electronics

MDPI

Article

Cost-Effective and Energy-Aware Resource Allocation in Cloud Data Centers

Abadhan Saumya Sabyasachi * and Jogesh K. Muppala

Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong 999077, China
* Correspondence: abadhan@connect.ust.hk

Abstract: Cloud computing supports the fast expansion of data and computer centers; therefore, energy and load balancing are vital concerns. The growing popularity of cloud computing has raised power usage and network costs. Frequent calls for computational resources may cause system instability; further, load balancing in the host requires migrating virtual machines (VM) from overloaded to underloaded hosts, which affects energy usage. The proposed cost-efficient whale optimization algorithm for virtual machine (CEWOAVM) technique helps to more effectively place migrating virtual machines. CEWOAVM optimizes system resources such as CPU, storage, and memory. This study proposes energy-aware virtual machine migration with the use of the WOA algorithm for dynamic, cost-effective cloud data centers in order to solve this problem. The experimental results showed that the proposed algorithm saved 18.6%, 27.08%, and 36.3% energy when compared with the PSOCM, RAPSO-VMP, and DTH-MF algorithms, respectively. It also showed 12.68%, 18.7%, and 27.9% improvements for the number of virtual machine migrations and 14.4%, 17.8%, and 23.8% reduction in SLA violation, respectively.

Keywords: virtual machine; SLA; WOA; migration; cloud computing

Citation: Sabyasachi, A.S.; Muppala, J.K. Cost-Effective and Energy-Aware Resource Allocation in Cloud Data Centers. *Electronics* **2022**, *11*, 3639. https://doi.org/10.3390/electronics11213639

Academic Editor: Fernando De la Prieta

Received: 4 October 2022
Accepted: 3 November 2022
Published: 7 November 2022

1. Introduction

New computing paradigms, such as cloud computing, have emerged due to changes in the computer landscape over the last several years. Cloud computing systems have become a popular computing paradigm as they can host huge computer systems, provide various services for users to virtually access a host of resources through the internet, as well as have users only pay for the resources they use [1,2]. Virtualization technology is one of the most important features of modern data centers; further, it is particularly useful for infrastructure-based services that are housed on the cloud. Power usage in environmentally friendly data centers has skyrocketed as customers' computing demands have, similarly, skyrocketed. With the explosion in both the number and size of cloud data centers has come a corresponding rise in the problem of energy consumption, which in turn drives up the price of energy for service providers and is a major contributor to the release of harmful greenhouse gases [3,4].

As a result, several strategies for reducing energy use have been proposed in published works. Researchers have, lately, shown an interest in this issue with the hope that virtual machine (VM) migration may reduce the need for requiring many physical computers. In addition, moving virtual machines may significantly impact saving power [5]. Keeping up with the service level agreement (SLA) is a huge challenge when providing support for cloud customers, which is why distributed load balancing on physical computers is crucial. A load balancing method is employed in the cloud in order to distribute work equitably amongst available computer resources [6–8]. It is crucial in the cloud environment as it maximizes resource utilization, speeds up responses, and shortens the time it takes to do tasks. According to studies, an uneven distribution of resources may negatively impact

the performance of a cloud data center. Thus, load balancing is crucial in order to ensure the continued success of the cloud. On the other hand, in a cloud setting, the data center must regularly host the cloud service. As a result, cloud data centers use a great amount of power, adding to their operational expenses and, thus, leaving a carbon impact on the environment [9–11].

Metaheuristic approaches have been shown to be more effective in resolving the VM allocation problem (i.e., the NP-hard issue). In particular, the whale optimization algorithm (WOA) and other metaheuristics, such as particle swarm optimization (PSO) and grey wolf optimization (GWO, GA, etc.) methods have been utilized in order to optimize resource usage and decrease power consumption. Ali et al. [12] developed an improved chaotic binary GWO method in order to improve the VM allocation and to optimize resource use, balance multidimensional resources, and reduce communication traffic [13]. In order to identify an ideal assignment issue homogeneously and to simplify the VM with the often heterogeneous servers in cloud data centers, Sasan et al. [14] suggested using a chaotic hybrid optimization algorithm in order to anticipate and decrease energy usage in cloud computing [15,16].

In this work, we propose CEWOAVM (Cost-Effective Whale Optimization Algorithm), a VM placement method that is based on the whale optimization algorithm. The WOA method has a high convergence rate and is straightforward due to its resilience toward control parameters. Furthermore, in order to maximize certain aspects, such as energy efficiency, power consumption, and total resource usage, the suggested strategy considers a wide range of resources to utilize beyond using only the central processing unit (CPU) [17–21]. The system architecture of VM allocation is represented in Figure 1.

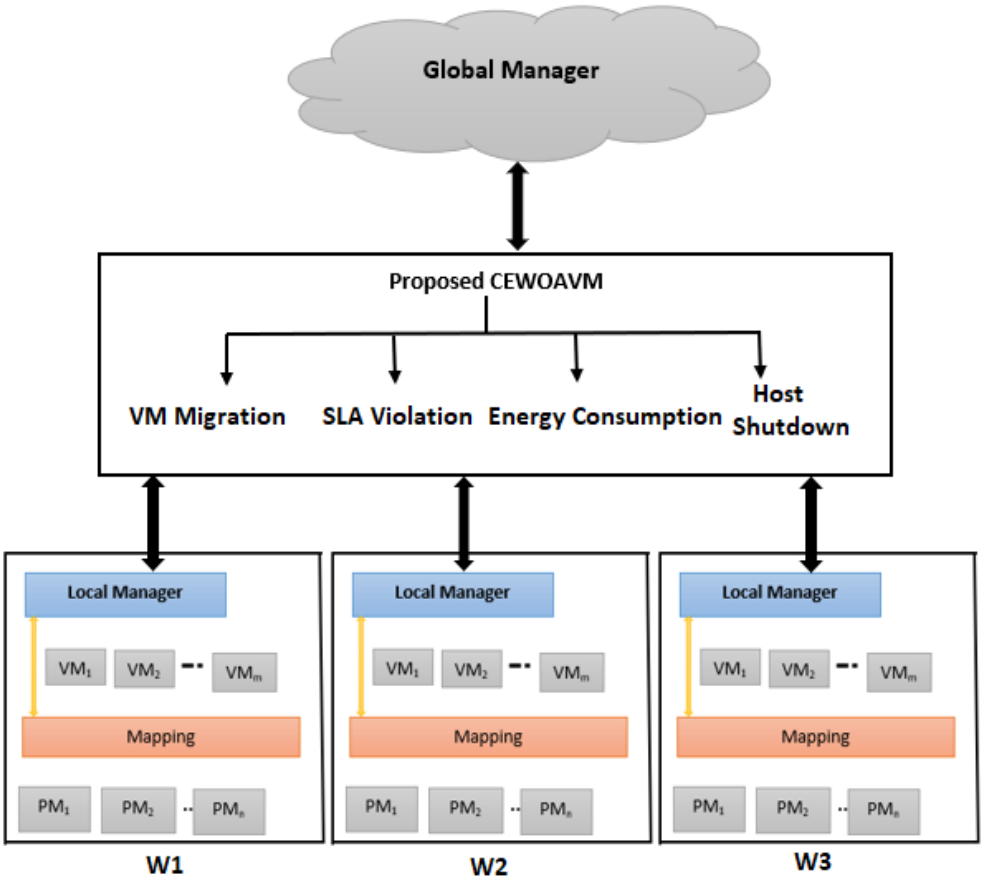

Figure 1. System architecture of VM allocations.

The significant contributions of this paper are as follows:

1. Virtual machine (VM) placement is modeled in order to maximize energy efficiency as an optimization issue.
2. The CEWOAVM technique reduces data center power usage via a cost-efficient VM allocation without breaching SLAs.
3. CEWOAVM reduces energy consumption, VM migrations, and host shutdowns, according to experimental results.
4. The fitness function is adjusted as time-varying in order to prevent the system from settling into a local optimum.

The rest of the paper is structured as follows. In Section 2, we will go over the relevant research; Section 3 will go over the suggested technique in detail; Sections 4 and 5 will show the simulation results and the conclusions.

2. Related Work

This section will provide an overview of the VM consolidation algorithms that were chosen as well as outline the essential characteristics of each. An approach to virtual machine (VM) consolidation, which is based on the WOA and hybridized with a new bandwidth allocation policy, was proposed by Abdel-baset et al. [22]. The team have been concentrating on solving the VM consolidation issue, which has been phrased as a variable-sized bin packing problem, considering the available frequency. They have executed their idea utilizing the CloudSim toolkit with 25 distinct datasets and various bandwidths implemented at random in order to verify its efficacy. The acquired findings verify that, when compared to alternative metaheuristic techniques, the mechanism reduces the number of PMs [23]. However, their approach has solely focused on improving bandwidth while ignoring other critical resources, such as memory and processing power. Furthermore, the issue of optimizing power use has not been considered.

Al-Moalmi et al. [24] proposed that the overhead caused by the dynamic placement of VMs may be caused from the time that is spent migrating. As a result, they have thought of using a static approach in order to distribute virtual machines. Their primary objective is to unify the container-to-virtual-machine placement and the virtual-machine placement-to-physical-machine placement into a single optimization challenge. The time needed to resolve the positioning challenges may be drastically cut down by using this method. To cut down on the time and energy spent on VM creation, WOA has been used to optimize this issue. However, they must double check crucial metrics, such as migration time and SLA breach [25].

A new multi-objective strategy, based on double thresholds and the ACO algorithm, was presented by Xiao et al. [26]. It uses two levels of CPU usage as cut-offs in order to determine whether or not the PM is overloaded. When a host becomes overburdened or underutilized, virtual machine consolidation is initiated. In consolidating VMs, the ACO algorithm uses several selection strategies that consider the loads of the PMs to choose VMs from both the overloaded as well as the destination PMs. In addition to minimizing migrations, performance degradation, service level agreement (SLA) violations, and overall energy consumption, the suggested approach effectively uses available computing and storage resources. However, the difficulty of calculating continues to be a significant issue [27–29].

Fatima et al. [30] have proposed a new hybrid algorithm, LMOGWO. The suggested method took design cues from grey wolves, modeling itself after the animals' hunting and pack-leading techniques. It also came with a storage archive for secondary options. The top three answers are the alpha wolves who rally the pack to pounce on their victim. Alpha, beta, and delta wolves represent the pack's top three leaders, while omega wolves represent the remaining members who have found a solution. The suggested method determines the wolf step size based on the levy flight. Finally, the suggested method is put through its paces using nine industry-standard benchmark functions.

Dahsti et al. [31] developed a solution to address the requirements provided by both service providers and users of these technologies. As a result of the research, a one-of-a-kind PaaS service for organizing client errands was developed. In the cloud, excessive energy usage and an energy performance tradeoff may occur if the specifications of the physical machine and the user's expectations are incompatible, resulting in lower provider profitability. Using the PSO, energy efficiency is increased without compromising service quality. The final aim of these strategies was to reallocate the relocated virtual machines inside the whole host. The results of the CloudSim simulations showed that the circumstances inside the simulation were very comparable to those seen in the real world [32].

Therefore, we suggest a VM placement method based on the whale optimization algorithm in order to lower data center power usage without SLA violation. The suggested method concentrates on server CPU consumption while looking for the near ideal location for VMs. The whale optimization algorithm (WOA) method, whose conventional form is appropriate for continuous problems, is used in a discrete form in this system. In particular, a suitable CEWOAVM method is employed in order to address the VM placement issue.

3. Problem Formulation

Virtual machines (VMs) may move from one host computer to another in the cloud throughout a process. A VM has the option of moving to one of numerous host computers. Simultaneously, VM migration may affect how much power a system uses. As a result, it is essential to position correctly. Good VM organization thus organizes VMs on host computers and powers down those not used in order to maximize efficiency and reduce energy usage. In this scenario, n number of VMs are expected to run on m physical servers. The challenge is developing a paradigm for the relocation of VMs on host machines. This architecture should also facilitate the transfer of a more significant number of VMs while simultaneously lowering energy usage.

The second kind of virtual machine placement happens when virtual machines are moved as part of a consolidation effort. Let List of VMs = {VM_1, VM_1 … … VM_n} and List of Hosts = {PM_1, PM_2 … … PM_m}.

Next, the aim is to influence the direction of each PM to host several VMs and offer resources in order to accomplish this. Further, hypervisors are to maintain VMs for each PM. The VM migration should consolidate VMs into the fewest active hosts, without breaking SLA, for the purposes of power management. Local managers should continually check server resource use. A global manager at the primary node should communicate with local managers in order to monitor resource use. As demonstrated in Figure 1, we can see what may determine VM migrations to reduce data center power usage. Each server's management would need to frequently check VM resource use. Local managers would choose overcrowded and underloaded servers. Further, the local managers choose the best VMs to move from overcrowded hosts. Local managers report resource use to the global manager. The global manager should optimize migrating the VM placement using power-aware VM placement based on the whale optimization algorithm. The global manager then should instruct hypervisors to migrate VMs and shut down idle servers.

The objective is to place the migrated VMs into the respective physical machines according to Equations (1)–(3).

$$\sum_{i=0}^{k-1} CPU_i + CPU_{mig} < CPU_j \quad (1)$$

$$\sum_{i=0}^{k-1} RAM_i + RAM_{mig} < RAM_j \quad (2)$$

$$\sum_{i=0}^{k-1} BW_i + BW_{mig} < BW_j \quad (3)$$

where CPU, RAM, and BW are considered necessary resources for the virtual machine (VM) to be migrated.

In addition, a VM cannot be hosted by more than one PM at once. Hence, a VM_i can only be mapped to one server. Therefore, the suggested method represents mapping moved VMs as in Equation (4).

$$V_{map} = \begin{cases} 1, & \text{VM and PM belong to the List of VM and Host} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

3.1. Energy Consumption Model

The energy needed to run a system is proportional to how often the CPU and RAM are being used. The CPU mostly drives the energy needs of these devices. Therefore, the percentage of time a computer's CPU is actively used determines how much of the machine's resources are being used. A VM energy consumption model may be calculated using Equation (5), which considers the processor's utilization concerning the machine's current workload.

$$CPU_{U_i} = \sum_{j=1}^{n} CPU_{U_{i,j}} \quad (5)$$

$$Host_{P_i} = f \times P_{max} + (1 - f) \times P_{max} \times CPU_{U_i} \quad (6)$$

$$E = \int_{t-1}^{t} Host_{P_i}\left(CPU_{U_i}(t)\right)dt \quad (7)$$

where P_{max} is the power spent at full utilization of the physical machine, f is the power consumed by the machine while it is idle, and CPU_{U_i} is the processor utilization rate (CPU_{U_i} [0, 1]) that is calculated from the workload caused by the execution of the cloud service on the physical machine ($Host_{P_i}$), which is represented by Equation (6).

In Equation (7), E is the energy used by a machine throughout the interval [t − 1, t], and $CPU_{U_i}(t)$ is the power drawn at time t, as determined.

3.2. Fitness Function

A fitness function assigns a value to a solution depending on the parameters effective in the solution's quality, allowing one to judge whether or not the solution can deliver a timely response. When applied to all solutions generated by the suggested algorithm, this function determines the value of each solution. The solution with the best value, either the maximum or the lowest depending on the parameter placement strategy, is the most practical. Thus, Equation (8) provides the fitness function that may be used in order to evaluate the quality of each response.

$$F = \frac{1}{\sum_{1}^{n} \int_{t-1}^{t} Host_{P_i}\left(CPU_{U_i}(t)\right)dt} \quad (8)$$

3.3. Proposed Method

The cloud data center will have homogenous and heterogeneous VMs. VMs require hardware from real machines in order to process services, and the cloud data centers host virtual computers on physical hardware. Hardware resources are required as cloud services demand rises. Hardware resources raise the expenses for these centers. Thus, deploying VMs in the cloud in order to optimize resource use will save expenses and allow real computers to host more VMs. Thus, effective VM placement maximizes physical machine processing power, preventing the loss of hardware resources in the cloud data center. The aim is to minimize physical machine resource utilization for a high-performance cloud data center. Cloud data center VM utilization decreases migrations and bandwidth use. Details of the CEWOAVM algorithm suggested in this research to schedule dependent activities in order to reduce energy usage and improve load balancing, are presented in this section. The reduced power consumption, CPU utilization, VM migration, and SLA violation in cloud data centers are due to the CEWOAVM algorithm.

WOA algorithms operate in a continuous cloud environment. Cloud data centers map virtual computers to actual equipment separately. The suggested technique for VM placement on physical machines in cloud data centers uses new operators in VMs in order to discretely address the deployment issue.

Whale Optimization Algorithm (WOA)

We would propose to use a WOA-based mechanism for assigning resources to physical hosts. In addition to the algorithm's time complexity, load congestion reduces data center energy usage. Furthermore, WOA is ideal for VM allocation to physical hosts (which, as a function, is an NP-hard issue). Moreover, WOA is the finest metaheuristic for tackling these situations. This method has accelerated and delivered a more efficient solution than prior solutions.

Each whale conducts the exploration by randomly updating its location in the search space or using the optimal search agent, which is determined by vector $\vec{A}$. When $\vec{A} > 1$, the whales move randomly, but when $\vec{A} < 1$ they prefer to undertake a local search. We may represent the discovery stage using arithmetic as shown in Equations (9) and (10).

$$\vec{D} = \left| \vec{C} \times \vec{W}_{rand} - \vec{X} \right| \tag{9}$$

$$\vec{X}_{(t+1)} = \vec{W}_{rand} - \vec{A} \times \vec{D} \tag{10}$$

where $\vec{W}_{rand}$ is a randomly picked whale.

The location of the prey is employed as a mathematical representation of the best response available at the simulation time, which is the exploitation phase. Each second that passes pushes the relative locations of the whales closer to the center of the circle, representing the approaching prey. The predator's behavior may resemble a spiral or an encirclement of the victim. The encircling is mathematically defined by the following Equation (11).

$$\vec{P}_{(i+1)} = \vec{X}_b(i) - \vec{A} \times \vec{D} \tag{11}$$

where at each iteration t, the agent's location is represented by the vector $\vec{P}_{(i)}$, and the vector $\vec{X}_b$ indicate the best agent (i). In Equation (12), $\vec{A}$ stands for the coefficient vector, and D stands for the distance from the best agent, as illustrated in Equation (13).

$$A = 2 \times \vec{a} \times r \; - \vec{a} \tag{12}$$

$$\vec{D} = \left| \vec{C} \times \vec{X}_b(t) - \vec{X}(t) \right| \tag{13}$$

$$\vec{C} = 2 \times r \tag{14}$$

where r ranges from [0, 1] is a random integer, $\vec{a}$ is a decreasing vector from 2 to 0, and $\vec{C}$ is an adjustment factor by which search agents capture the local regions.

Algorithm 1 presents the pseudocode of the proposed CEWOAVM algorithm.

Algorithm 1 CEWOAVM Algorithm

Input: List of Hosts = {PM_1, PM_2 … … PM_m} and Migrated VM
Output: Best_VM_{mig}
1:**while** (t < Itr_{max}) do
2: **for** each P_i do

if (q < 0.5) then
  if | A | < 1 then
    Update the Equation (12)
  else if | A | ≥ 1 then
    Initialize (X_{rand})
    Update the Equation (13)
  end if
else if (q ≥ 0.5) then
  Update the Equation (14)
end if
end for
for t ∈ max
  for VM ∈ List of VM_{mig}
    Evaluate $F = \frac{1}{\sum_{1}^{n} \int_{t-1}^{t} Host_{P_i}(CPU_{U_i}(t))dt}$
  end for
end for
Count t = t + 1
end while
Return $Best\_VM_{mig}$

CEWOAVM cycles through various stages to get a VM position closer to the optimal route. First, each particle's velocity is updated with each iteration, allowing for a recalculation of its location. Next, the whale's current location is compared to both the optimal whale position and the optimal whale position in order to determine the appropriate speed adjustment. Then, in order to obtain an index in the list of possible hosts, we round the new location results and new speed calculations to the closest integer. Finally, the fitness function for each whale is recalculated based on the updated locations—the results of these computations aid particles in their quest to discover optimal solutions. Therefore, the swarm may use this to arrive at the best possible result. The whole working process is shown in Figure 2.

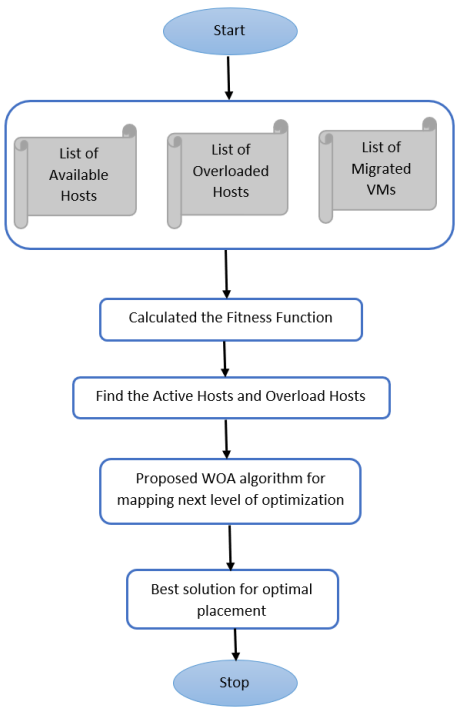

Figure 2. Workflow diagram of the CEWOAVM method.

4. Experimental Setup

The simulation test bench and analysis of outcomes from running the proposed method in a simulated environment are described in this section. In the experiments shown in Table 1, VM and PM sizes are varied, and random workloads are applied to each. Each experiment is repeated ten times in order to guarantee the accuracy of the findings. An accurate result may be given by comparing these findings to the widely used VM placement method PABFD. Table 2 lists the experimentally used parameter values. Due to the significant number of PMs and VMs used in our tests, scaling them up to evaluate the suggested approach in a real-world setting would be impractical and expensive. As an alternative, the suggested VM placement method is tested in a simulated setting. We specifically chose the simulator platform CloudSim toolkit for the experiment. Further, 50–150 PMs and 50–200 VMs were employed for this study. Workload W1, W2, and W3 data centers contain 50 VMs and 50 PMs; 100 VMs and 100 PMs; and 150 VMs and 150 PMs, respectively; further, they are utilized in order to assess the suggested method. The trials are carried out using the suggested method in order to reduce VM migrations and PM shutdowns, as well as energy consumption and SLA violations in data centers.

Table 1. Experimental cases.

	Workload		
	W1	**W2**	**W3**
Number of VM	50	100	150
Number of PM	50	100	150

Table 2. Proposed CEWOAVM parameters.

HP ProLiant ML 110G4 (S1)	1860 (IMPS), Memory 6 GB
HP ProLiant ML 110G4 (S2)	2660 (IMPS), Memory 6 GB
High CPU VM	2200 IMPS
Small VM	1000 IMPS
Micro VM	500 IMPS
N_p	30
I_t	100
IW_{min}	0.5
IW_{max}	1
C_1	2.0
C_2	2.0

Experiment-specific parameter values are listed in Table 1. Due to the enormous number of PMs and VMs used in our studies, it would be impractical and expensive to replicate them in a real-world setting in order to evaluate the suggested method. Instead, the suggested method for VM placement is put through its paces in a simulated setting. Experiments imitate two different kinds of servers and four different virtual machines, the details of which are listed in Tables 2 and 3, respectively. To determine which hosts are overburdened and which virtual machines should be moved, a combination of Local Regression is used. As measured by various performance indicators, CEWOAVM outperformed PSOCM, RAPSO-VMP, and DTH-MF.

Table 3. Energy utilization at S1 and S2.

Type of Server	**0%**	**20%**	**40%**	**60%**	**80%**	**100%**
S1	85.8	93.1	98.5	105.6	113	116.5
S2	94	100.8	111.2	120.61	128.5	136

4.1. Simulation Results and Analysis

The CEWOAVM method is compared to existing state-of-the-art methods, including PSOCM [28], RAPSO-VMP [29], and DTH-MF [30].

4.1.1. Energy Consumption Analysis

CEWOAVM's objective is to reduce the number of hosts; therefore, it merges the migrated virtual machines (VMs) into as few of them as possible. Thus, it boosts CPU use in active servers while allowing other hosts to go to sleep. The results of this analysis are shown in Figure 3, which shows how overall energy usage may be reduced. Compared to PSOCM, RAPSO-VMP, and DTH-MF, the suggested method is shown to lower power consumption by an average of 18.6%, 27.08%, and 36.3%. The suggested method may reduce power consumption by a sufficient amount, provided that it is bound by the need to avoid SLA violations.

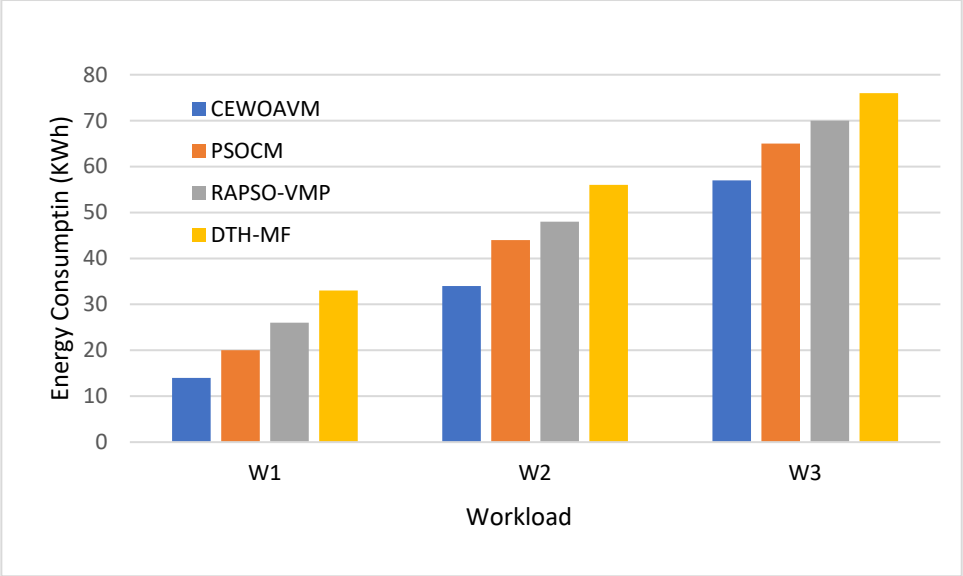

Figure 3. Comparison analysis of energy consumption.

4.1.2. VM Migration

There is a risk that live VM migrations may cause system performance degradation and, therefore, breach the SLA. In other words, performance degrades as the number of migrated VMs across hosts increases. CEWOAVM uses a method to reduce the number of busy servers and the percentage of overworked hosts. Consequently, increasing the CPU usage of hosts by decreasing the number of active servers results in an optimally small number of underloaded servers. Moreover, it decreases the number of stressed servers, as can be seen in Figure 4; in addition, this attention to the two causes of migration results in fewer VM migrations. Compared to PSOCM, RAPSO-VMP, and DTH-MF, the number of VM migrations, using the CEWOAVM method, may be reduced by 12.68%, 18.7%, and 27.9%, respectively. CEWOAVM may accomplish this goal with a drastically reduced number of VM migrations, in contrast to VM consolidation strategies, which focus on performing several VM migration operations across servers to lessen the overall power consumption in data centers. As a result of this reduction in the number of VM migration operations, the newly offered services will be of a much higher quality.

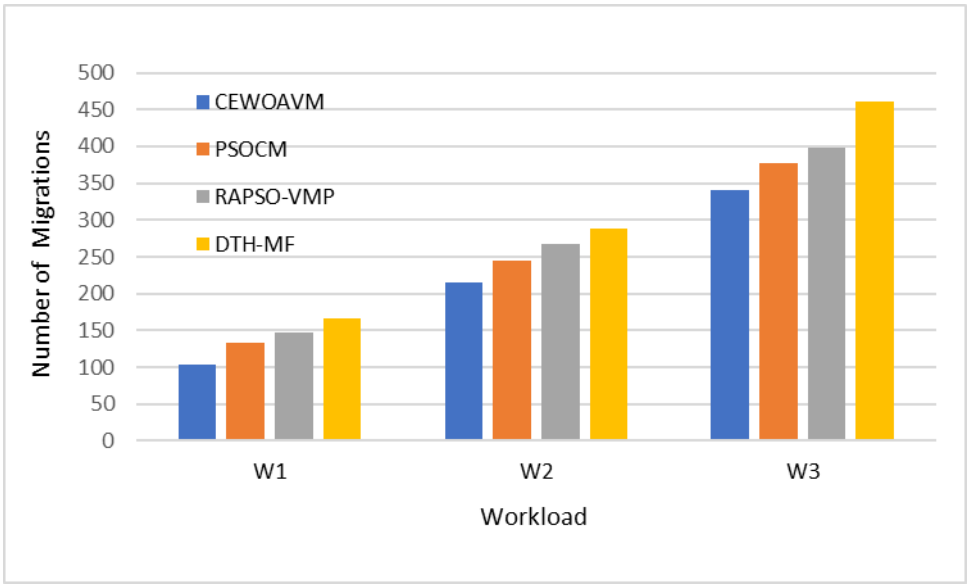

Figure 4. Comparison analysis of VM migration.

4.1.3. Average CPU Utilization

In Figure 5, we can see the typical CPU load of the assigned S1 and S2 kinds of servers. FFD's average server utilization and the average usage of all active servers are both low. FFD cannot effectively balance diverse resources as there is a large discrepancy between the CPU consumption of types S1 and S2. It is worth noting that type S1 active servers have the maximum CPU usage for CEWOAVM, whereas type S2 servers are less taxed than those for PSOCM, RAPSO-VMP, and DTH-MF. Compared to DTH-MF, which uses low-profile servers, CEWOAVM favors those with higher configurations, such that the VMs can work together more seamlessly. Moreover, the maximum degree of consolidation is limited by memory capacity. CEWOAVM has the most significant average memory consumption, close to 100%, demonstrating its ability to achieve the maximum consolidation of VMs with high resource use.

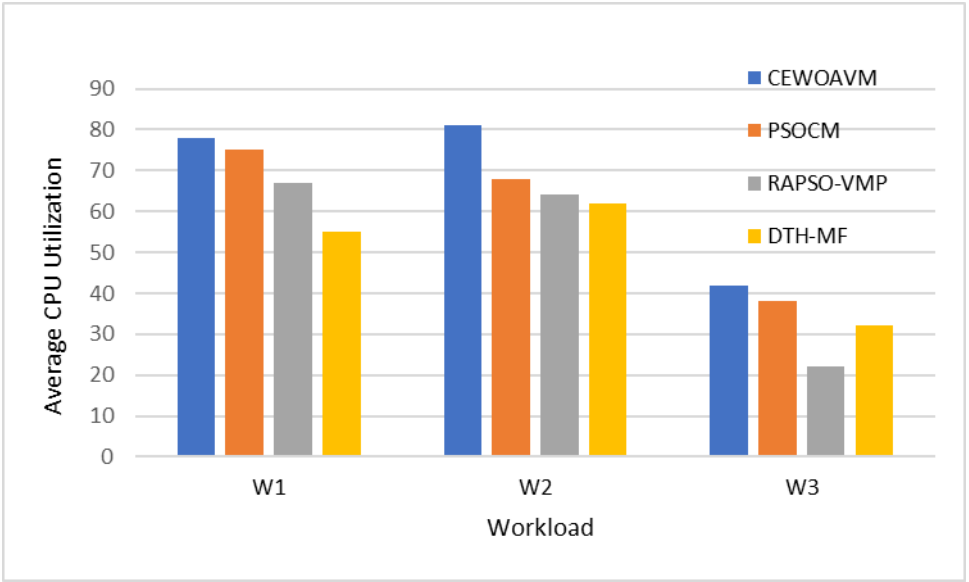

Figure 5. Comparison analysis of average CPU utilization.

4.1.4. SLA Violation

As a result, in an IaaS setting, evaluating the SLA offered to a VM's user necessitates the introduction of a statistic unrelated to workloads. Two metrics are established for the purposes of monitoring the severity of SLA violations: the first measure is when the time servers are at 100% utilization and the percentage of service degradation that occurs due to migrations. Performance decline during migrations is the other measure (PDM). As a result, we present a composite measure, SLA violation (SLAV), which is derived from the two individual metrics. In contrast to the PSOCM, RAPSO-VMP, and DTH-MF methods, experimental findings demonstrated that the CEWOAVM does not violate SLA (see Figure 6). Moreover, the CEWOAVM prevents an increase in SLA violation. However, it may minimize SLA violations as the number of VMs and PMs grows. The primary motivation for this effort is to reduce energy usage by consolidating migrating virtual machines onto as few hosts as possible. However, this work may also prevent SLA violations by avoiding their two significant causes. Compared to PSOCM, RAPSO-VMP, and DTH-MF, respectively, CEWOAVM reduces the amount of SLA violations by 14.4%, 17.8%, and 23.8%, respectively, on average.

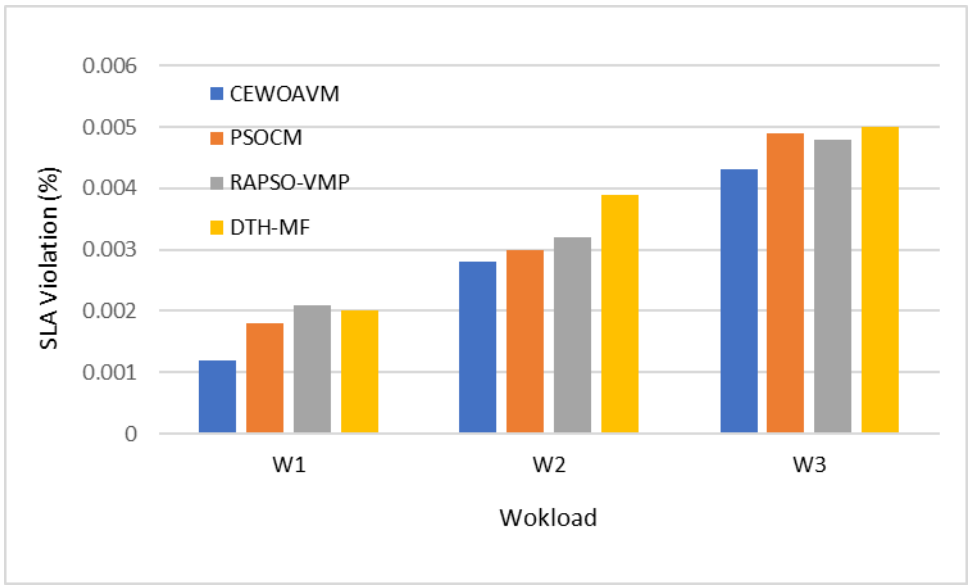

Figure 6. Comparison analysis of SLA violations.

4.1.5. The Number of Host Shutdowns

The frequency with which hosts have shut down also significantly impacts QoS. VMs must be moved off a server when CPU consumption drops below a certain threshold. In the future, the server may host virtual machines (VMs) that were previously moved, and it could then relocate those VMs again if they are underutilized. As a result, its condition may be switched back to the low power mode to save energy. This situation harms not only energy usage, but also user experience because of the frequent VM migrations. Repeatedly running a host with low loads is not cost-effective. If CEWOAVM is configured to prevent overloaded and underloaded cases, as shown in Figure 7, fewer host shutdowns will occur. Compared to PSOCM, RAPSO-VMP, and DTH-MF, the suggested method reduces the average number of host shutdowns by 45.39%, 52.94%, and 66.46%, respectively.

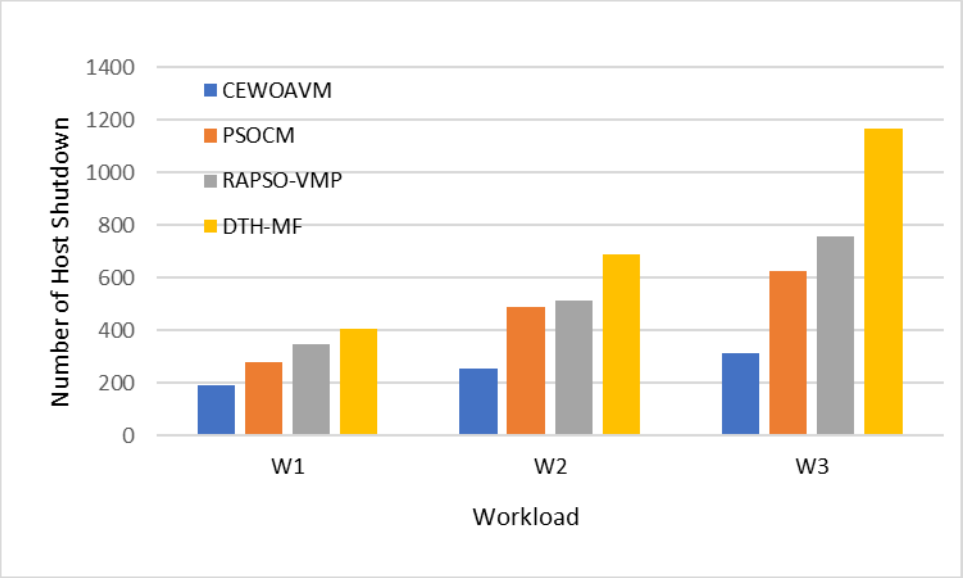

Figure 7. Comparison analysis of host shutdowns.

4.1.6. Error Analysis

Figure 8 shows the error bar chart with an interpolation line created to show how the algorithm performed differently. Figure 8 demonstrates that the CEWOAVM performed better than the RAPSO-VMP. It was noted that the PSOCM and DTH-MF algorithms either performed averagely or poorly in comparison.

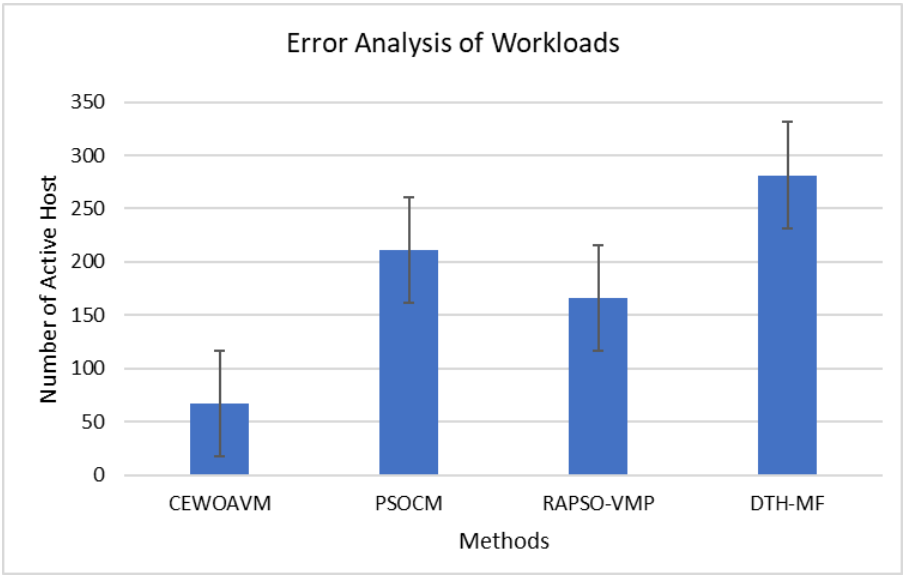

Figure 8. Error analysis bar plot.

4.2. Discussion

The methodology known as CEWOAVM has been assessed using three different test cases while being subjected to various workloads, as mentioned in the prior section. It does this by utilizing the capabilities of the WOA in order to look for the optimal VM placement option both locally and globally. Its purpose is to reduce the amount of power wasted in data centers by implementing an efficient power-aware method that can route migrating virtual machines to the servers most suited to host them. In particular, it works to maintain servers in the normal mode by minimizing the number of servers that are either overloaded or underloaded relative to their capacity. Overloaded servers harm the

quality of the services that are provided to users. On the other hand, underloaded servers cause a waste of resources and increase the data center power consumption that is needed to perform virtual machine migrations, which violates the service level agreement (SLA). The results of experiments, when compared to DTH-MF, have shown that CEWOAVM can prevent SLA violations by preventing servers from entering an overload or underload condition. Consolidating a large number of virtual machines onto a smaller number of physical servers is another way CEWOAVM may cut down on the number of power servers used. According to the findings of simulations, CEWOAVM has the potential to reduce an average of roughly 36.3% of the power usage, 27.9% of the number of VM migrations, and 66.46% of host shutdowns.

In conclusion, the effectiveness of CEWOAVM is brought into focus by the SLA violation. CEWOAVM's primary purpose is to reduce the power used while simultaneously protecting SLAs from being violated. In order to demonstrate how effective CEWOAVM is, we will proceed with an alternative scenario predicated on aggressive consolidation utilizing WOA. The experimental results showed that the proposed algorithm saved 18.6%, 27.08%, and 36.3% of energy compared with the PSOCM, RAPSO-VMP, and DTH-MF algorithms, respectively. In addition, it also showed 12.68%, 18.7%, and 27.9% improvements for the number of virtual machine migrations and 14.4%, 17.8%, and 23,8% reduction in SLA violation, respectively. The identical method is used in this circumstance, except that the fitness function does not consider the total number of overloaded hosts. It is only concerned with maximizing usage while simultaneously reducing the number of active hosts. As a result, it can achieve a more significant decrease in the power utilized when compared to CEWOAVM. However, this decrease comes at the price of the performance, which entirely breaches the service level agreement (SLA).

5. Conclusions

Cloud computing and its related services are prevalent; hence, data center building has risen. Data center power usage needs to be managed. Moreover, consolidating VMs onto the fewest possible servers reduces power usage. Further, cloud computing migrates VMs from underloaded servers to other hosts, such that their original hosts may go into sleep mode. Migrating VMs from overloaded servers helps prevent SLA violations. Moreover, finding hosts for migrating VMs is crucial. CEWOAVM, a cost-efficient VM placement strategy based on the WOA algorithm, reduces power usage without breaching SLA. WOA uses decimal encoding for ongoing issues, such as VM placement. A fitness function reduced active and overburdened servers. In addition, the suggested approach was applied in CloudSim, and simulation results proved its efficiency in terms of used energy, host shutdowns, VM migrations, and CPU utilization. CEWOAVM's efficiency may be validated via real-world deployment. Moreover, memory, bandwidth, and network parameters may also be optimized further.

Author Contributions: Conceptualization, A.S.S. and J.K.M.; methodology, A.S.S.; software, A.S.S.; validation, A.S.S.; formal analysis, A.S.S.; investigation, A.S.S.; resources, A.S.S.; data curation, A.S.S.; writing—original draft preparation, A.S.S.; writing—review and editing, A.S.S. and J.K.M.; visualization, A.S.S.; supervision, J.K.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data generated or analyzed during this study are included in this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Garg, H. A hybrid PSO-GA algorithm for constrained optimization problems. *Appl. Math. Comput.* **2016**, *274*, 292–305. [CrossRef]
2. Zhou, Z.; Chang, J.; Hu, Z.; Yu, J.; Li, F. A modified PSO algorithm for task scheduling optimization in cloud computing. *Concurr. Comput. Pract. Exp.* **2018**, *30*, e4970. [CrossRef]
3. Patwal, R.S.; Narang, N.; Garg, H. A novel TVAC-PSO based mutation strategies algorithm for generation scheduling of pumped storage hydrothermal system incorporating solar units. *Energy* **2018**, *142*, 822–837. [CrossRef]
4. Kumar, D.; Raza, Z. A PSO-based VM resource scheduling model for cloud computing. In Proceedings of the 2015 IEEE International Conference on Computational Intelligence & Communication Technology, Ghaziabad, India, 13–14 February 2015; pp. 213–219.
5. Xu, M.; Tian, W.; Buyya, R. A survey on load balancing algorithms for virtual machines placement in cloud computing. *Concurr. Comput. Pract. Exp.* **2017**, *29*, e4123. [CrossRef]
6. Mondal, S.K.; Sabyasachi, A.S.; Muppala, J.K. On Dependability, Cost and Security Trade-Off in Cloud Data Centers. In Proceedings of the 2017 IEEE 22nd Pacific Rim International Symposium on Dependable Computing (PRDC), Christchurch, New Zealand, 22–25 January 2017; pp. 11–19.
7. Maciel, O.; Cuevas, E.; Navarro, M.A.; Zaldıvar, D.; Hinojosa, S. Side-blotched lizard algorithm: A polymorphic population approach. *Appl. Soft Comput.* **2020**, *88*, 106039. [CrossRef]
8. Cuevas, E.; Fausto, F.; Gonzalez, A. The locust swarm optimization algorithm. In *New advancements in Swarm Algorithms: Operators and Applications*; Springer: Cham, Switzerland, 2020; pp. 139–159.
9. Galvez, J.; Cuevas, E.; Becerra, H.; Avalos, O. A hybrid optimization approach based on clustering and chaotic sequences. *Int. J. Mach. Learn. Cybern.* **2020**, *11*, 359–401. [CrossRef]
10. Arora, S.; Anand, P. Chaotic grasshopper optimization algorithm for global optimization. *Neural Comput. Appl.* **2019**, *31*, 4385–4405. [CrossRef]
11. 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**, *22*, 4745–4766. [CrossRef]
12. Ali, M.; Masdari, M.; Gharehchopogh, F.S.; Jafarian, A. Improved chaotic binary grey wolf optimization algorithm for workflow scheduling in green cloud computing. *Evol. Intell.* **2021**, *14*, 1997–2025.
13. Ali, M.; Masdari, M.; Gharehchopogh, F.S.; Jafarian, A. A hybrid multi-objective metaheuristic optimization algorithm for scientific workflow scheduling. *Clust. Comput.* **2021**, *24*, 1479–1503.
14. Sasan, G.; Masdari, M.; Jafarian, A. Power efficient virtual machine placement in cloud data centers with a discrete and chaotic hybrid optimization algorithm. *Clust. Comput.* **2021**, *24*, 1293–1315.
15. Sasan, G.; Masdari, M.; Jafarian, A. Virtual machine placement in cloud data centers using a hybrid multi-verse optimization algorithm. *Artif. Intell. Rev.* **2021**, *54*, 2221–2257.
16. Sasan, G.; Masdari, M.; Jafarian, A. The placement of virtual machines under optimal conditions in cloud datacenter. *Inf. Technol. Control* **2019**, *48*, 545–546.
17. Hekimoglu, B. Optimal tuning of fractional order PID controller for DC motor speed control via chaotic atom search optimization algorithm. *IEEE Access* **2019**, *7*, 38100–38114. [CrossRef]
18. Sayed, G.I.; Hassanien, A.E.; Azar, A.T. Feature selection via a novel chaotic crow search algorithm. *Neural Comput. Appl.* **2019**, *31*, 171–188. [CrossRef]
19. Qureshi, B. Profile-based power-aware workflow scheduling framework for energy-efficient data centers. *Future Gener. Comput. Syst.* **2019**, *94*, 453–467. [CrossRef]
20. Li, X.; Yu, W.; Ruiz, R.; Zhu, J. Energy-aware cloud workflow applications scheduling with geo-distributed data. *IEEE Trans. Serv. Comput.* **2020**, *15*, 891–903. [CrossRef]
21. Xu, P.; He, G.; Li, Z.; Zhang, Z. An efficient load balancing algorithm for virtual machine allocation based on ant colony optimization. *Int. J. Distrib. Sens. Netw.* **2018**, *14*, 1550147718793799. [CrossRef]
22. Abdel-Basset, M.; Abdle-Fatah, L.; Sangaiah, A.K. An improved Le'vy based whale optimization algorithm for bandwidth-efficient virtual machine placement in cloud computing environment. *Clust. Comput.* **2019**, *22*, 8319–8334. [CrossRef]
23. Yadav, R.; Zhang, W.; Kaiwartya, O.; Singh, P.R.; Elgendy, I.A.; Tian, Y. Adaptive energy-aware algorithms for minimizing energy consumption and sla violation in cloud computing. *IEEE Access* **2018**, *6*, 55923–55936. [CrossRef]
24. Al-Moalmi, A.; Luo, J.; Salah, A.; Li, K.; Yin, L. A whale optimization system for energy efficient container placement in data centers. *Expert Syst. Appl.* **2021**, *164*, 113719. [CrossRef]
25. Hsieh, S.Y.; Liu, C.-S.; Buyya, R.; Zomaya, A.Y. Utilizationprediction-aware virtual machine consolidation approach for energy efficient cloud data centers. *J. Parallel Distrib. Comput.* **2020**, *139*, 99–109. [CrossRef]
26. Xiao, H.; Hu, Z.; Li, K. Multi-objective VM consolidation based on thresholds and ant colony system in cloud computing. *IEEE Access* **2019**, *7*, 53441–53453. [CrossRef]
27. Gomathi, B.; Balaji, B.S.; Kumar, V.K.; Abouhawwash, M.; Aljahdali, S.; Masud, M.; Kuchuk, N. Multi-Objective Optimization of Energy Aware Virtual Machine Placement in Cloud Data Center. *Intell. Autom. Soft Comput.* **2022**, *33*, 1771–1785. [CrossRef]
28. Bhagyalakshmi, M.; Malhotra, D. Resource-Efficient VM Placement in the Cloud Environment Using Improved Particle Swarm Optimization. *Int. J. Appl. Metaheuristic Comput.* **2022**, *13*, 1–32.

29. Fard, Z.; Yahya, S.; Reza Ahmadi, M.; Adabi, S. A dynamic VM consolidation technique for QoS and energy consumption in cloud environment. *J. Supercomput.* **2017**, *73*, 4347–4368.
30. Fatima, A.; Javaid, N.; Anjum Butt, A.; Sultana, T.; Hussain, W.; Bilal, M.; Hashmi, M.; Akbar, M.; Ilahi, M. An enhanced multi-objective gray wolf optimization for virtual machine placement in cloud data centers. *Electronics* **2019**, *8*, 218. [CrossRef]
31. Dashti, S.E.; Rahmani, A.M. Dynamic VMs placement for energy efficiency by PSO in cloud computing. *J. Exp. Theor. Artif. Intell.* **2016**, *28*, 97–112. [CrossRef]
32. Kim, M.; Hong, J.; Kim, W. An efficient representation using harmony search for solving the virtual machine consolidation. *Sustainability* **2019**, *11*, 6030. [CrossRef]