number of VMs to be migrated. Each dimension holds a value that refers to the index of the VMs. A PM can host several VMs but the reverse does not hold, i.e., a VM can only be placed on one PM. The mappings between VMs and hosts have been further evaluated until reaching the best fitness function. An example of the generation of initial mapping has been explained below. The assumptions under consideration are:

$VirMach_{list}$ [list of n VMs to be migrated] = $\{VM_1, VM_2, VM_3, VM_4\}$

$Host_{list}$ [list of total number of hosts] = $\{H_1, H_2, H_3, H_4, H_5, H_6, H_7, H_8, H_9, H_{10}\}$

$OLHost_{list}$ [list of overloaded hosts] = $\{H_3, H_6, H_7, H_{10}\}$

$AvailHost_{list}$ [list of available hosts, m] = $\{H_1, H_2, H_4, H_5, H_8, H_9\}$

VM_1, VM_2, VM_3 and VM_4 need to be placed (Table 1) on the available hosts. H_3, H_6, H_7 and H_{10} are overloaded. Thus, the list of available hosts comprises of H_1, H_2, H_4, H_5, H_8 and H_9 (Table 2).

Table 1. List of VMs to be migrated

VM_1	VM_2	VM_3	VM_4
--------	--------	--------	--------

Table 2. List of available hosts

H_1	H_2	H_4	H_5	H_8	H_9
-------	-------	-------	-------	-------	-------

Table 3 represents 6 random sequences of the VMs that need to be placed. The placement of the random sequences to available hosts on first fit basis have been shown in Table 4, Table 5, Table 6, Table 7, Table 8 and Table 9 respectively.

Table 3. Sequences of VMs

1234	2341	3412	4123	2134	1342
------	------	------	------	------	------

Table 4. Placement sequence 1

For Sequence 1			
VM_1	VM_2	VM_3	VM_4
H_1	H_2	H_4	H_5

Table 5. Placement sequence 2

For Sequence 2			
VM_2	VM_3	VM_4	VM_1
H_2	H_5	H_8	H_9

Table 6. Placement sequence 3

For Sequence 3			
VM_3	VM_4	VM_1	VM_2
H_2	H_2	H_1	H_5

Table 7. Placement sequence 4

For Sequence 4			
VM_4	VM_1	VM_2	VM_3
H_1	H_8	H_4	H_5

Table 8. Placement sequence 5

For Sequence 5			
VM_2	VM_1	VM_3	VM_4
H_5	H_2	H_4	H_8

Table 9. Placement sequence 6

For Sequence 6			
VM_1	VM_3	VM_4	VM_2
H_5	H_2	H_1	H_9

Table 10. Initial mapping

	VM_1	VM_2	VM_3	VM_4
H_1	1	0	0	0
H_2	0	0	1	1
H_3	0	0	0	0
H_4	0	0	0	0
H_5	0	1	0	0
H_6	0	0	0	0
H_7	0	0	0	0
H_8	0	0	0	0
H_9	0	0	0	0
H_{10}	0	0	0	0

Table 11. Particle representation

VM_1	VM_2	VM_3	VM_4
H_1	H_2	H_4	H_5
H_9	H_2	H_5	H_8
H_1	H_5	H_2	H_2
H_8	H_4	H_5	H_1
H_2	H_5	H_4	H_8
H_5	H_9	H_2	H_1

Figure 4. RAPSO_VMP initial mapping

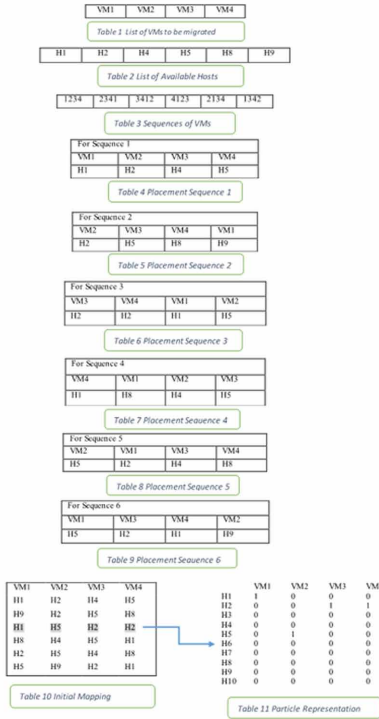


Table 10 represents the arrangement of the above-obtained placements in ascending order of VM index, thus, providing the initial mapping of particles. The matrix [0,1] representation of a particle is shown in Table 11. Thus, Figure 4 pictorially explains the initial mapping between the VMs to be migrated and the available hosts.

Fitness Function Evaluation: This component calculates the fitness function (FF) of all the mappings done in the previous step.

Fitness function(FF): To reach to a quality solution, the optimization techniques depend on a fitness function. Depending upon the value of the fitness function, the position of the particles in the population is updated until the best value is achieved. The fitness function can be either a minimizing or maximizing function. In the proposed technique, the goal is to minimize the total power consumption by trying to make the resource utilizations work at their best. The idea here is to reduce the Euclidean distance between the current and optimal utilizations. Equation 9 is the mathematical expression to evaluate the energy efficiency λ of a host .

$$\lambda_j = \sqrt[2]{\sum_{i=1}^d (\sigma_{j_i} - \sigma_{j_{best_i}})^2} \quad (9)$$

Where, $d = 3$ corresponding to the type of resource for e.g. CPU, RAM, Disk

λ_j = Energy efficiency of a single host j

A = current utilization of resources under consideration i.e. CPU, RAM and the disk.

\tilde{A}_{j_i} = current resource utilization of the resource i for host j , and
 $\tilde{A}_{j_{best_i}}$ = optimal utilization of the resource i for host j .

In order to bring the utilizations of different resources to a common scale, their normalised values need to be evaluated. Energy efficiency of a host j after the normalization of resources has been achieved with the help of equation 10.

$$\lambda_{jnormalised} = \sqrt[2]{\sum_{i=1}^d \left\{ \left(\frac{\omega_{j,current} - \omega_{j,min}}{\omega_{j,max} - \omega_{j,min}} \right) - \left(\frac{\omega_{j,best} - \omega_{j,min}}{\omega_{j,max} - \omega_{j,min}} \right) \right\}^2} \quad (10)$$

Where, $\lambda_{jnormalised}$ = normalised energy efficiency of host j ,

$\omega_{j,current}$ = current utilization of resource i for host j ,
 $\omega_{j,min}$ = lower utilization threshold of resource i for host j ,
 $\omega_{j,min,j,max}$ = upper utilization threshold of resource i for host j , and
 $\omega_{j,best}$ = optimal utilization of the resource i for host j

Equation 11 signifies the current utilization of a host j . which is equal to the summation of utilizations of k VMs residing on that host. Therefore,

$$\omega_{j,current} = \sum_{x=1}^k \omega_{j_i x} \quad (11)$$

Where, $\omega_{j_i x}$ = utilization of VM x corresponding to resource i for host j ,

k = total number of VMs on host j

Equation 12 presents the mathematical expression for calculating the energy efficiency factor.

$$\lambda_{jnormalised} = \sqrt[2]{\sum_{i=1}^d \left\{ \left(\frac{\left(\sum_{x=1}^k \dot{E}_{j_i x} \right) - \dot{E}_{j_i min}}{\dot{E}_{j_i max} - \dot{E}_{j_i min}} \right) - \left(\frac{\dot{E}_{j,best} - \dot{E}_{j,min}}{\dot{E}_{j,max} - \dot{E}_{j,min}} \right) \right\}^2} \quad (12)$$

The summation of Euclidean distances of all the active nodes simultaneously gives the estimate of the total energy consumption of the entire system. The objective of the current research is to make the active hosts work at the maximum capacity by minimizing the value of the total energy efficiency factor obtained in equation 12. The fitness function (FF) shown in equation 13 can be obtained by substituting the value of energy efficiency factor, $\lambda_{jnormalised}$, from equation 12.

$$FF = Min. \left(\sum_{j=1}^m \lambda_{jnormalised} \right) \quad (13)$$

Thus, minimizing the summation of energy efficiency factor of all m available hosts can achieve the optimum value of the FF.

Solution Space Updation: The generated population by the “initial mapping” component acts as an input to this block. The initial mapping corresponds to the initial position of the particles. The iterative process of binary PSO is carried out in this component to update the solution space. Thus, the updation of particle position and velocity takes place in this step. In order to reach an optimum solution, the use of transfer function plays a vital role in a binary optimization algorithm. With the help of the transfer function, the continuous search space is mapped to the discrete search space. The existing S shaped, V shaped and various other linear transfer function sometimes get trapped into local optima which leads to poor exploration resulting in premature convergence. However, in order to achieve a good result, the transfer function should essentially provide a balance between exploration and exploitation. Considering the limitations of the existing transfer function, a mirrored S transfer function has been used, in the proposed scheme. The time varying nature of the transfer function helps in avoiding local optima in the initial steps by performing strong exploration which gets switched to exploitation in the final steps for finding the best results. Equation 14 and 16 denote the two different components of the mirrored transfer function, S_1 and S_2 , that has been utilized to update the position of the particle. The mirrored transfer function helps in evaluating the two intermediate positions, y_{t+1} and z_{t+1} of the particle, based on equation 15 and 17. Equation 18 helps in finding the best among these two calculated positions.

$$S_1((V_{t+1}), \Upsilon) = \frac{1}{1 + e^{\Upsilon(-V_{t+1})}} \quad (14)$$

$$y_{t+1} = \begin{cases} 0 & \text{if } random1 < S_1((V_{t+1}), \Upsilon) \\ 1 & \text{if } random1 > S_1((V_{t+1}), \Upsilon) \end{cases} \quad (15)$$

Where, $S_1((V_{t+1}), \Upsilon)$ = first time varying sigmoid function and y_{t+1} is the first intermediate position of the particle. Υ = time varying factor. $random1$ = random number $\in [0, 1]$

$$S_2((V_{t+1}), \Upsilon) = \frac{1}{1 + e^{\Upsilon(V_{t+1})}} \quad (16)$$

$$z_{t+1} = \begin{cases} 0 & \text{if } random2 < S_2((V_{t+1}), \Upsilon) \\ 1 & \text{if } random2 > S_2((V_{t+1}), \Upsilon) \end{cases} \quad (17)$$

Where, $S_2((V_{t+1}), \Upsilon)$ = second time varying sigmoid function and z_{t+1} is the second intermediate position of the particle. $random2 = \text{random number} \in [0, 1]$

$$X_{t+1} = \begin{cases} y_{t+1} & \text{if } FF(y_{t+1}) \text{ is better than } FF(z_{t+1}) \\ z_{t+1} & \text{if } FF(z_{t+1}) \text{ is better than } FF(y_{t+1}) \end{cases} \quad (18)$$

Where, X_{t+1} = next position of the particle, $FF(y_{t+1})$ = fitness of the particle at position y_{t+1} , and $FF(z_{t+1})$ = fitness of the particle at position z_{t+1}

If the value of fitness function evaluated at y_{t+1} is better than z_{t+1} , then y_{t+1} is selected as the next position of the particle otherwise the position of the particle is update with z_{t+1} .

Optimal Placement: This role of this component is to represent the best VM-Host mapping by evaluating the minimum value for the fitness function.

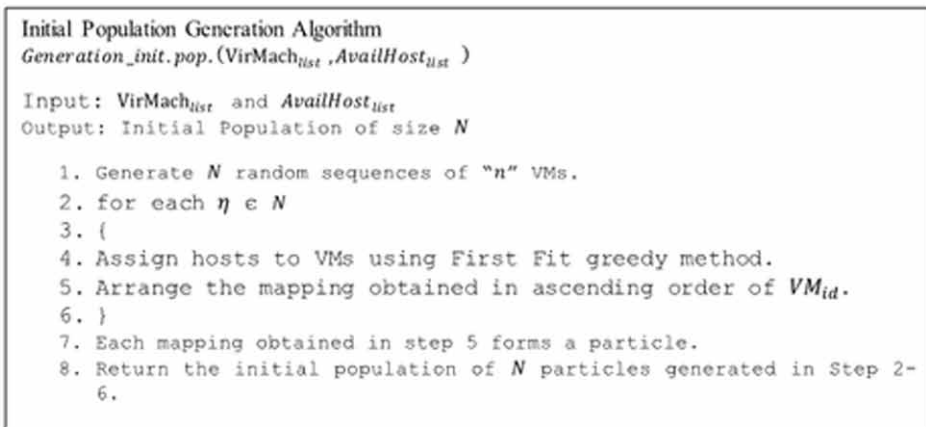
Modifications in BPSO

This section describes various modifications that have been incorporated in the traditional binary PSO for mapping the problem of VM placement to BPSO.

Population Initialization

Figure 5 describes the Initial Population Generation Algorithm explaining the steps involved in the generation of swarm of particles constituting the initial population.

Figure 5. Initial population generation algorithm



To get started, RAPSO_VMP is initialized by defining few parameters namely, number of particles, number of iterations, number of dimensions, the minimum and maximum limits of time varying variable, the values for cognitive parameter and the social parameter. The population size N refers to the number of particles where each particle represents a mapping of VM and the available hosts. The dimension of each particle equals the number of VMs that need to be migrated and placed. In order to generate the N particles, create N random sequences of VMs. Then, for each sequence

find a suitable host on first fit basis (greedy method). It will lead to N VM-host distributions. Arrange them in increasing order of VM_{id} . Each such mapping, arranged in increasing VM_{id} , constitutes to the list of the particles and the generated sequences form the initial position.

Particle Representation

Each particle P is represented in the form of a $n \times m$ matrix with n VMs are mapped to m PMs. Each entry α_{ij} of the matrix is either 0 or 1. If i^{th} VM is assigned to j^{th} PM, then the corresponding entry is 1 otherwise 0.

$$\begin{bmatrix} 0 & 1 & \dots & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 & 1 & 0 \\ \vdots & 0 & \ddots & 0 & \dots & 0 \\ 0 & 0 & 1 & \ddots & \dots & 1 \\ \vdots & 0 & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & 1 & 0 & 0 \end{bmatrix}$$

Checking Constraints

1. *Capacity Constraint*: Once the position has been updated, one must ensure that the new position satisfies the constraints as mentioned in equation 2. According to equation 2, the allocated host should have the capacity to accommodate the new Virtual machine. Thus, the requirements of the VM should be checked against the remaining capacity of the server before updating the position to 1.
2. *Placement Constraint*: Equation 4 holds that a VM can be allocated only to a single host. If the host has the capacity to house the VM and if the same VM has not been yet hosted by any other server, only then the position of the VM in the particle can be updated to 1. For this purpose, an array of size j (equal to the number of rows/VM) is maintained that keeps track of the entries of the particle. When a VM is allocated to a server then the value of the corresponding array index is set to 1. Thus the position α_{ij} of a particle is updated to 1 if and only if i^{th} server has the capacity to hold j^{th} VM and i^{th} column does not contain any entry equal to 1.

RAPSO_VMP Algorithm

Figure 6 shows the RAPSO_VMP Algorithm explaining the flow of execution of RAPSO_VM placement.

Figure 6. RAPSO_VMP algorithm

```

RAPSO_VMP Algorithm
Input: Set of VMs with their computing and memory demands, set of
physical servers with their capacity threshold and set of parameters
Output: Optimal VM placement mapping

1. Initialize the position of the swarm of particles using Initial
   Population Generation Algorithm.
2. While stopping condition is not met do
3. Compute the fitness value for each particle using equation
   (12) and update the gBestValue, gBestPosition, pBestValue and
   pBestPosition.
4. For all particles do
5. Update the velocity using equation (5)
6. Update the position using equations (14) through (18)
7. End for
8. End while
9. Return gBestPosition

```

The set of VMs that are to be migrated are fed as input which ultimately provides the best VM-host mapping as the solution. The algorithm begins with the particle generation and initialization. This is followed by the start of the iterative process where velocity and position is updated until the stopping criteria is met. Each iteration also includes the best position and best values calculations.

RAPSO_VMP Pseudocode

1. Initialize parameters: N_p , N_t , c_1 , c_2 , w , Υ_{max} , Υ_{min}
2. Input $VirMach_{list}$, $AvailHost_{list} = Host_{list} - OLHost_{list}$
3. Generate initial population using Initial Population Generation Algorithm.
4. $gBestValue \leftarrow Max.Value$
5. for each particle $p \in P$ do
6. {
7. $pBestValue \leftarrow Max.Value$
8. }
9. $t \leftarrow 0$
10. while $t < N_t$
11. {
12. for each particle $p \in P$ do
13. {
14. Calculate FF for p based on eq. (13)
15. if $(FF > pBestValue)$ then
16. {
17. $pBestValue \leftarrow FF$
18. $pBestPosition \leftarrow X_t$
19. }
20. if $(FF > gBestValue)$ then
21. {
22. $gBestValue \leftarrow FF$
23. $gBestPosition \leftarrow X_t$
24. }
25. }
26. for each particle $p \in P$ do

```

27. {
28.   $r_1 \leftarrow \text{random}(0,1)$ 
29.   $r_2 \leftarrow \text{random}(0,1)$ 
30.   $r_3 \leftarrow \text{random}(0,1)$ 
31.   $r_4 \leftarrow \text{random}(0,1)$ 
32.   $V_{t+1} \leftarrow [(w \times v_t) + r_1 c_1 (pBestPosition - X_t) + r_2 c_2 (gBestPosition - X_t)]$ 
33.   $\Upsilon \leftarrow \left( (\Upsilon_{max} - \Upsilon_{min}) \left( \frac{t}{N_t} \right) + \Upsilon_{min} \right)$ 
34.   $S_1((V_{t+1}), \Upsilon) = \frac{1}{1 + e^{\Upsilon(-(V_{t+1}))}}$ 
35.   $y_{t+1} \leftarrow \begin{cases} 0 & \text{if } r_3 < S_1((V_{t+1}), \Upsilon) \\ 1 & \text{if } r_3 > S_1((V_{t+1}), \Upsilon) \end{cases}$ 
36.   $S_2((V_{t+1}), \Upsilon) = \frac{1}{1 + e^{\Upsilon(V_{t+1})}}$ 
37.   $z_{t+1} \leftarrow \begin{cases} 0 & \text{if } r_4 < S_2((V_{t+1}), \Upsilon) \\ 1 & \text{if } r_4 > S_2((V_{t+1}), \Upsilon) \end{cases}$ 
38.   $X_{t+1} \leftarrow \begin{cases} y_{t+1} & \text{if } FF(y_{t+1}) > FF(z_{t+1}) \\ z_{t+1} & \text{if } FF(z_{t+1}) > FF(y_{t+1}) \end{cases}$ 
39. }
40.  $t \leftarrow t + 1$ 
41. }
42. return  $gBestPosition$ 

```

The very first step is to initialize number of particles, number of iterations, number of dimensions, the minimum and maximum limits of time varying variable, the values for cognitive parameter and the social parameter. Thus, the algorithm begins with the initialization of these parameters (line 1). Next step is responsible for excluding the overloaded hosts from the process of placement, thereby producing a list of the available hosts for receiving the migrated VMs (line 2). The flow then continues by generating the initial population (line 3). The goal is to find out the best particle among the swarm. Since the fitness function is the minimization one, the best position has been initialized with a maximum value (line 4). The best position of each particle has also been initialized with a maximum value (line 7). After the initializations are done, the technique iterates through a number of steps until the goal is achieved (reaching the goal), beginning with the initialization of iteration number to 0 (line 9). The iteration process continues until the iteration counter does not reach the max number of iterations (line 10). Each particle is then evaluated for fitness function using equation 13 (lines 12-14). If the particles' new fitness value is better than its best value, then updates are required in the particles' best position and particles' best value (lines 15-19). The best position in the swarm is found out by evaluating the particle for which fitness function has the best value (lines 20-24). This is followed by updating each particle positions (lines 26-49). In the updating process, few random numbers are used (lines 28-31). A new speed is calculated using PSO velocity updating equation (line 32). The time varying variable is updated depending upon the current iteration number (line 33). Next step is to update the position using the velocity evaluated in the previous step, using a time varying transfer function (lines 34-48). The value for time varying sigmoid function is calculated (line 34).

This value is compared with a random number generated to compute first intermediate position (line 35). Second sigmoid function is also calculated (line 36) to find out the value of second intermediate position (line 37). The position with better fitness is considered as the new position (line 38). Before jumping to the next iteration, the iteration counter is incremented by 1 (line 40). Once the set of iterations is over, the position of the best particle is returned (line 42).

EXPERIMENTATION AND RESULTS

This section details about the simulation test bench and the analysis of the results obtained on executing the proposed scheme in the simulation environment.

Simulation Test Bench

In order to evaluate the efficiency of the proposed algorithm, CloudSim 3.0.3 has been used. CloudSim is an open source toolkit developed by CLOUDS laboratory at the University of Melbourne. CloudSim provides an appropriate environment for virtualization, resource management, VM migration and energy efficiency along with evaluation of SLA violations (SLAV). The simulation experiments have been carried out on a PC with Inter Core(TM) i5-8250U CPU @ 1.60 GHz and 8 GB of RAM based on windows environment with Eclipse IDE. The servers and VMs used for simulation are heterogeneous in nature. Two types of servers have been used: HP ProLiant ML110 G4 server and HP ProLiant ML110 G5 server.

Experimental Setup

Table 12 represents the three different test cases, with different number of VMs and hosts, that have been used for performing the experiments. Random workload has been used by the test cases.

Table 12. Evaluated test cases

Test Cases	W1	W2	W3
Number of VMs	50	75	100
Number of PMs	50	75	100

Each experiment has been executed 10 times and the average of the results have been compared with the well-known placement algorithm PABFD(Beloglazov & Buyya, 2012). Four types of VMs have been incorporated, each being single core. The parameters have been initialized with the values as shown in Table 13.

Table 13. Parameter values for RAPSO_VMP

N_p	N_t	c_1	c_2	w	σ_{max}	σ_{min}
30	50	1.49	1.49	0.7298	1.0	0.1

The characteristics of servers and VMs have been presented in Table 14 and Table 15. Performance metrics show that RAPSO_VMP outperforms the existing scheme.

Table 14. Server characteristics

Server type	Frequency of Core (MIPs)	Number of Cores	Memory (GB)	Bandwidth (Gb/s)	Storage (GB)
HPProLiantML110G4	1860	2	4	1	1000
HPProLiantML110G4	2660	2	4	1	1000

Table 15. VM characteristics

VM type	Frequency of Core (MIPs)	Number of Cores	Memory (MB)	Bandwidth (Mb/s)	Storage (MB)
High CPU VM	2500	1	870	100	2.5
Extra Large VM	2000	1	1740	100	2.5
Medium VM	1000	1	1740	100	2.5
Micro VM	500	1	613	100	2.5

Analysis of Results

To analyse the performance of RAPSO_VMP, the proposed algorithm has been compared with different benchmark algorithms (Beloglazov & Buyya, 2012). The major metrics for evaluating the performance of the consolidation schemes are: Energy Consumption, SLA time per active host, SLA performance degradation due to migration, SLAV and Energy performance metric (ESV and ESM). The authors (Beloglazov & Buyya, 2012) have proved the dominance of LRMMT over other schemes. In our experiments also, with LRMMT as the DVMC scheme, RAPSO_VMP shows that maximum improvement against PABFD (Table 16).

Table 16. Comparison with different existing schemes

DVMC Scheme		Energy Consumption (KWh)	Number of VM Migrations	Number of Host Shutdowns	SLAV (%)	Energy SLAV	Energy SLAV Migrations
LRMMC	PABFD	50.12	3771.67	1063	0.000038	0.00002	0.085
	RAPSO_VMP	48.08	3456.7	856.48	0.000032	0.000017	0.069
LRMMT	PABFD	51.58	5002.67	1256	0.000035	0.000019	0.113
	RAPSO_VMP	48.74	4545.93	970.34	0.000030	0.000016	0.09
LRMU	PABFD	51.92	4758.33	1267.67	0.000040	0.000022	0.121
	RAPSO_VMP	49.39	4503.7	1010.37	0.000037	0.00002	0.103
LRRS	PABFD	49.96	3811	1057.33	0.000041	0.000021	0.088
	RAPSO_VMP	49.35	3599.82	972.93	0.000037	0.000018	0.071
MADMMC	PABFD	66.01	7315.33	2164.33	0.000067	0.000044	0.345

Table 16 continued on next page

Table 16 continued

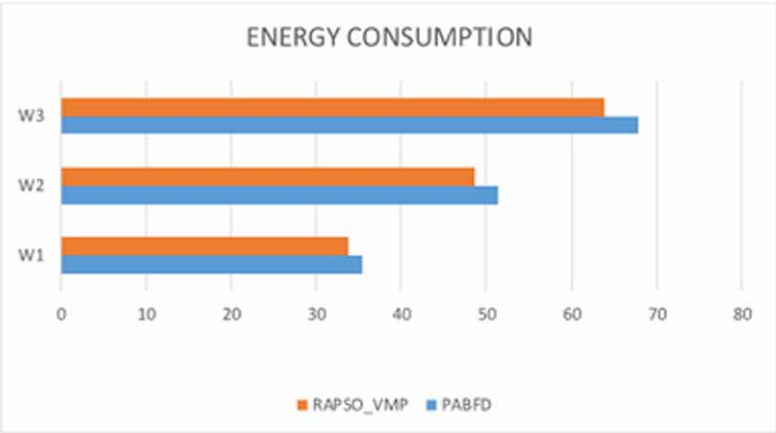
DVMC Scheme		Energy Consumption (KWh)	Number of VM Migrations	Number of Host Shutdowns	SLAV (%)	Energy SLAV	Energy SLAV Migrations
	RAPSO_VMP	64.29	7316.81	1987.98	0.000067	0.000043	0.342
MADMMT	PABFD	67.02	7860.67	2220.67	0.000045	0.000031	0.257
	RAPSO_VMP	64.94	7881.51	1977.28	0.000042	0.000027	0.225
MADMU	PABFD	69.55	8367.67	2370.67	0.000064	0.000044	0.394
	RAPSO_VMP	67.54	8496.97	2187.82	0.000066	0.000044	0.401
MADRS	PABFD	65.77	7311.33	2172	0.000067	0.000045	0.354
	RAPSO_VMP	63.99	7220.07	1964.68	0.000063	0.000041	0.32
IQRMMC	PABFD	69.18	7666.67	2242	0.000055	0.000038	0.313
	RAPSO_VMP	67.4	7690.2	2015.05	0.000055	0.000037	0.306
IQRMMT	PABFD	70.07	8248	2279.33	0.000039	0.000027	0.238
	RAPSO_VMP	68.19	8085.06	2053.31	0.000033	0.000022	0.196
IQRMU	PABFD	72.37	8838	2409.33	0.000057	0.000042	0.398
	RAPSO_VMP	70.31	8815.26	2209.69	0.000056	0.00004	0.379
IQRRS	PABFD	68.89	7625.67	2254.67	0.000058	0.000041	0.335
	RAPSO_VMP	67.66	7604.33	2061.43	0.000059	0.00004	0.323
THRMMC	PABFD	60.32	6513	2013.67	0.000098	0.000059	0.41
	RAPSO_VMP	58.81	6521.29	1850.83	0.000099	0.000059	0.415
THRMMT	PABFD	61.89	7136	2033	0.000061	0.000037	0.275
	RAPSO_VMP	60.03	7165.07	1793.88	0.000053	0.000031	0.234
THRMU	PABFD	65.43	7821.67	2269	0.000087	0.000055	0.452
	RAPSO_VMP	63.32	7889.78	2023.89	0.000081	0.00005	0.409
THRRS	PABFD	60.88	6633.33	2022	0.000094	0.000057	0.409
	RAPSO_VMP	59.66	6617.58	1845.28	0.000090	0.000054	0.387

Energy Consumption

Energy consumption is a major metric and can be defined as the sum of the energy consumed by all the hosts in the cloud data centre. It is obtained using the integral of the consumption of power over a time period. Figure 7 shows the minimized average energy consumption for all three test cases. The X-axis represent the amount of energy consumed in KWh against each workload, represented on the Y-axis. There is an average improvement of 5.51% when RAPSO_VMP is compared with PABFD.

$$Power = \int_{t_0}^{t_1} P(x(t)) dt$$

Figure 7. Energy consumption vs VM placement scheme

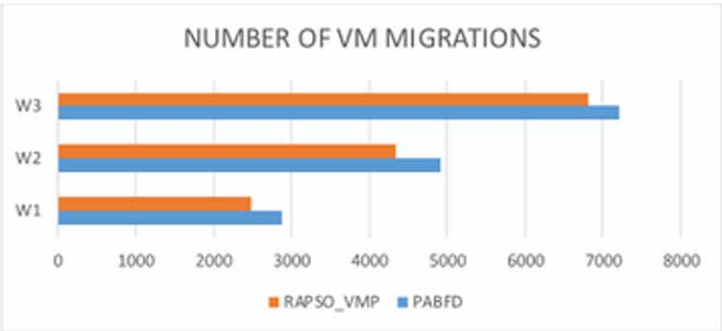


Number of VM Migrations

Number of VM migrations represent the total number of VMs that have been migrated during the process of consolidation. Minimal number of migrations is desirable for reduced performance degradation and lesser consumption of bandwidth. Large number of migrations can also lead to the violation of SLA and system degradation. RAPSO_VMP tends to increase the utilization of the hosts, thereby reducing the number of underloaded hosts. The proposed scheme reduces the number of migrations as compared to PABFD. The number of incurred VM migrations are depicted along the X-axis whereas the corresponding workload is shown along the Y-axis. There is an improvement of 9.13%.

The results for the number of VM migrations incurred have been shown in Figure 8.

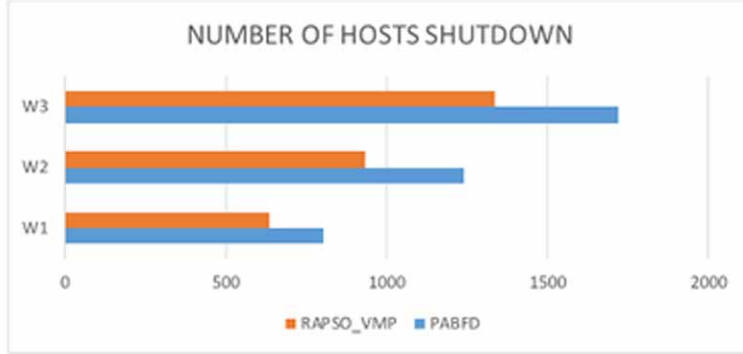
Figure 8. Number of VM migrations vs VM placement scheme



Number of Host Shutdown

Number of hosts shutdown is another important that should be minimized performing the consolidation. Frequent host on-off leads to poor consolidation. However, fewer and longer shutdowns make the system more energy efficient. The reduced number of hosts shutdown for the proposed scheme has been depicted in Figure 9.

Figure 9. Number of host shutdown vs VM placement scheme



SLA Violation (SLAV)

SLA violation is the product of SLA time per active host (SLATAH) and performance degradation (PDM) due to migrations. This metric should be kept as minimum as possible to meet the QoS requirements.

$$SLAV = SLATAH \times PDM$$

$$SLATAH = \frac{1}{N} \sum_{i=1}^N \frac{T_{f_i}}{T_{a_i}}$$

SLATAH is the percentage of time for which active hosts have experienced 100% utilization. Where N is the total number of servers, T_{f_i} is the time period for which the server experiences full utilization and the total activation time of server is denoted by T_{a_i} .

$$PDM = \frac{1}{M} \sum_{j=1}^M \frac{C_{d_j}}{C_{r_j}}$$

C_{d_j} is the estimated performance degradation due to migration, C_{r_j} is the total requested capacity during the migration operation, M is the number of VMs. To ensure that the desired QoS is maintained in the cloud environment, there should be no violation of SLA. According to the experimental results, RAPSO_VMP does not violate SLA. However, in few cases it has been observed that the proposed

scheme avoids the SLA violations. It can be seen from figure 10 that the performance degradation remains the same for both the cases.

Figure 10. Performance degradation due to migration vs VM placement scheme

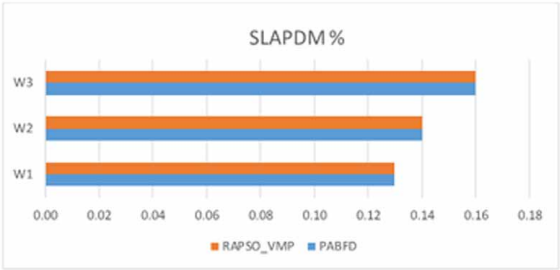


Figure 11 and figure 12 represent the results for SLAPDM, SLATAH and SLAV.

Figure 11. SLA time per active host vs VM placement scheme

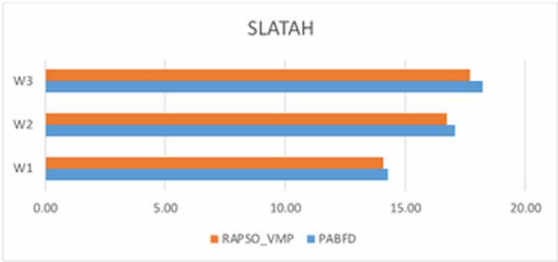
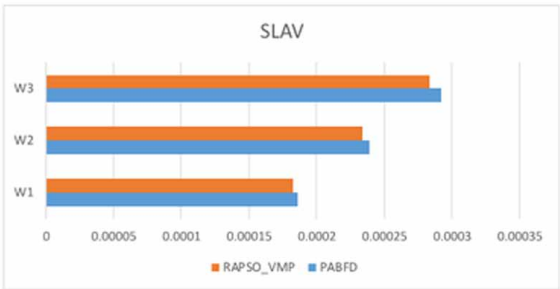


Figure 12. SLA violation vs VM placement scheme

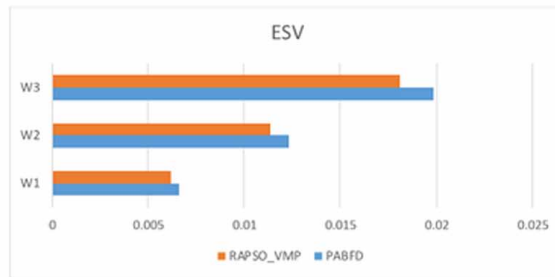


Energy-SLA Violation (ESV)

ESV denotes the product of energy consumption and SLA violation. The data centre aims at minimizing this metric. The cloud service providers have observed a trade-off between energy consumption and SLAV. Thus, to evaluate the efficiency of VM Placement schemes, a combined metric, ESV, has been formulated.

The ESV parameter for RAPSO_VMP in comparison to PABFD has been depicted in Figure 13.

Figure 13. ESV vs VM placement scheme



DISCUSSION

The energy consumed by a cloud data center has a direct relationship with resource utilization (Fan et al., 2007)(Kusic et al., 2008). Low server utilization and idle power consumption are the major reasons for poor energy efficiency. The reason behind low server utilization can be either overutilization or underutilization of resources. Better resource utilization can lead to improved energy efficiency. In RAPSO_VMP, an effort has been made to make the hosts work at their best utilization levels during consolidation. This helps in reducing the energy consumption of the whole system. The number of host shutdowns taking place in the cloud data center is an important factor for measuring the effectiveness of a consolidation scheme. Whenever a host becomes underloaded, the VMs placed on it are migrated to other machines, to turn off the underloaded machines. However, there is a chance of these machines getting turned on again during the process of live migration to accommodate the migrated machines. This frequent turning on/off of the low-loaded machines increases the energy consumption and also leads to the degradation of the system performance. With the proposed scheme, RAPSO_VMP, the VMs from the overloaded hosts are migrated to the active hosts, excluding the underloaded ones. In case no suitable host is found, then the placement is done in one of the underloaded hosts or the new ones are created. Thereby, reducing the chances of an increase in the number of underloaded hosts and a decrease in the number of host shutdowns, as shown in Table 16. Also, more migrations lead to degradation of performance thereby violating the service level agreements. The simulation results show that there is a dip in the number of VM migrations. The reason behind the decreased value of migrations is majorly is the reduced number of host shutdowns. The observations from Table 16 depict that RAPSO_VMP exhibits best results with MMT as the VM selection policy. According to the results, ESM achieved for $LRMMT < IQRMMT < THRM MT < MADMMT$. RAPSO_VMP works best with LRMMT having a maximum improvement of 18.37% in respect to the combined metric of Energy, SLAV and the number of VM migrations.

CONCLUSION AND FUTURE WORK

The ease in the availability of services offered by cloud computing has commanded to the doubling of cloud data centres across the world. However, such expanded use has led to high resource utilization and energy consumption. Consequently, there arises a need to properly manage both, the resources and the energy efficiency. Dynamic consolidation has proved to be of great significance in this respect. Consolidation comprises of finding overloaded/underloaded hosts and migrating one or few VMs from then to another suitable hosts. Thus, finding an appropriate host through VM placement is a vital task. In the current work, RAPSO_VMP has been proposed with the purpose of improving energy efficiency while making the system more resource aware. The objective has been achieved by incorporating PSO in VM Placement problem. The fitness function has been defined in terms of resource utilizations to improve the energy efficiency. Modifications in transfer function and initial population generator function have also been applied to basic BPSO to map well to the defined VM Placement problem. The simulation results show that RAPSO_VMP outperforms LRMMT in terms of energy consumption, number of migrations of VMs and number of hosts shut down. As a future research direction, the implementation of RAPSO_VMP can be implemented in real cloud environment. PlanetLab workload has been used in the current research. However, other types of workloads along with different parameters like GPU can also be used in future for testing the efficiency of the proposed scheme.

FUNDING AGENCY

The publisher has waived the Open Access Processing fee for this article.

REFERENCES

- Alharbi, F., Tian, Y. C., Tang, M., Zhang, W. Z., Peng, C., & Fei, M. (2019). An Ant Colony System for energy-efficient dynamic Virtual Machine Placement in data centers. *Expert Systems with Applications*, 120, 228–238. doi:10.1016/j.eswa.2018.11.029
- Bangyal, W. H., Ahmad, J., & Rauf, H. T. (2019). Optimization of Neural Network Using Improved Bat Algorithm for Data Classification. *Journal of Medical Imaging and Health Informatics*, 9(4), 670–681. doi:10.1166/jmihi.2019.2654
- Bangyal, W. H., Hameed, A., Alosaimi, W., & Alyami, H. (2021). *A New Initialization Approach in Particle Swarm Optimization for Global Optimization Problems*. 10.1155/2021/6628889
- Bangyal, W. H., Nisar, K., Ibrahim, A. A. B. A., Haque, M. R., Rodrigues, J. J. P. C., & Rawat, D. B. (2021). Comparative analysis of low discrepancy sequence-based initialization approaches using population-based algorithms for solving the global optimization problems. In *Applied Sciences (Switzerland)* (Vol. 11, Issue 16). doi:10.3390/app11167591
- Bangyal, W. H., Rauf, H. T., Batool, H., Bangyal, S. A., Ahmed, J., & Pervaiz, S. (2019). An improved Particle Swarm Optimization algorithm with Chi-Square mutation strategy. *International Journal of Advanced Computer Science and Applications*, 10(3), 481–491. doi:10.14569/IJACSA.2019.0100362
- Békési, J., Galambos, G., & Kellerer, H. (2000). A 5/4 linear time bin packing algorithm. *Journal of Computer and System Sciences*, 60(1), 145–160. doi:10.1006/jcss.1999.1667
- Beloglazov, A., Abawajy, J., & Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing. *Future Generation Computer Systems*, 28(5), 755–768. doi:10.1016/j.future.2011.04.017
- Beloglazov, A., & Buyya, R. (2012). Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers. *Concurrency and Computation*, 24(13), 1397–1420. doi:10.1002/cpe.1867
- Chen, T., Zhu, Y., Gao, X., Kong, L., Chen, G., & Wang, Y. (2018). Improving Resource Utilization via Virtual Machine Placement in Data Center Networks. *Mobile Networks and Applications*, 23(2), 227–238. doi:10.1007/s11036-017-0925-7
- Costello, K., & Rimol, M. (2021). *Gartner Forecasts Worldwide IT Spending to Grow 6.2% in 2021*. Gartner Newsroom. <https://www.gartner.com/en/newsroom/press-releases/2020-01-25-gartner-forecasts-worldwide-it-spending-to-grow-6-point-2-percent-in-2021>
- Donyagard Vahed, N., Ghobaei-Arani, M., & Souiri, A. (2019). Multiobjective virtual machine placement mechanisms using nature-inspired metaheuristic algorithms in cloud environments: A comprehensive review. *International Journal of Communication Systems*, 32(14), e4068. doi:10.1002/dac.4068
- Dupont, C., Giuliani, G., Hermenier, F., Schulze, T., & Somov, A. (2012). An energy aware framework for virtual machine placement in cloud federated data centres. *Proceedings of the 3rd International Conference on Future Energy Systems: "Where Energy, Computing and Communication Meet", e-Energy 2012*. doi:10.1145/2208828.2208832
- Energy demand data centers globally by type 2021. (2021). <https://www.statista.com/statistics/186992/global-derived-electricity-consumption-in-data-centers-and-telecoms/>
- Fan, X., Weber, W., & Barroso, L. A. (2007). Power provisioning for a warehouse-sized computer. *ACM SIGARCH Computer Architecture News*, 35(2), 13–23. doi:10.1145/1273440.1250665
- Farahnakian, F., Pahikkala, T., Liljeberg, P., Plosila, J., Hieu, N. T., & Tenhunen, H. (2016). Energy-aware VM Consolidation in Cloud Data Centers Using Utilization Prediction Model. *IEEE Transactions on Cloud Computing*, 7(2), 524–536. doi:10.1109/TCC.2016.2617374
- Gartner Inc. (2013). *Gartner Says by 2019, 90 Percent of Organizations Will Have Personal Data on IT Systems They Don't Own or Control*. <https://www.gartner.com/newsroom/id/2513615>

- Hammer, H. L., Yazidi, A., & Begnum, K. (2017). *An Inhomogeneous Hidden Markov Model for Efficient Virtual Machine Placement in Cloud Computing Environments*. 10.1002/for.2441
- Hieu, N. T., Di Francesco, M., & Yla-Jaaski, A. (2017). Virtual Machine Consolidation with Multiple Usage Prediction for Energy-Efficient Cloud Data Centers. *IEEE Transactions on Services Computing*, 13(1), 186–199. doi:10.1109/TSC.2017.2648791
- Huang, Z., & Tsang, D. H. K. (2012). SLA guaranteed virtual machine consolidation for computing clouds. *IEEE International Conference on Communications*, 1314–1319. doi:10.1109/ICC.2012.6363970
- Huang, Z., & Tsang, D. H. K. (2016). M-Convex VM Consolidation: Towards a Better VM Workload Consolidation. *IEEE Transactions on Cloud Computing*, 4(4), 415–428. doi:10.1109/TCC.2014.2369423
- Ibrahim, A., Noshay, M., Ali, H. A., & Badawy, M. (2020). PAPS0: A power-aware VM placement technique based on particle swarm optimization. *IEEE Access: Practical Innovations, Open Solutions*, 8, 81747–81764. doi:10.1109/ACCESS.2020.2990828
- Kaaouache, M. A., & Bouamama, S. (2015). Solving bin packing problem with a hybrid genetic algorithm for VM placement in cloud. *Procedia Computer Science*, 60(1), 1061–1069. doi:10.1016/j.procs.2015.08.151
- Kennedy, J., & Eberhart, R. (1995). *Particle Swarm Optimization*. Academic Press.
- Kennedy, J., & Eberhart, R. C. (1997). Discrete binary version of the particle swarm algorithm. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 5, 4104–4108. doi:10.1109/ICSMC.1997.637339
- Kusic, D., Kephart, J. O., Hanson, J. E., Kandasamy, N., & Jiang, G. (2008). Power and performance management of virtualized computing environments via lookahead control. *5th International Conference on Autonomic Computing, ICAC 2008*, 3–12. doi:10.1109/ICAC.2008.31
- Li, R., Zheng, Q., Li, X., & Yan, Z. (2017). Multi-objective optimization for rebalancing virtual machine placement. *Future Generation Computer Systems*. Advance online publication. doi:10.1016/j.future.2017.08.027
- Liu, X. F., Zhan, Z. H., Deng, J. D., Li, Y., Gu, T., & Zhang, J. (2018). An Energy Efficient Ant Colony System for Virtual Machine Placement in Cloud Computing. *IEEE Transactions on Evolutionary Computation*, 22(1), 113–128. doi:10.1109/TEVC.2016.2623803
- Liu, X. F., Zhan, Z. H., & Zhang, J. (2017). An energy aware unified ant colony system for dynamic virtual machine placement in cloud computing. *Energies*, 10(5), 609. Advance online publication. doi:10.3390/en10050609
- Mapetu, J. P. B., Kong, L., & Chen, Z. (2020). A dynamic VM consolidation approach based on load balancing using Pearson correlation in cloud computing. *The Journal of Supercomputing*. Advance online publication. doi:10.1007/s11227-020-03494-6
- Noshay, M., Ibrahim, A., & Ali, H. A. (2018). Optimization of live virtual machine migration in cloud computing : A survey and future directions. *Journal of Network and Computer Applications*, 110(March), 1–11. doi:10.1016/j.jnca.2018.03.002
- Pahlevan, A., Del Valle, P. G., & Atienza, D. (2016). Exploiting CPU-load and data correlations in multi-objective VM placement for geo-distributed data centers. *Proceedings of the 2016 Design, Automation and Test in Europe Conference and Exhibition, DATE 2016*, 1333–1338. doi:10.3850/9783981537079_0143
- Pervaiz, S., Ul-Qayyum, Z., Bangyal, W. H., Gao, L., & Ahmad, J. (2021). A Systematic Literature Review on Particle Swarm Optimization Techniques for Medical Diseases Detection. *Computational and Mathematical Methods in Medicine*, 2021, 1–10. doi:10.1155/2021/5990999 PMID:34557257
- Rahmanian, A. A., Ghobaei-Arani, M., & Tofighy, S. (2018). A learning automata-based ensemble resource usage prediction algorithm for cloud computing environment. *Future Generation Computer Systems*, 79, 54–71. doi:10.1016/j.future.2017.09.049
- Rajabzadeh, M., & Toroghi, A. (2017). Energy-aware framework with Markov chain-based parallel simulated annealing algorithm for dynamic management of virtual machines in cloud data centers. *The Journal of Supercomputing*, 73(5), 2001–2017. doi:10.1007/s11227-016-1900-y

- Satpathy, A., Addya, S. K., Turuk, A. K., Majhi, B., & Sahoo, G. (2018). Crow search based virtual machine placement strategy in cloud data centers with live migration. *Computers & Electrical Engineering*, 69, 334–350. doi:10.1016/j.compeleceng.2017.12.032
- Sharma, O., & Saini, H. (2019). Energy and SLA Efficient Virtual Machine Placement in Cloud Environment Using Non-Dominated Sorting Genetic Algorithm. *Energy and SLA Efficient Virtual Machine Placement in Cloud Environment Using Non-Dominated Sorting Genetic Algorithm.*, 13(1), 1–16. doi:10.4018/IJISP.2019010101
- Srikantaiah, S., Kansal, A., & Zhao, F. (2008). Energy aware consolidation for cloud computing. *Workshop on Power Aware Computing and Systems, HotPower 2008*.
- Tang, M., & Pan, S. (2015). A Hybrid Genetic Algorithm for the Energy-Efficient Virtual Machine Placement Problem in Data Centers. *Neural Processing Letters*, 41(2), 211–221. doi:10.1007/s11063-014-9339-8
- Tchana, A., De Palma, N., Safieddine, I., & Hagimont, D. (2016). Software consolidation as an efficient energy and cost saving solution. *Future Generation Computer Systems*, 58, 1–12. doi:10.1016/j.future.2015.11.027
- Teyeb, H., Balma, A., Tata, S., & Ben Hadj-Alouan, N. (2017). Traffic-aware virtual machine migration scheduling problem in geographically distributed data centers. *IEEE International Conference on Cloud Computing, CLOUD*, 798–801. doi:10.1109/CLOUD.2016.108
- Tripathi, A., Pathak, I., & Vidyarthi, D. P. (2018). Energy Efficient VM Placement for Effective Resource Utilization using Modified Binary PSO. *The Computer Journal*, 61(6), 832–846. doi:10.1093/comjnl/bxx096
- Tseng, F.-H., Chen, C.-Y., Chou, L.-D., Chao, H.-C., & Niu, J.-W. (2015). *Service-Oriented Virtual Machine Placement Optimization for Green Data Center*. 10.1007/s11036-015-0600-9
- Wang, H., & Tianfield, H. (2018a). Energy-Aware Dynamic Virtual Machine Consolidation for Cloud Datacenters. *IEEE Access: Practical Innovations, Open Solutions*, 6(c), 15259–15273. doi:10.1109/ACCESS.2018.2813541
- Wang, H., & Tianfield, H. (2018b). Energy-Aware Dynamic Virtual Machine Consolidation for Cloud Datacenters. *IEEE Access: Practical Innovations, Open Solutions*, 6, 15259–15273. doi:10.1109/ACCESS.2018.2813541
- Xiao, H., Hu, Z., & Li, K. (2019). Multi-Objective VM Consolidation Based on Thresholds and Ant Colony System in Cloud Computing. *IEEE Access: Practical Innovations, Open Solutions*, 7, 53441–53453. doi:10.1109/ACCESS.2019.2912722
- Xu, G., Ding, Y., Zhao, J., Hu, L., & Fu, X. (2013). A novel artificial bee colony approach of live virtual machine migration policy using bayes theorem. *TheScientificWorldJournal*, 2013, 1–13. Advance online publication. doi:10.1155/2013/369209 PMID:24385877
- Yan, J., Zhang, H., Xu, H., & Zhang, Z. (2018). Discrete PSO-based workload optimization in virtual machine placement. *Personal and Ubiquitous Computing*, 22(3), 589–596. doi:10.1007/s00779-018-1111-z
- Zeng, D., Guo, S., Huang, H., Yu, S., & Leung, V. C. M. (2015). Optimal VM placement in data centres with architectural and resource constraints. *International Journal of Autonomous and Adaptive Communications Systems*, 8(4), 392–406. doi:10.1504/IJAACS.2015.073187
- Zhang, F. A. W. (2019). Energy-efficiency virtual machine placement based on binary gravitational search algorithm. *Cluster Computing*, 0. Advance online publication. doi:10.1007/s10586-019-03021-0
- Zhang, L., Zhuang, Y., & Zhu, W. (2013). Constraint Programming based Virtual Cloud Resources Allocation Model. *International Journal of Hybrid Information Technology*, 6(6), 333–344. doi:10.14257/ijhit.2013.6.6.30
- Zheng, Q., Li, R., Li, X., Shah, N., Zhang, J., Tian, F., Chao, K. M., & Li, J. (2016). Virtual machine consolidated placement based on multi-objective biogeography-based optimization. *Future Generation Computer Systems*, 54(March), 95–122. doi:10.1016/j.future.2015.02.010
- Zhou, Z., Hu, Z., & Li, K. (2016). *Virtual Machine Placement Algorithm for Both Energy-Awareness and SLA Violation Reduction in Cloud Data Centers*. Academic Press.

APPENDIX A - ABBREVIATIONS

Table 17. Abbreviations and their meanings

Abbreviation	Meaning
VM	Virtual Machine
PM	Physical Machine
PSO	Particle Swarm Optimization
CPU	Central Processing Unit
RAM	Random Access Memory
BW	Bandwidth
BPSO	Binary PSO
λ_j	Energy efficiency of a single host j
$Host_{list}$	List of hosts
$VirMach_{list}$	List of VMs ready for migration
$OLHost_{list}$	List of Overloaded hosts
$AvailHost_{list}$	List of hosts excluding overloaded hosts
N_p	Population size
N_d	Dimension count (Number of migrated VMs)
N_t	Iteration count
c_1	Cognitive parameter
c_2	Social parameter
w	Inertia weight coefficient
Υ	Time varying variable
Υ_{min}	Minimum value of Υ

Table 17 continued on next page

Table 17 continued

Abbreviation	Meaning
Υ_{max}	Maximum value of Υ
FF	Value of fitness function
P	Swarm of particles
$gBestValue$	Best FF value among all particles
$pBestValue$	Best FF value of a particle
$pBestPosition$	Position of a particle with best FF value
$gBestPosition$	Best position for VMs among the swarm
p	A specific particle
t	The current iteration
r_1	A random value between 0 and 1
r_2	A random value between 0 and 1
r_3	A random value between 0 and 1
r_4	A random value between 0 and 1
X_t	Current position of a particle
X_{t+1}	Particle new position
V_t	Speed of a particle in current iteration
V_{t+1}	Speed of a particle in the next iteration

Table 17 continued on next page

Table 17 continued

Abbreviation	Meaning
S_1	A Sigmoid function
S_2	A Sigmoid function
HMM	Hidden Markov Model
ABC	Artificial Bee Colony
MIPS	Million Instruction per Second
SLA	Service Level Agreement
SLAV	SLA Violation
LR	Linear Regression
IQR	Inter Quartile Range
MAD	Median Absolute Deviation
THR	Static Threshold
MMT	Minimum Migration Time
MC	Maximum Correlation
MU	Minimum Utilization
RS	Random Selection
PABFD	Power Aware Best Fit decreasing
DVMC	Dynamic Virtual Machine Consolidation
ESV	Energy and SLAV
ESM	Energy, SLAV and number of VM Migration
SLAPDM	SLA Performance Degradation due to Migration
SLATAH	SLA Time per Active Host

Bhagyalakshmi Magotra pursued a Bachelor of Computer Science from the University of Jammu, India in 2010 and a Master of Computer Science and Engineering from Shri Mata Vaishno Devi University, India in the year 2013. She is currently pursuing a Ph.D. from the Department of Computer Science & Information Technology, School of Applied Sciences from the Central University of Jammu. Her main research work focuses on routing in Computer Networks and Cloud Computing.

Deepti Malhotra pursued a Bachelor of Computer Science from the University of Jammu, India in 2005, an M.E. Computer Science from the Thapar University, India in the year 2007, and completed her Ph.D. from the University of Jammu in 2013. She is currently working as Assistant Professor, in the Department of CS&IT, Central University of Jammu, J&K, India. Having a teaching and research experience of 10 years, she has published more than 37 research papers in reputed international journals and conferences. She is a Member of the Institution of Engineers (India) and a Life Time Member of All India Science Congress. In 2019, she has also organized an AICTE training and learning academy (ATAL) Faculty Development Programme(FDP) on the topic "Artificial Intelligence". Her main research work focuses on Grid Computing, Cloud Computing, Machine Learning and Natural Language Processing.