An Energy Efficient Particle Swarm Optimization based VM Allocation for Cloud Data Centre: EEVMPSO

Abhishek Kumar Pandey[1] *, Sarvpal Singh[1]

[1]Department of Information Technology and Computer Application, Madan Mohan Malaviya University of Technology, Gorakhpur, (UP.) India.

Abstract

Cloud computing has witnessed exponential growth in recent years, resulting in a significant surge in energy consumption and operational costs of cloud data centres. Efficiently allocating Virtual Machines (VMs) within these data centres is crucial to achieve energy efficiency and optimize resource utilization. System instability may result from repeated requests for computing resources. One of the most critical difficulties facing virtualization technology is finding the best way to stack virtual machines on top of physical devices in cloud data centers. The host must move virtual machines from overloaded to underloaded hosts as part of load balancing, which has an impact on energy consumption. We propose energy-efficient particle swarm optimization algorithm (EEVMPSO) for Virtual Machine allocation is designed to maximize the load balancing. System resources, including CPU, storage, and memory, are optimized using EEVMPSO. The energy-aware virtual machine migration using the Particle Swarm Optimization Algorithm for dynamic VMs placement and energy-efficient cloud data centers. We conducted extensive experiments and simulations to evaluate the performance of the proposed algorithm in comparison to existing VM allocation methods. The results demonstrate the superiority of our approach in achieving energy efficiency and resource optimization. The experimental result shown in the proposed method, consumption energy in comparison to the PAPSO, KHA, EALBPSO, and RACC-MDT algorithm by 10.86%, 18.22%, 25.8%, and 31.34%, respectively, demonstrated the improvements in the service level agreements violation 5.77%, 15.3%, 26.19%, and 30.4%, as well as the average CPU utilization 2.2%, 24%, 22.6%, and 14.6%.

Keywords: Particle Swarm Optimization algorithm, Cloud Computing, Virtual Machine Placement, Cloud data center, Service level agreements

Received on 15 April 2023, accepted on 30 July 2023, published on 01 August 2023

doi: 10.4108/eetsis.3254

*Corresponding author. Email: akpsiet@gmail.com

1. Introduction

The increasing demand for cloud computing services has led to the rapid expansion of cloud data centres worldwide. While cloud computing offers unparalleled scalability and flexibility, it also presents significant challenges, particularly in terms of energy consumption and resource utilization. Cloud data centres are known for their substantial energy requirements, leading to rising operational costs and a significant environmental impact [1-2]. Cloud computing is more popular as its widespread acceptance has increased. The implications of cloud computing for everyday life are imminent (e.g., social networks, sensor networks, etc.). More and more people are turning to the cloud model due to the popularity of smart gadgets. Rapid expansion in the number and scale of cloud data centres is occurring [3-5]. Recently, high performance in cloud infrastructure has been a primary issue and accomplished without putting a primary emphasis on the amount of energy used in cloud environments [6-8]. On the other hand, the cloud environment requires the data center to host the cloud applications [9].

Optimizing the allocation of Virtual Machines (VMs) within cloud data centres is a critical aspect of achieving energy efficiency and enhancing overall resource utilization. Efficient VM allocation not only reduces energy consumption but also improves the overall performance and cost-effectiveness of cloud services [10-12]. The best way to deploy virtual machines (VMs) on bare metal servers in a cloud data center is an important pressing concern in cloud computing. Minimizing power usage and resource wastage may be achieved by strategically deploying virtual machines on real devices [13-14]. Therefore, appropriate optimization techniques are used for virtual machine placement in a critical and pivotal step toward the so-called goal. Placement algorithms for virtual machines employ optimization techniques to partition actual hardware fairly. Resulted, fewer physical computers are required to do the same amount of work in the cloud data center, less waste is produced, and the use of computing resources is increased. Recently, many techniques for locating and deploying virtual machines have been suggested [15-17].

This article proposes a new algorithm for efficiently allocating computational resources by determining where to place virtual machines on physical machines. In the first step, the proposed algorithm seeks to lessen the load on cloud data centers' infrastructure by decreasing the number of online computers. The optimal placement of virtual machines on physical machines in cloud data centers is the second goal of the proposed scheme. The third goal is to reduce the service level agreements (SLAs) of cloud data centers' active physical machines. Also taken into account were reductions in the use of central processing units, memory, bandwidth, storage space, running computers, and migrations. The suggested solution made fewer changes in the cloud data center [18].

This proposed work presents a Particle Swarm Optimization-based approach for energy-efficient VM (EEVMPSO) to reduce data center energy usage, and it consolidates VMs onto as few servers as possible while still protecting QoS for newly added customers. The suggested method utilizes a metaheuristic algorithm, Particle Swarm Optimization (PSO). Combining local and global search may provide an almost ideal virtual machine placement solution. To determine the ideal VM location, EEVMPSO uses a minimized fitness function. The suggested method is implemented in CloudSim and tested with random workloads on various-sized virtual machines and power management units (PMs) before being compared to the existing state-of-the-art algorithms.

The main contribution of the proposed technique is:

1. VM placement is an optimization problem to maximize system efficiency. The transfer function should show time-varying behavior to avoid reaching a local optimum.
2. EEVMPSO finds the near-optimal solution using a viable minimization fitness function. Weighted sum merges active and overloaded hosts. It uses PSO to find local and global solutions.
3. PSO algorithm reduces the VM migrations, consumption of energy, average CPU utilization, and host shutdowns by 19.16%, 25.8%, 22.6% and 55.45% respectively over EALBPSO algorithm.
4. EEVMPSO outperforms PAPSO [17], KHA [12], EALBPSO [20], and RACC-MDT [2] by attaining 5.77%, 15.3%, 26.19% and 30.4% on the combined performance indicator of Energy SLA Violation, which measures both energy consumption and SLA compliance.

The rest of the paper is outlined as follows: the relevant literature is discussed in Section 2, the proposed method is described in Section 3, and the simulation results and a conclusion and suggestions for future study are presented in Sections 4 and 5, respectively.

2. Literature Review

This section includes previous research on the selected virtual machine (VM) consolidation techniques, as well as a description of the main properties of each. A method for consolidating virtual machines based on the PSO algorithm could optimally and rapidly converge on a mapping from virtual machine to power management unit [19-20]. The authors have been concentrating their efforts on finding a solution to the issue of VM consolidation, which has been recast as a problem of variable-sized bin packing. In addition, the author has been considering the frequency that is readily accessible. The collected data show that, compared to the other meta-heuristics techniques, the machine is suitable in terms of its capacity to reduce the number of PMs. However, a notable drawback is that it has prioritized bandwidth enhancement, neglecting other crucial resources such as memory and processing power.

Masoudi et al. [20] suggested a unique EALBPSO approach for objective functions-based performance evaluation. This strategy can maximize the best possible Standard Deviation load balancing of processors while reducing the system's total energy consumption. It provides a complete method for VM allocation based on an evolutionary algorithm. It can efficiently converge on a VM-to-PM mapping while maintaining the highest feasible level of energy efficiency. The initial allocation of virtual machines is suggested to employ a novel optimization technique called EALBPSO. Maximum efficiency is required in the data center's energy use and the load distribution among its CPUs.

Basu et al. [21] provided a strategy for improving Genetic Algorithm (GA) because the poor VM placement method wastes memory and energy. The study showcased a new and efficient evolutionary technique for VM allocation that can potentially minimize energy consumption while accommodating a larger number of reserved VMs. Alharbi et al. [22] suggested an Ant Colony Optimization (ACO) method for solving the optimization problem of arranging VMs on physical machines in a data center in an energy-efficient manner. Consolidating the virtual machines (VMs) into fewer physical machines might improve energy efficiency, and this approach may make it easier to quickly find an appropriate allocation solution for reserved VMs [23]. The VM is constructed depending on the application's needed resources and system operations. The virtual machine (VM) is placed on one of the available servers per the placement strategy. It is a significant challenge to figure out how to allocate virtual machines (VMs) to appropriate servers to cut down on energy use. Ant colony optimization, genetic algorithms (GA), particle swarm optimization (PSO), and whale optimization algorithm (WOA) are the population-based methods of VMP [24-25].

Tseng et al. [26] proposed the idea of using a multi-objective genetic algorithm (GA) for the aim of resource prediction and allocation. This GA forecasts the demand for resources before allocating VMs to maximize the utilization of resources and the consumption of energy. The GA predicts VM resource demand, followed by energy-efficient server allocation. Cloud computing uses virtual machines; whenever a user requests a cloud-based application, the VM is created to accomplish the request [27].

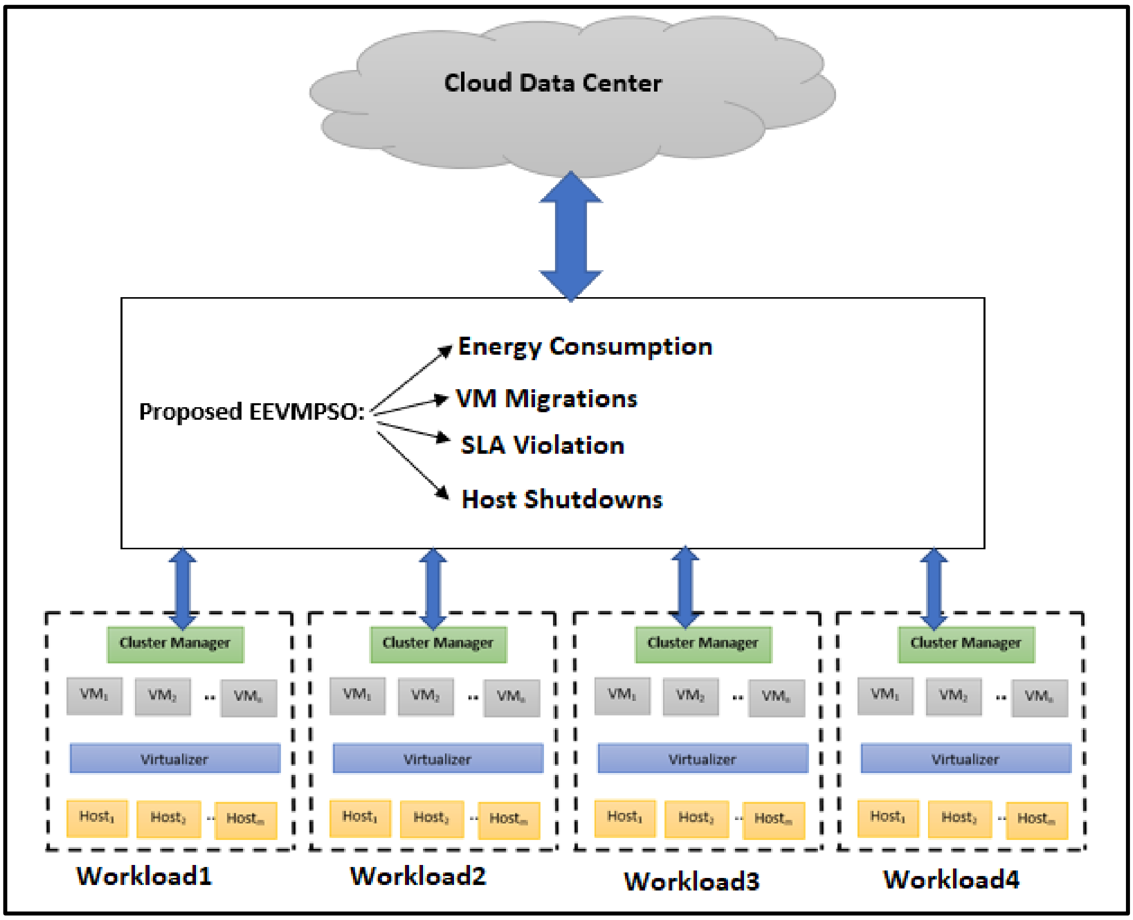

Figure 1. The architecture of EEVMPSO system

Sharma et al. [28] introduced an HGAPSO (hybrid method) for VM allocation in the cloud data center. The goal of this approach was to reduce resource waste and SLA violations. The virtual machine allocation is encoded using this method as a particle vector, where the value of a bit is set to 1 if the server is active and 0 otherwise. Since the bit value of the velocity vector is simply dependent on the presence or absence of VMs, this approach is suitable for application in situations with homogenous VMP. This approach is unsuitable for scenarios with heterogeneous VMP since it does not accurately encode the number and kind of VMs.

Shun Yao et al. [29] devised multi-objective multi-swarm optimization to optimize data center process scheduling. The data center's infrastructure is used as a basis for searching for non-dominated scheduling solutions. Particles in separate swarms can interact so that the shared information across the swarms. Multiple swarms were established to accomplish diverse goals. However, the Pareto-based multi-objective PSOCM method that the system proposes requires minimal parameters and has a naturally better means of communication amongst its constituent particles. Compared to other optimization methods like GA and ACO, this aids in delivering faster convergence.

Dahsti et al. [30] devised a solution to meet the technologies' and consumers' needs. A Platform-as-a-Service (PaaS) solution was implemented during the inquiry to facilitate the scheduling of client responsibilities. If the needs for the physical machines and the users' expectations are not compatible in the cloud, then excessive energy use and a trade-off between energy and performance may arise, which will restrict the provider's profitability [31]. PSO enhances energy efficiency without sacrificing service. These techniques reallocated migrating virtual machines to a full host. According to the findings of the simulations carried out in CloudSim, the conditions of the simulation were quite similar to those of the actual environment [32].

Ibrahim et al. [33] projected PAPSO to decrease energy consumption used in data centers by condensing virtual machines onto the fewest number of servers possible. Nevertheless, it considers client QoS as the service. The recommended strategy uses a technique known as Particle Swarm Optimization or PSO for short. It can offer the capability of both local and worldwide search, which is one of the contributing reasons that lead to the realization of the near-optimal VM placement solution. Another contributing aspect is that it can simultaneously deliver local and global searches. PAPSO employs a minimization fitness function to solve the placement of virtual machines that come with near-optimal solutions. The proposed approach is implemented in CloudSim, where compared to the PABFD using randomized workloads executed on VMs and PMs of different sizes [34].

Table 1 represents the Metaheuristic optimization approaches and their constraints. We then discussed the method objective and research gap that use metaheuristics like WAO, PSO, ABC, GA, etc. Table 1 shows research on the different methods used for VM allocation in cloud data centers, whereas the PSO method has been used to improve the performance of cloud data centers.

Table 1: Study reference on optimization techniques

Study references	Optimization Technique	Targeted attributes	Method Objectives	Research Gap
Gomathi et al. (2022) [35]	PSOCM method	CPU, PDM, SLAT	Reducing consumption of energy and SLA violation.	The Pareto method is time-consuming yet produces the best possible results.
Al-Moalmi et al. (2021) [36]	Whale Optimization Algorithm (WAO)	CPU and memory	Minimizing overhead and energy consumption.	There have been gaps in the evaluation of many significant metrics, including migration time and SLA violation.
Liu et al. (2016) [37]	Ant Colony Optimization (ACO)	CPU and memory	Conservation of resources and use of less energy.	The quality of service that is provided by the virtual machine resources is not taken into consideration by the technique.
Li et al. (2018) [38]	Artificial Bee Colony (ABC)	CPU, memory, and bandwidth	Increasing resource productivity while also reducing lower the number of relocations	Low scalability and a large processing cost are problems associated with PM overload risk estimation.
Kim et al. (2019) [39]	Harmony Search (HS)	CPU and memory	Energy efficiency and a decrease in the number of virtual machine migrations.	It disregards migration latency, migration overhead, and SLA violations.
Riahi et al. (2018) [40]	Genetic Algorithm (GA)	CPU and memory	Minimizing resource waste and the number of active PMs	Attempts to deal with huge data have not proven productive.
Sasan et al. (2021) [41]	Hybrid Algorithm ((Sine– Cosine Algorithm and Salp Swarm Algorithm)	CPU, memory, and bandwidth	Minimize SLA among active cloud physical machines.	It utilizes a fixed strategy to place virtual machines and does not provide a useful method for striking a balance between energy efficiency and cost.
Yavari et al. (2019) [42]	Firefly Algorithm (FA)	CPU, memory, and bandwidth	Reducing the number of migrations, SLA violations, and energy use.	The scalability and optimization of resource use of suggested algorithms have not been verified.

Wang et al. [43] introduced a novel approach known as local search-based genetic algorithm (LSGA), which incorporates a genetic algorithm (GA) along with a unique local search technique. Initially, the system employs a matrix coding scheme to represent individuals and subsequently formulates the appropriate crossover and mutation operations. The performance of LSGA was evaluated by comparing it with several state-of-the-art algorithms on Sudoku puzzles of varying difficulty levels. Yang et al. [44] introduced a framework called bi-directional feature fixation (BDFF) for particle swarm optimization (PSO). This framework presents a unique approach to mitigate the search space in the context of large-scale feature selection. The BDFF algorithm employs a dual search strategy, utilizing two opposing directions, in order to effectively guide particles in the exploration of feature subsets of varying sizes. By employing two distinct search directions, BDFF is able to address the selection states of certain features and subsequently prioritize others during the particle update process, thereby effectively reducing the overall search space.

Ge et al. [45] introduced a distributed segment-based genetic algorithm (DSGA) that effectively addresses the issues of data privacy, communication cost, and load balance. In order to safeguard privacy, this study proposes a digit-based anonymity strategy that leverages attribute characteristics. This approach aims to preserve information integrity while enabling fuzzy identification. Subsequently, a three-tier distributed framework is introduced for the purpose of enhancing search efficiency in multi-objective optimization and attaining a balance between communication cost and load distribution.

Li et al. [46] introduced a novel three-layer framework called DDE-ARA, which incorporates adaptive resource allocation. This framework consists of three layers: the algorithm layer, responsible for the evolution of diverse differential evolution (DE) populations; the dispatch layer, which allocates individuals from the DE populations to different distributed machines; and the machine layer, which facilitates the utilization of distributed computers. Within the DDE-ARA framework, three innovative approaches are additionally suggested. The ultimate objective is to enhance the search efficiency of the system. Ge et al. [47] proposed a multitasking distributed differential evolution method. The presented system facilitates communication across various database fragmentation issues by sharing general and efficient allocation data.

In conclusion, energy efficiency is a critical concern in cloud data centres, and VM allocation plays a vital role in optimizing resource usage. Particle Swarm Optimization (PSO) has shown promise in solving complex optimization problems and can be adapted to address the energy-efficient VM allocation problem in cloud data centres. Our proposed EEVMPSO approach aims to contribute to the growing body of research focused on achieving sustainable and energy-efficient cloud infrastructures.

3. System Model: EEVMPSO

The EEVMPSO technique is selected for the data center with various processing metrics including the energy consumption, VM migration, CPU utilization, SLA violation, energy SLA violation, and host shutdown. Memory and processing power define a machine. Users request a VM with needed resources via the data center user interface. VMs are assigned to the first available physical machine based on resource needs. Datacentre managers need to reduce energy consumption by relocating VMs from idle physical equipment and shutting off passive machines.

Table 2 listed the abbreviation utilized in this study and their short descriptions to help understand the systems model and suggested method.

Table 2: Abbreviation

Notations	Description
EEVMPSO	*Energy efficient particle swarm optimization algorithm for virtual machine*
PAPSO	*Power-aware technique depending on particle swarm optimization*
KHA	*Kill herd algorithm*
EALBPSO	*Energy-aware load balancing particle swarm optimization*
RACC-MDT	*Residual available computing capacity and minimum data transfer*
CPU_{U_i}	*CPU utilization rate*
P_{max}	*The power used at maximum capacity*
$Host_{P_i}$	*The load placed on a physical machine in a cloud service*
$VTHSLA$	*Time per active host in SLA violation*
$Total_{ti}$	*The total time utilize for a server*
TA_{Host_i}	*The time for active host*
PDM	*Performance degradation of migration*
M_{VM}	*Total number of VMs*
C_d	*VMs Performance degradation in CPU utilization operations*
C_r	*Total number of CPUs requested in the particular operation*
$VSLA$	*SLA violation*
P_{m_i}	*Usage of memory i^{th} physical machines*
U_{m_i}	*Usage of average memory in active physical machines*
s_i	*Number of servers i*
CPU_{mig}	*CPU utilization for migrated VMs*
RAM_{mig}	*Memory utilization for migrated VMs*
BW_{mig}	*Bandwidth utilization for migrated VMs*
$X_i(t)$	*Particle positions at time t*
$V_i(t)$	*Particle velocities at time t*
L_{ibest}	*Best-possible prior location*
c_1, c_2	*Acceleration coefficient*
r_1, r_2	*Random integers range from 0 to 1*
G_{ibest}	*Best-possible current location*
P_i	*Number of particles i*
VM	*Virtual machine*
$\boldsymbol{I_t}$	*Maximum number of iterations*

3.1 VM Placement

Allocation in suitable destinations for migrated virtual machines is crucial to effective VM consolidation. In the circumstances involving VM consolidation, the server judged to be the most appropriate for accepting the migrated VM is selected. Despite this, the difficulty of VM placement is not restricted to just these kinds of circumstances. Another scenario for virtual machine deployment involves searching for a suitable host that will accept the virtual machine as an initial placement. We suggest a strategy for the placement of virtual machines from the perspective of virtual machine consolidation and attention to previous publications that have addressed VM consolidation. Numerous research with varying aims and approaches have been offered to develop efficient methods. Fig. 1 clearly shows a broad classification of VM placement strategies based on their underlying approaches, their goals, and the number of these goals. Some methods seek the optimal mapping that accomplishes a single aim, while others seek the best mapping that accomplishes several goals. These goals might include bettering the quality of service offered to customers or cutting expenses for service providers. The recognized method has been employed by various techniques, all of which use meta-heuristic algorithms to decide which PM will get the migrated VM. First Fit (FF) is one example of a greedy heuristic approach that may be used, in which each virtual machine is assigned to the first host that meets its requirements. The BF method uses a mapping between VMs and the best possible PMs to suit the data.

3.2 Energy Model

The duration for which a computer's processor and memory are utilized directly affects the level of energy the machine requires. The central processing unit is the primary consumer of power in modern gadgets. The degree to which the frequency of CPU usage employs a computer's resources. Applying Equation (1), we can calculate an energy consumption model for a virtual machine that accounts for the processor's usage in light of the machine's present workload.

$$CPU_{U_i} = \sum_{j=1}^{n} CPU_{U_{i,j}} \tag{1}$$

Where P_{max} is the power used at maximum capacity, f is defined as the energy used. At the same time, the device is unused, and CPU_{U_i} is the CPU utilization rate (CPU_{U_i} [0, 1]) determined by the load placed on the $Host_{P_i}$ as a result of running a cloud service by Equation (2).

$$Host_{P_i} = f * P_{max} + (1 - f) * P_{max} * CPU_{U_i} \tag{2}$$

$$E = \int_{t-1}^{t} Host_{P_i}(CPU_{U_i}(t))dt \tag{3}$$

$CPU_{U_i}(t)$ is the power consumed at time t, and E is defined as the total power used by the device for the period $[t-1, t]$ specified in Equation (3).

$$VTHSLA = \frac{1}{S}\sum_{i=0}^{M-1}\frac{Total_{ti}}{TA_{Host_i}} \tag{4}$$

Where in the above Equation (4), $VTHSLA$ is calculated for time per active host in SLA violation, $Total_{ti}$ represents the total time utilized for a server, TA_{Host_i} represents the time for the active host, and S is defined as the total number of servers.

$$PDM = \frac{1}{M_{VM}}\sum_{i=0}^{M_{VM}-1}\frac{C_d}{C_r} \tag{5}$$

In Equation (5), PDM is defined as the metric to measure the degradation of migration performance. C_d represents the CPU utilization to calculate the VMs performance degradation, C_r represents the total number of CPUs requested in the particular operation, and M_{VM} represents the total number of VMs.

$$VSLA = VTHSLA * PDM \tag{6}$$

In Equation (5), $VSLA$ represents the SLA violation to estimate the multiplication value of $VTHSLA$ and PDM.

3.3 Proposed EEVMPSO

A virtual machine may migrate to any of several different host machines. Concurrently, power consumption may change as a result of VM movement. Consequently, optimizing performance and decreasing energy consumption is crucial to appropriately place and organize VM on host systems and powering down those not in use. In this case, n virtual machines (VMs) will share m physical hosts. The difficulty is in formulating a model for transferring the VMs across systems. Additionally, this design can reduce energy consumption and increase VM transfer capacity.

The relocation of virtual machines as part of an overall attempt to consolidate is an example of the second kind of virtual machine placement. Let the VMs = {VM1, VM1……. VMn} and List of Hosts = {PM1, PM2…… PMm}
The goal is to assign each migrated virtual machine to its corresponding physical host using equations (7) to (9).

$$\sum_{i=0}^{k-1} CPU_i + CPU_{mig} < CPU_j \tag{7}$$

$$\sum_{i=0}^{k-1} RAM_i + RAM_{mig} < RAM_j \tag{8}$$

$$\sum_{i=0}^{k-1} BW_i + BW_{mig} < BW_j \tag{9}$$

The determination of the crucial resources required for the migration of a virtual machine (VM) involves identifying the central processing unit (CPU), random access memory (RAM), and bandwidth as indispensable components.

Furthermore, a PM cannot host the same VM simultaneously; hence each VMi can only be associated with a single physical host. The approach that has been proposed reflects the mapping of migrated VMs in the form of Equation (10).

$$V_{map} = \begin{cases} 1, & VM\ and\ PM\ belongs\ to\ the\ List\ of\ VM\ and\ Host \\ 0, & otherwise \end{cases} \tag{10}$$

Fitness Function: A fitness function is derived from the factors that affect the solution's quality to determine whether the solution can provide a desirable outcome. The value of each solution produced by the proposed algorithm is calculated using this function, and the most optimal solution is identified as having the highest or lowest value, respectively, based on the parameter placement approach. Therefore, the fitness function to measure the efficacy of the optimum result is given by equation (11).

$$F = \sum_{i=0}^{n-1} \left(\frac{\left(Host_{P_i} - CPU_{U_i}\right)^2}{Host_{P_i}} \right) + \left(\frac{\left(P_{m_i} - U_{m_i}\right)^2}{U_{m_i}} \right) * s_i \quad (11)$$

Where $Host_{P_i}$ is defined as the usage of processors in ith physical machines. CPU_{U_i} is defined as the usage of the average processor in active physical machines. P_{m_i} represents the usage of memory ith physical machines. U_{m_i} represents average memory usage in active physical machines, and s_i is defined as the ith server is used (i=1 means one server is used or i=0 not used).

Illustration: Each particle in the initial mapping generated by EEVMPSO has a small number of dimensions. The number of virtual machines (VMs) to be migrated is the same as the number of dimensions. For each axis, there is a number that represents the virtual machine (VM) index. A PM can host several virtual machines, but the inverse is false. Each combination of the virtual machine to host has been tested until the optimal fitness function has been found. Below, we examine an illustrative instance of initial mapping generation. The underlying presumptions are:

Host [10] = {Host0, Host1, Host2, Host3, Host4, Host5, Host6, Host7, Host8, Host9}

Migrated VM [5] = {VM1, VM2, VM3, VM4, VM5}

Available Host [7] = {Host0, Host2, Host3, Host5, Host6, Host7, Host8}

Overloaded Host [3] = {Host1, Host4, Host9}

The initial phase arrangement of the particle placements in ascending order of VM index is shown in Table 3, and the particles represent 0 and 1 matrix in initial mapping.

Table 3: VMs particle representation

	VM_1	VM_2	VM_3	VM_4	VM_5
$HOST_0$	1	0	0	0	0
$HOST_1$	0	0	0	0	0
$HOST_2$	0	0	1	0	0
$HOST_3$	0	1	0	0	0
$HOST_4$	0	0	0	0	0
$HOST_5$	1	0	0	0	0
$HOST_6$	0	0	0	1	0
$HOST_7$	0	1	0	0	0
$HOST_8$	0	0	0	0	1
$HOST_9$	0	0	0	0	0

Tables 4, 5, 6, 7, 8, 9, and 10 show, in order of scenario, the random sequences assigned to available hosts based on first fit.

Table 4: VM placement Scenario 1

Scenario 1				
VM_1	VM_2	VM_3	VM_4	VM_5
$Host_0$	$Host_3$	$Host_5$	$Host_7$	$Host_8$

Table 5: VM placement Scenario 2

Scenario 2				
VM_1	VM_2	VM_3	VM_4	VM_5
$Host_0$	$Host_2$	$Host_6$	$Host_7$	$Host_8$

Table 6: VM placement Scenario 3

Scenario 3				
VM_1	VM_2	VM_3	VM_4	VM_5
$Host_2$	$Host_3$	$Host_0$	$Host_6$	$Host_7$

Table 7: VM placement Scenario 4

Scenario 4				
VM_1	VM_2	VM_3	VM_4	VM_5
$Host_3$	$Host_0$	$Host_6$	$Host_7$	$Host_5$

Table 8: VM placement Scenario 5

Scenario 5				
VM_1	VM_2	VM_3	VM_4	VM_5
$Host_2$	$Host_3$	$Host_8$	$Host_6$	$Host_7$

Table 9: VM placement Scenario 6

Scenario 6				
VM_1	VM_2	VM_3	VM_4	VM_5
$Host_8$	$Host_7$	$Host_0$	$Host_2$	$Host_3$

Table 10: VM placement Scenario 7

Scenario 7				
VM_1	VM_2	VM_3	VM_4	VM_5
$Host_7$	$Host_8$	$Host_5$	$Host_3$	$Host_0$

Table 11: Migrated VM mapping

VM_1	VM_2	VM_3	VM_4	VM_5
$Host_0$	$Host_3$	$Host_5$	$Host_7$	$Host_8$
$Host_0$	$Host_2$	$Host_6$	$Host_7$	$Host_8$
$Host_2$	$Host_3$	$Host_0$	$Host_6$	$Host_7$
$Host_3$	$Host_0$	$Host_6$	$Host_7$	$Host_5$
$Host_2$	$Host_3$	$Host_8$	$Host_6$	$Host_7$
$Host_8$	$Host_7$	$Host_0$	$Host_2$	$Host_3$
$Host_7$	$Host_8$	$Host_3$	$Host_6$	$Host_0$

Table 11 represents the particles (migrated VM) consisting of the number of PM indexes in the list of available hosts.

3.4 Problem Formulation

The PSO algorithm is utilized to acquire a suitable system for distributing assets across physical hosts. Congestion of the system's load at any one time reduces energy usage in data centers, which helps to offset the algorithm's temporal complexity. An allocation of virtual machines to physical hosts is an NP-Hard issue. Therefore. The most effective approach is to use a metaheuristic algorithm. The present analysis employed the PSO method as its primary approach. The PSO algorithm, among other comparable algorithms, has been shown to have the highest performance and is thus employed in addressing these issues. For this reason, this method was designed to speed things up and to provide a more practical solution with a better function than existing methods.

EEVMPSO iterates numerous times to get to a nearly optimum VM location. In each cycle, the velocity of each particle is updated, allowing for the updating of its location. An individual particle's current position is compared to the best position for the particle and the swarm to establish the amount of velocity adjustment to be applied. New coordinates and speed are truncated to the nearest integer before being utilized to choose a suitable host from the available ones. Upon establishing the novel coordinates, it is possible to reevaluate the fitness function of every individual particle. The results of these calculations guide particles in their search for the best possible solutions.

$$X_i(t) = x_1, x_2 \dots \dots x_{iD} \tag{12}$$

Particle positions at time t are expressed as vectors using Equation (12) and Equation (13) in iteration t also shows that each particle has a velocity represented by a vector.

$$V_i(t) = v_1, v_2 \dots \dots v_{iD} \tag{13}$$

$$L_{ibest} = l_1, l_2 \dots \dots l_{iD} \tag{14}$$

Vector L_{ibest} derived from Equation (14) shows that each particle remembers its best-possible prior location at each repetition. In Equation (15), vector G_{ibest} represents the best possible current location for every particle in a swarm.

$$G_{ibest} = g_1, g_2 \dots \dots g_{iD} \tag{15}$$

$$V_i(t+1) = wV_i(t) + \big(L_{ibest}(t) - X_i(t)\big)c_1 r_1(t) + \big(G_{ibest}(t) - X_i(t)\big)c_2 r_2(t) \tag{16}$$

Equations (16) and (17) are used to calculate the latest changes in position and velocity for each component.

$$X_i(t+1) = V_i(t+1) + X_i(t) \tag{17}$$

In the above equations (12-17), t represents the current iteration, c_1 and c_2 are acceleration factors of the particle motion, r_1 and r_2 are initialized random integers ranging from 0 to 1, and w is a weight coefficient.

The set of particles taken in the population to be $P_i = \{P_1, P_2, \dots, P_N\}$, where N is the total number of particles. A bit frame structure or vector is used to represent each individual particle. The number of virtual machines determines the size of the bit frame or vector. When examining the mapping of virtual machines, the virtual machine with ID 1 occupies the first available position. The appropriate position in the vector is updated with a '1' if the VM picked it as the global best; otherwise, it remains unchanged. The fitness function for each particle is calculated using Equation (11), and the CH selection algorithm utilized in the proposed EEVMPSO is described in algorithm 1 for every particle in P_i, and calculates its fitness value. In the first time period, the Pbesti is the particle P_i itself. The current iteration's best Pbesti is used to assign the output Gbest, which denotes the chosen optimum VM. After every iteration, the velocity and position of the particles are updated, which means that a new population of particles is created. Again their Pbest is calculated, and accordingly, the Gbest is updated. The process continues while stopping criteria meet, and the stopping criteria is the user sets the maximum number of iterations.

Algorithm - 1. EEVMPSO Algorithm.

Input: Swarms P_i, $1 \le i \le P_N$

Output: *Gbest*

1. **Begin**
2. **While** *round* <= I_t **do**
3. **For** *p belongs to* P_i ***do***
4. Pbesti = maximum fitness value
5. **For** $j = 1: N$ **do**
6. X_j = rand(Xmax_value, Xmin_value)
7. V_j = rand(Vmax_value, Vmin_value)

8. **End For**
9. **End For**
10. No_of_VMs = 0
11. **For** $j = 1{:}\ N$ **do**
12. **If** $(Vi(j) == 1)$ **then**
13. No_of_VMs = No_of_VMs + 1
14. **End If**
15. **End For**
16. Calculate F(i) /*using *Equation (11)* */
17. L_{ibest} = F(i)
18. $G_{ibest} = \{L_{ibest}k \mid F(L_{ibest}k) = \min (F\ (L_{ibest}),1 \le i \le P_N)\}$
19. **For** $i = 1{:}\ P_N$ **do** /* Position and velocity are updated*/
20. **If** $F(i) < F(L_{ibest})$ **then**
21. $L_{ibest} = \boldsymbol{P_i}$
22. **If** $F(i) < F(G_{ibest})$ **then**
23. $G_{ibest} = L_{ibest}$
24. **End For**
25. **End While**
26. **End**

The paper proposes a plan to reduce the total energy consumption and the energy overhead associated with migration. A technique for approximating the memory and CPU utilization rates required by individual virtual machines is presented. A Particle Swarm Optimization (PSO) technique was employed to address the problem of consolidating energy-efficient virtual machines, yielding a solution. Fig. 2 depicts the overall flowchart of the operational procedure of EEVMPSO.

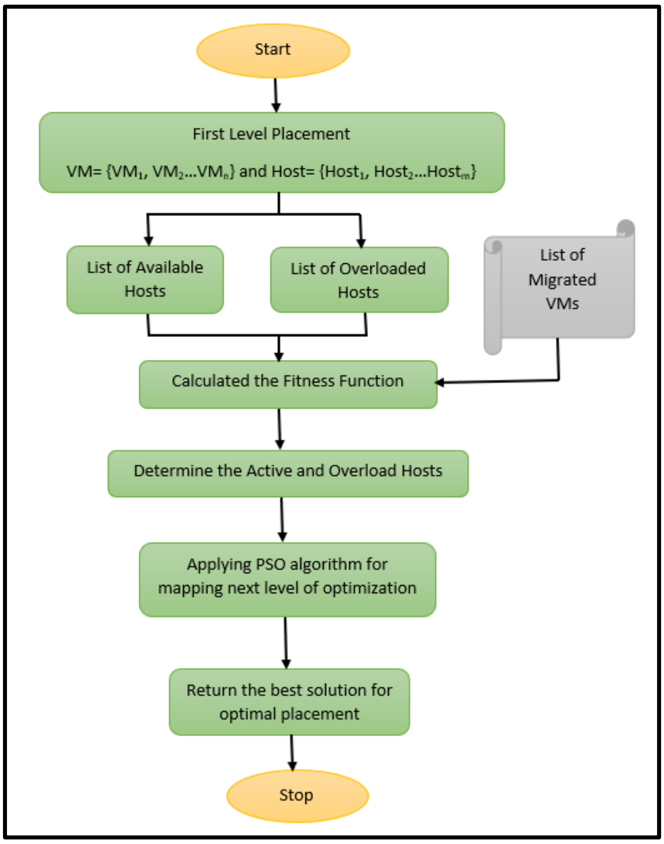

Figure 2. Working process of EEVMPSO

4. Experimental Setup

This study presents the proposed methodology's results in a simulated setting. The simulation test bench utilized in the experimentation is described in detail. Table 12 shows experiments with varying VM and PM sizes and workloads; since each experiment was repeated 20 times, we may have confidence in the findings. A trustworthy result is achieved if this information is combined with that from the standard VM placement method EEVMPSO. Experiment-specific parameter values are provided in Table 13. Analyzing the proposed technique in a real-world environment would need a significant increase in the number of physical and virtual machines (PMs and VMs) employed in our testing, which would be a time-consuming and costly endeavor. The recommended VM placement method is tested in a simulated environment as an alternate option. To design this paper, we have settled on using a simulator platform called the CloudSim toolkit. During our investigation, we used anything from fifty to two hundred actual and simulated personal computers. The proposed method is evaluated using four distinct datasets, each of which contains a different number of provisioning managers and virtual machines: Workload1, which comprises 50 VMs and 50 PMs; Workload2, which shall consist of 100 VMs and 100 PMs; Workload3, which includes 150 VMs and 150 PMs; and Workload4, which contains 200 VMs and 200 PMs. The suggested method is tested to reduce virtual machine migrations, power management shutdowns, energy consumption, and service level agreement violations in data centers.

Table 12: Experimental cases

	Workload			
	Workload 1	Workload 2	Workload 3	Workload 4
VM List	50	100	150	200
PM List	50	100	150	200

Table 13: EEVMPSO parameters

HP ProLiant ML 110G4 (S1)	1860 (MIPS), Memory 6 GB
HP ProLiant ML 110G4 (S2)	2660 (MIPS), Memory 6 GB
High CPU VM	2200 MIPS
Small VM	1000 MIPS
Micro VM	500 MIPS
Population Size	30
I_t	100
I_{min}	0.5
I_{max}	1
Initial Position	9.6
Initial Velocity	0

Table 14: Server1 and Server2 Energy utilization

Types	0%	20%	40%	60%	80%	100%
Server1	84.9	92.1	97.8	104.5	112	116.5
Server2	95	99.6	110.3	119.72	127.4	136

The Parameters used in this experiment are detailed in Table 13. It would be impossible and costly to duplicate our research in a real-world situation to assess the proposed strategy for many VMs and PMs utilized. The proposed method for VM placement is tested in a simulated environment. Information on the two types of servers and four types of virtual machines that the experiments simulate may be found in Tables 12 and 14, respectively. Using a mix of Local Regression can identify the underutilized hosts, and the VMs are relocated. Compared to PAPSO, KHA, EALBPSO, and RACC-MDT, EEVMPSO has been shown to perform better across various performance metrics.

4.1 Experimental Results Discussion

The performance of the proposed EEVMPSO for energy efficient VM allocation is evaluated using various performance metrics including the energy consumption, VM migration, CPU utilization, SLA violation, energy SLA violation, and host shutdown. EEVMPSO technique is associated with four state-of-the-art protocols, including PAPSO [17], KHA [12], EALBPSO [20], and RACC-MDT [2].

4.1.1Energy Consumption Analysis:

EEVMPSO aims to consolidate the migrated VMs onto as few hosts as is practically practicable. It increases CPU utilization on busy servers while putting idle hosts to rest. Fig. 3 displays the outcomes of this investigation, which demonstrate potential methods for cutting down on energy use in the home. Energy savings of 10.86%, 18.22%, 25.8%, and 31.34% are shown when the suggested technique is compared to PAPSO, KHA, EALBPSO, and RACC-MDT, respectively. As long as it is constrained by the requirement to prevent SLA violations, the EEVMPSO approach may adequately consume power usage.

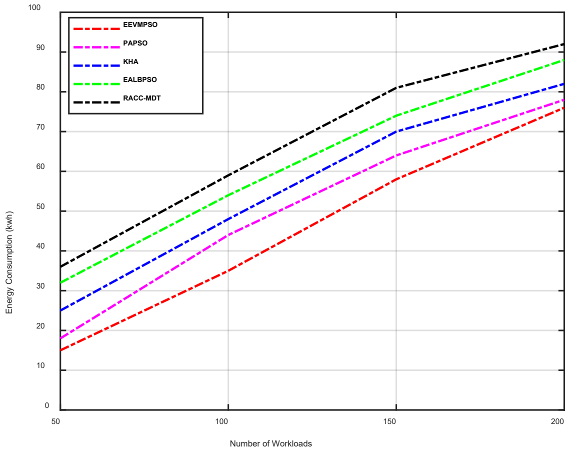

Figure 3. Comparing the consumption of energy

4.1.2 VM Migration:

When VM migration, there is a chance that the system's performance will degrade, which would violate the SLA. Performance suffers as the number of virtual machines (VMs) transferred across hosts grows. EEVMPSO implements a strategy to lessen the share of overworked hosts and the prevalence of busy servers. Therefore, decreasing the number of servers in use may increase the CPU usage of hosts, allowing for an ideally saturated set of servers. It does, however, reduce the total number of overworked servers.

Taking these two factors into account has reduced VM migrations, as seen in Fig. 2. When compared to PAPSO, KHA, EALBPSO, and RACC-MDT, the EEVMPSO approach has the potential to minimize the VM migrations by 10.09%, 10.45%, 19.16%, and 28.82%, respectively. EEVMPSO achieved this goal with fewer VM migrations, unlike VM consolidation solutions, which depend on completing numerous VM migration operations among servers to minimize the total power consumption in data centers. The attainment of the objective of diminishing the overall power consumption necessitates this measure. As a direct and immediate result of this reduction in the total number of virtual machine (VM) migration processes being carried out, the newly delivered services will be of a much higher quality.

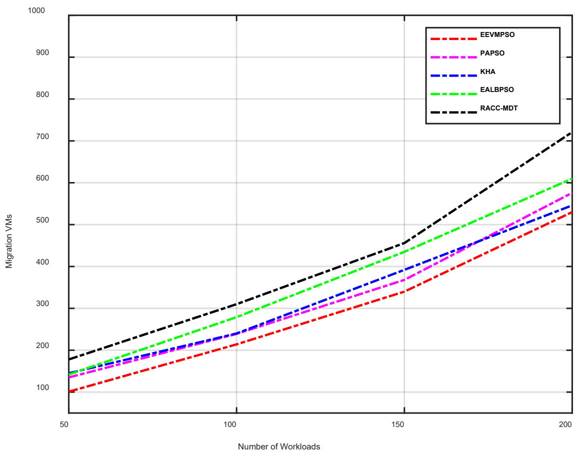

Figure 4. Comparing the number of VM migration

4.1.3 The Average Utilization of CPU

The average CPU use of the designated Server1 and Server2 classes of servers is shown in Fig. 5. The utilization rates of both individual FFD servers and the full pool of available servers are relatively low. Since there is a wide gap between the CPU utilization of Server1 and Server2, it is apparent that FFD cannot efficiently manage varied resources. It is important to note that EEVMPSO places the highest CPU demand on servers of type Server1, whereas PAPSO, KHA, EALBPSO, and RACC-MDT place lower orders on servers of type Server2.

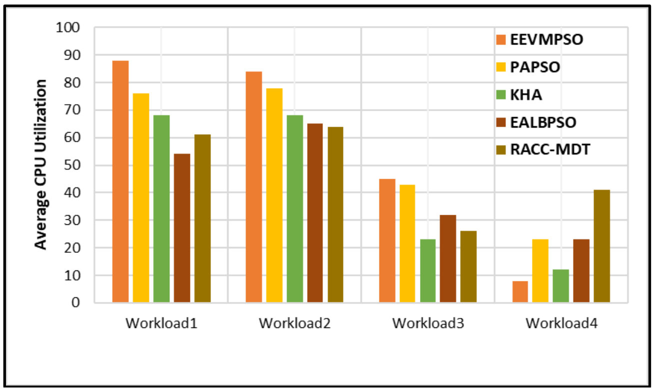

Figure 5. Comparison analysis of average CPU Utilization

EEVMPSO prefers highly configured servers over DTH-low-profile MF ones so that the VMs can communicate with one another more efficiently. Consolidation may go as deep as one's memory will allow. The fact that EEVMPSO's average memory use is thus close to 100% demonstrates how well it can consolidate high-demand virtual machines.

4.1.4 SLA Violation:

A new metric independent of processing loads must be introduced to measure the SLA provided to a virtual machine's end user. The proportion of time that servers are at 100% utilization that happens as a consequence of migrations is developed as two metrics for assessing the severity of SLA breaches. Another metric is the decline in performance that occurs during migrations. Because of this, we introduce a holistic metric called SLA Violation. Experiment results showed that the EEVMPSO does not violate SLA, unlike the PAPSO, KHA, EALBPSO, and RACC-MDT. EEVMPSO can reduce SLA violations even when the virtual machines and PMs increase. The primary objective of this research is to reduce the amount of energy used by moving as many virtual machines as possible onto as few hosts as is practically practicable. Nevertheless, by eliminating two of the most common causes of SLA violations, this endeavor may also reduce such infractions. SLA violations are reduced by 6.4%, 8.7%, 14.6%, and 20.5% on average with EEVMPSO compared to PAPSO, KHA, EALBPSO, and RACC-MDT, respectively.

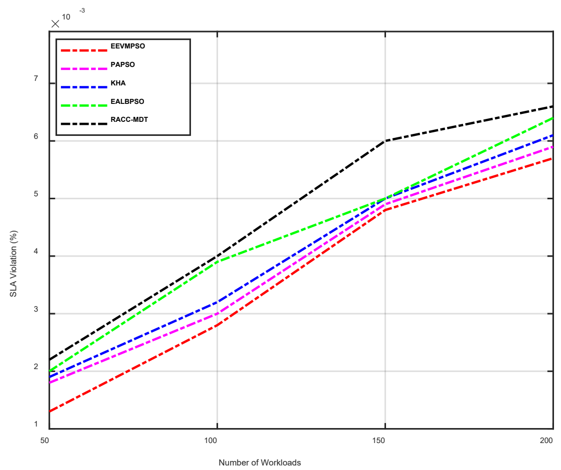

Figure 6. Comparing the SLA violations

4.1.5 Number of Migration Vs Iteration:

The average number of transitions across CPUs with different processing energy and memory is shown in Fig. 7. The proposed approach involved allocating a separate host computer to each virtual machine. The proposed solution outperformed alternatives when the actual devices were in varying states, as it took advantage of the status of physical machines in the reallocation of virtual machines and deactivated more of them. Virtual machines (VMs) were employed in all the other methods, even though they performed better when coupled with actual computers after they crossed a certain performance threshold. The experimental findings showed that the suggested approach might increase the number of migrated virtual machines by 11.19%, 19.25%, 32.86%, and 38.2% in different iterations compared to the PAPSO, KHA, EALBPSO, and RACC-MDT, respectively.

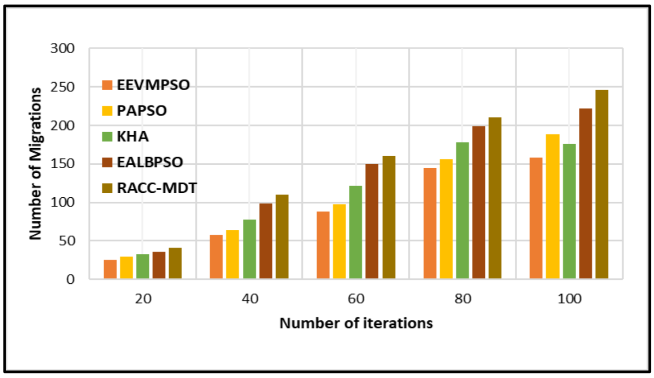

Figure 7. Comparison analysis of the number of migrations Vs different iterations

4.1.6 Energy SLA Violations:

The VM placement problem reveals that it is a crucial problem. Specifically, cloud service providers condense virtual machines (VMs) onto as few servers as possible to save operational. Conversely, consumer attention is directed toward service performance, which must remain unchanged throughout the consolidation. As a result, cloud providers are looking for ways to save costs by lowering energy use without affecting service level agreements. As a result, a new measure for judging VM location has been introduced: the trade-off, which considers both energy use and SLA breaches. Our suggested method for lowering energy usage using EEVMPSO does not infringe on any Service Level Agreements. In Fig. 8, EEVMPSO achieves better results than PAPSO, KHA, EALBPSO, and RACC-MDT when it comes to lowering energy SLA violation. When considering data center energy use and SLA breaches, an estimated 5.77%, 15.3%, 26.19%, and 30.4% of savings compared to PAPSO, KHA, EALBPSO, and RACC-MDT.

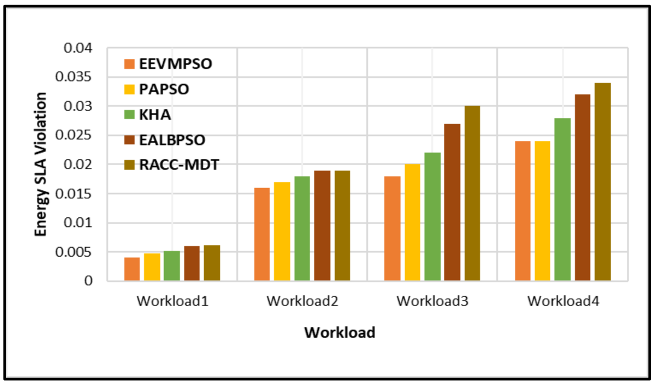

Figure 8. Comparison analysis of energy SLA violation

4.1.7 Host Shutdown:

Quality of service may also be significantly affected by the rate at which hosts are restarted. If the CPU utilization of a server falls below a predetermined threshold, the remaining virtual machines are removed. The server could host virtual machines (VMs) that have already been relocated, and if it is underutilized, it might migrate those VMs once again. As a direct result, the status could be reset to the low power mode to save energy. This circumstance harms the use of energy and user experience due to the frequent migrations of virtual machines (VMs). They are repetitively subjecting a host to a light workload and wasting resources. As seen in Fig. 9, EEVMPSO is set up to prevent overloaded and underloaded circumstances, leading to fewer host shutdowns. The proposed strategy decreases the average number of host shutdowns by 40.7%, 46.7%, 55.45%, and 6% compared to PAPSO, KHA, EALBPSO, and RACC-MDT, respectively.

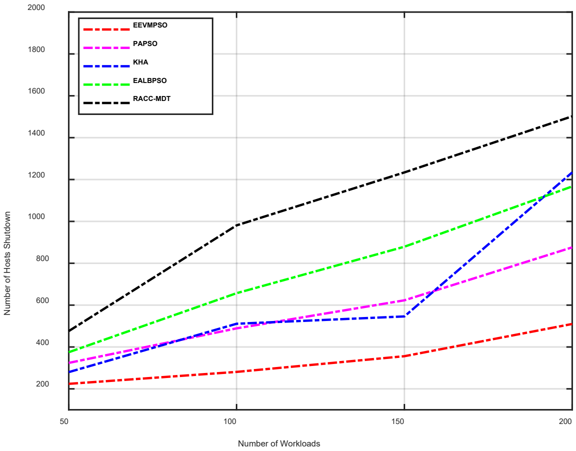

Figure 9. Comparison analysis of Hosts shutdown

5. Conclusion

In this research paper, we proposed an EEVMPSO method for dynamically assigning virtual machines to physical hosts, considering the actual consumption of cloud resources at any given time. Our research investigated the limitations of traditional VM allocation methods and demonstrated that Particle Swarm Optimization (PSO) is a promising approach for tackling the complex VM allocation problem in large-scale cloud environments. A PSO-based VM allocation strategy decreases data center energy consumption and SLA violations. Avoiding service level agreement (SLA) violations is made more accessible by moving virtual machines off servers nearing capacity. It is essential to locate hosts before transferring virtual machines. EEVMPSO is a PSO-based VM placement approach that decreases energy consumption and 2.2% violating service level agreements (SLAs). Compared to PAPSO, KHA, EALBPSO, and RACC-MDT, studies have shown that using the suggested technique may reduce energy use by an average of 10.86%, 18.22%, 25.8%, and 31.34%. The proposed method is used in CloudSim, and simulation results confirmed its usefulness concerning CPU usage, energy consumption, VM migrations, host shutdowns, and energy SLA breaches. Real-world implementation of EEVMPSO may verify its efficacy. Additionally, we recognize the constant evolution of cloud technologies, and future research should explore the integration of dynamic workload scenarios and other optimization techniques to further improve energy efficiency.

Acknowledgments.

Declaration of interests

The authors declare that they have no known competing financial interests in this paper.

Conflict of Interest

All authors have no conflict of interest to report.

Reference

[1] Garg, H., ''A hybrid PSO-GA algorithm for constrained optimization problems,'' Appl. Math. Comput., 2016, vol. 274, pp. 292–305.

[2] Ding W, Luo F, Han L, Gu C, Lu H, Fuentes J. "Adaptive virtual machine consolidation framework based on performance-to-power ratio in cloud data centers." Future Generation Computer Systems, 2020, vol.111, pp. 254-270.

[3] Patwal, R. S., N. Narang, and H. Garg, "A novel TVAC-PSO based mutation strategies algorithm for generation scheduling of pumped storage hydrothermal system incorporating solar units," energy, 2018, vol. 142, pp. 822–837.

[4] Braiki, K., Youssef, H.: Multi-objective virtual machine placement algorithm based on particle swarm optimization. In 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC), IEEE, 2018, pp. 279–284.

[5] M. Xu, W. Tian, and R. Buyya, ''A survey on load balancing algorithms for virtual machines placement in cloud computing,'' Concurrency Comput., Pract. Exper., 2017, vol. 29(12), e4123.

[6] Sun, G., Liao, D., Anand, V., Zhao, D., Yu, H.: A new technique for efficient live migration of multiple virtual machines. Future Gener. Comput. Syst. 2016, vol. 55, pp. 74–86.

[7] Yan, J., Zhang, H., Xu, H., Zhang, Z.: Discrete PSO-based workload optimization in virtual machine placement. Pers. Ubiquit. Comput., 2018, vol. 22(3), pp. 589–596.

[8] Addya, S.K., Turuk, A.K., Sahoo, B., Sarkar, M., Biswash, S.K.: Simulated annealing based VM placement strategy to maximize the profit for Cloud Service Providers. Eng. Sci. Technol. Int. J., 2017, vol. 20(4), pp. 1249–1259.

[9] Gharehpasha, S., Masdari, M., Jafarian, A.: The placement of virtual machines under optimal conditions in cloud datacenter. Inform. Technol. Control, 2019, vol. 48(4), pp. 545–556.

[10] Masdari, M., Khoshnevis, A.: A survey and classification of the workload forecasting methods in cloud computing. Clust. Comput., 2020, vol. 23(4), pp. 2399–2424.

[11] Masdari, M., Zangakani, M.: Efficient task and workflow scheduling in inter-cloud environments: challenges and opportunities. J. Supercomput., 2020, vol. 76(1), pp. 499–535.

[12] Soltanshahi, Minoo, Reza Asemi, and Nazi Shafiei. "Energy-aware virtual machines allocation by krill herd algorithm in cloud data centers." Heliyon 2019, Vol. 5(7), e02066.

[13] Shabeera, T., Kumar, S.M., Salam, S.M., Krishnan, K.M.: Optimizing VM allocation and data placement for data-intensive applications in cloud using ACO metaheuristic algorithm. Eng. Sci. Technol. Int. J., 2017, vol. 20(2), pp. 616–628.
[14] Liu, X.-F., Zhan, Z.-H., Deng, J.D., Li, Y., Gu, T., Zhang, J.: An energy efficient ant colony system for virtual machine placement in cloud computing. IEEE Trans. 2016, vol. 22(1), pp.113–128.
[15] Masdari, M., Zangakani, M.: Green cloud computing using proactive virtual machine placement: challenges and issues. J. Grid Comput, 2020, vol. 18(4), pp. 727–759.
[16] Masdari, M., Gharehpasha, S., Ghobaei-Arani, M., Ghasemi, V.: Bio-inspired virtual machine placement schemes in cloud computing environment: taxonomy, review, and future research directions. Clust. Comput., 2020, vol. 23.4, pp.2533-2563.
[17] S.Y. Hsieh, C.-S. Liu, R. Buyya, and A. Y. Zomaya, ''Utilization prediction-aware virtual machine consolidation approach for energy efficient cloud data centers,'' J. Parallel Distrib. Comput., 2020, vol. 139, pp. 99–109.
[18] Shadravan, S., Naji, H., Bardsiri, V.K.: The Sailfish Optimizer: a novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems. Eng. Appl. Artif. Intell., 2019, vol. 80, pp. 20–34.
[19] Maciel, O., Cuevas, E., Navarro, M.A., Zaldıvar, D., Hinojosa, S.: Side-blotched lizard algorithm: a polymorphic population approach. Appl. Soft Comput., 2020, vol. 88, pp. 106039.
[20] Masoudi, Javad, Behnam Barzegar, and Homayun Motameni. "Energy-aware virtual machine allocation in DVFS-enabled cloud data centers." IEEE Access, 2021, vol. 10, pp. 3617-3630.
[21] Sayantani Basu, G. Kannayaram, Somula Ramasubbareddy, C. Venkatasubbaiah, Improved genetic algorithm for monitoring of virtual machines in cloud environment, in: S.C. Satapathy, et al. (Eds.), Smart Intelligent Computing and Applications, Smart Innovation, Systems and Technologies, 105, Springer Nature Singapore Pte Ltd, 2019, vol. 2, pp. 319-326.
[22] Fares Alharbi, Yu-Chu Tian, Maolin Tang, Wei-Zhe Zhang, Chen Peng, Minrui Fei, An Ant colony system for energy-efþcient dynamic virtual machine placement in data centers, Exp. Sys. Appl., 2019, vol. 120, pp. 228-238.
[23] Xinqian Zhang, Tingming Wu, Mingsong Chen, Tongquan Wei, Junlong Zhou, Shiyan Hu, Rajkumar Buyya, Energy-aware virtual machine allocation for cloud with resource reservation, J. Syst. Softw., 2019, vol. 147, pp. 147–161.
[24] Donyagard Vahed, M. Ghobaei-Arani, and A. Souri, “Multi-objective virtual machine placement mechanisms using nature-inspired metaheuristic algorithms in cloud environments: A comprehensive review,” in Proc. Int. J. Commun. Syst., 2019, vol. 32(14), e. 4068.
[25] Singh, A. K., and J. Kumar, “Secure and energy aware load balancing framework for cloud data centre networks,” Electron. Lett., 2019, vol. 55, pp. 540–541.
[26] Tseng, F.-H., X. Wang, L.-D. Chou, H.-C. Chao, and V. C. Leung, “Dynamic resource prediction and allocation for cloud data center using the multi-objective genetic algorithm,” IEEE Syst. J., 2017, vol. 12(2), pp. 1688–1699.
[27] Saxena, D., and A. K. Singh, “A proactive autoscaling and energy-efficient vm allocation framework using online multi-resource neural network for cloud data center,” Neurocomputing, 2021, vol. 426, pp. 248–264.
[28] Sharma, N. K., and G. R. M. Reddy, “Multi-objective energy efficient virtual machines allocation at the cloud data center,” IEEE Trans. Services Comput., 2016, vol. 12(1), pp. 158–171.
[29] Shun Yao, G., Y. Sheng Ding and K. Rong Hao, “Multi objective workflow scheduling in cloud system based on cooperative multi-swarm optimization algorithm,” Journal of Central South University, 2017, vol. 24(5), pp. 1050– 1062.
[30] Dashti, S. E., and A. M. Rahmani, ``Dynamic VMs placement for energy efficiency by PSO in cloud computing," J. Experim. heor. Artif. Intell., 2016, vol. 28(1), pp. 97-112.
[31] Tharwat, A., Elhoseny, M., Hassanien, A.E., Gabel, T., Kumar, A.: Intelligent Bezier curve-based path planning model using Chaotic Particle Swarm Optimization algorithm. Clust. Comput., 2019, vol. 22(2), pp. 4745–4766.
[32] Beloglazov, A., and R. Buyya, ``Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," Concurrency Comput., Pract. Exper., 2012, vol. 24(13), pp. 1397-1420.
[33] Ibrahim, Abdelhameed, et al. "PAPSO: A power-aware VM placement technique based on particle swarm optimization." IEEE Access, 2020, vol., 8 pp. 81747-81764.
[34] Magotra, Bhagyalakshmi, and Deepti Malhotra. "Resource-Efficient VM Placement in the Cloud Environment Using Improved Particle Swarm Optimization." International Journal of Applied Metaheuristic Computing, 2022, vol.13(1), pp. 1-32.
[35] Gomathi, B., et al. "Multi-Objective Optimization of Energy Aware Virtual Machine Placement in Cloud Data Center." INTELLIGENT AUTOMATION AND SOFT COMPUTING, 2022, vol. 33.3, pp. 1771-1785.
[36] Al-Moalmi, A., et al.: A whale optimization system for energy efficient container placement in data centers. Expert Syst., 2021, vol. 164, pp. 113719.
[37] Liu, X.-F., et al., An energy efficient ant colony system for virtual machine placement in cloud computing. IEEE Transactions on Evolutionary Computation, 2016, vol. 22(1): pp. 113-128.
[38] Li, Z., et al.: Energy-aware and multi-resource overload probability constraint-based virtual machine dynamic

consolidation method. Future Gener. Comput. Syst., 2018, vol. 80, pp. 139–156.
[39] Kim, M., Hong, J., Kim, W.: An efficient representation using harmony search for solving the virtual machine consolidation. Sustainability,2019, vol.11(21), pp. 6030.
[40] Riahi, M., Krichen, S.: A multi-objective decision support framework for virtual machine placement in cloud data centers: a real case study. J. Supercomput, 2018, vol. 74(7), pp. 2984–3015.
[41] Gharehpasha, Sasan, Mohammad Masdari, and Ahmad Jafarian. "Power efficient virtual machine placement in cloud data centers with a discrete and chaotic hybrid optimization algorithm." Cluster Computing, 2021, vol. 24, pp. 1293-1315.
[42] Yavari, M., Rahbar, A.G., Fathi, M.H.: Temperature and energy aware consolidation algorithms in cloud computing. J. Cloud Comput., 2019, vol. 8(1), pp. 1-16.
[43] Wang C, Sun B, Du KJ, Li JY, Zhan ZH, Jeon SW, Wang H, Zhang J. ''A novel evolutionary algorithm with column and sub-block local search for sudoku puzzles". IEEE Transactions on Games., 2023.
[44] Yang JQ, Yang QT, Du KJ, Chen CH, Wang H, Jeon SW, Zhang J, Zhan ZH. "Bi-Directional Feature Fixation-based Particle Swarm Optimization for Large-Scale Feature Selection". IEEE Transactions on Big Data. 2022.
[45] Ge YF, Zhan ZH, Cao J, Wang H, Zhang Y, Lai KK, Zhang J. DSGA: "a distributed segment-based genetic algorithm for multi-objective outsourced database partitioning". Information Sciences. 2022, Vol. 612, pp. 864-86.
[46] Li JY, Du KJ, Zhan ZH, Wang H, Zhang J. "Distributed differential evolution with adaptive resource allocation". IEEE transactions on cybernetics. 2022.
[47] Ge YF, Orlowska M, Cao J, Wang H, Zhang Y. "MDDE multitasking distributed differential evolution for privacy-preserving database fragmentation". The VLDB Journal. 2022, Vol.31(5), pp. 957-75.