

Received March 13, 2020, accepted April 16, 2020, date of publication April 30, 2020, date of current version May 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2990828

# PAPSO: A Power-Aware VM Placement Technique Based on Particle Swarm Optimization

ABDELHAMEED IBRAHIM<sup>1</sup>, MOSTAFA NOSHY<sup>1</sup>, HESHAM ARAFAT ALI,  
AND MAHMOUD BADAWY<sup>1</sup>

Computer Engineering and Control Systems Department, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt

Corresponding author: Mostafa Noshy (mostafanoshy@mans.edu.eg)

**ABSTRACT** With the widespread usage of cloud computing to benefit from its services, cloud service providers have invested in constructing large scale data centers. Consequently, a tremendous increase in energy consumption has arisen in conjunction with its results, including a remarkable rise in costs of operating and cooling servers. Besides, increasing energy consumption has a significant impact on the environment due to emissions of carbon dioxide. Dynamic consolidation of Virtual Machines (VMs) into the minimal number of Physical Machines (PMs) is considered as one of the magic solutions to manage power consumption. The virtual machine placement problem is a critical issue for good VM consolidation. This paper proposes a Power-Aware technique depending on Particle Swarm Optimization (PAPSO) to determine the near-optimal placement for the migrated VMs. A discrete version of Particle Swarm Optimization (PSO) is adopted based on a decimal encoding to map the migrated VMs to the best appropriate PMs. Furthermore, an effective minimization fitness function is employed to reduce power consumption without violating the Service Level Agreement (SLA). Specifically, PAPSO consolidates the migrated VMs into the minimum number of PMs with a major constraint to decrease the number of overloaded hosts as much as possible. Therefore, the number of VM migrations can be reduced drastically by taking into consideration the main sources for VM migrations; overloaded hosts and underloaded ones. PAPSO is implemented in CloudSim and the experimental results on random workloads with different sizes of VMs and PMs show that PAPSO does not violate SLA and outperforms the Power-Aware Best Fit Decreasing algorithm (PABFD). It can reduce about 8.01%, 39.65%, 66.33%, and 11.87% on average in terms of consumed energy, number of VM migrations, number of host shutdowns and the combined metric Energy SLA Violation (ESV), respectively.

**INDEX TERMS** Cloud computing, live VM migration, dynamic VM consolidation, virtual machine placement, energy consumption, service level agreement, particle swarm optimization, CloudSim.

## I. INTRODUCTION

Virtualization enables cloud computing to prevail as a pioneer trend. It paved the way to achieve the best utilization from computing and storage resources. A Physical Machine (PM) can host multiple operating systems (OSs), where they can run effectively, independently and securely. Virtual Machines (VMs) can share the available resources, with an illusion of controlling their owned hardware. A VM provides a suitable environment for the OS to run efficiently.

The associate editor coordinating the review of this manuscript and approving it for publication was M. Shamim Hossain<sup>1</sup>.

Cloud computing allows delivering services on-demand over the Internet based on the pay-as-you-go model. These services can be Infrastructure as a Service (IaaS), Platform as a Service (PaaS) or Software as a Service (SaaS) [1].

With the steady increase in cloud computing usage to benefit from the enormous services introduced by data centers, a tremendous increase in the consumed power by these data centers has appeared as a critical problem. For example, an average data center consumes an amount of energy that is equivalent to 25000 households [2]. The electricity consumption by data centers in 2010 was ranged from 1.1% to 1.5% of the overall electricity usage [3]. Besides, it is estimated

that the amount of power consumption by data centers in the United States will be 73 billion kilowatt-hours (kWh) in 2020 [4]. Moreover, there is a prediction that there will be a massive increase, with about 66%, in electricity consumption due to data centers over the interval from 2011 to 2035 [5]. This tremendous amount in the energy consumption leads to a remarkable rise in costs for operating and cooling data centers, in addition to, the bad effect on the environment due to emissions of carbon dioxide (CO<sub>2</sub>). According to [2], emissions of carbon by data centers exceeded both Netherlands and Argentina. The Environmental Protection Agency (EPA) showed that about 0.72 tons of CO<sub>2</sub> can be emitted from each 1000 kWh of the consumed power [6].

Inefficient utilization of server resources is considered as the main reason for leveraging the consumed energy in data centers. For example, a study for six months showed that the CPU utilization for about 5000 servers is almost ranged only between 10% and 50% of the maximum level of utilization [7]. Also, studies showed that idle servers can consume an amount of energy which is equivalent to 70% of the power consumed by those ones that operate with their maximum power [8]. With the enormous amount of consumed energy, industry and academia have introduced many techniques to reduce the consumed power in data centers. Dynamic Voltage and Frequency Scaling (DVFS) is a major technique that has been adopted to reduce the consumed power in data centers [9], [10]. DVFS is based on adjusting the voltage and clock frequency of CPU to decrease the power consumption. On the other hand, reallocating VMs between hosts has arisen as a pioneering approach to minimize the power consumed by data centers. It is referred to this approach as dynamic consolidation of VMs, where live VM migration can be exploited to increase the resource utilization.

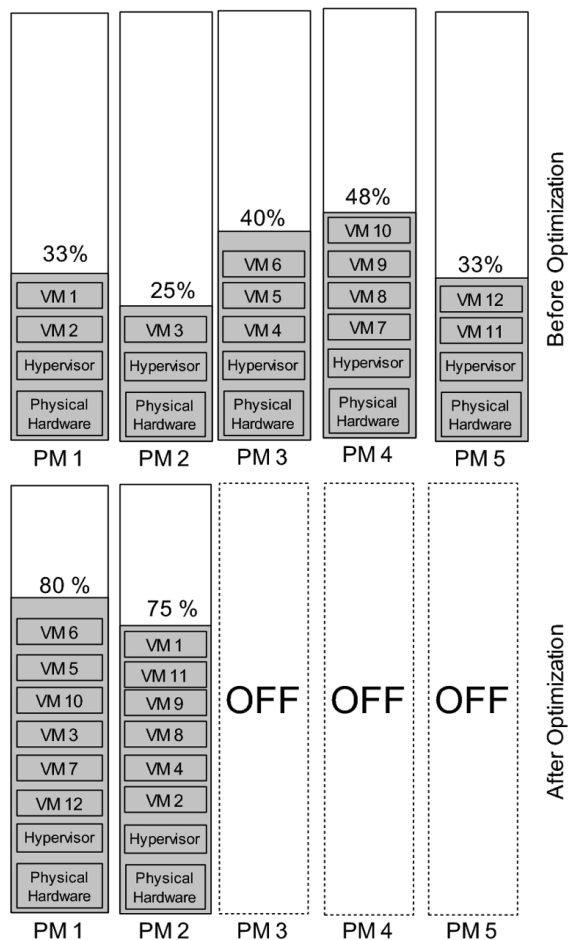
Live VM migration [11]–[13] is one of the leading technologies for managing data centers, where VMs continue to run during the migration process. In other words, it allows transferring a VM from one PM to another without disrupting the services. Thereby, the previous technology can be exploited in many situations to fulfill many benefits such as power management by consolidating VMs into a minimum number of servers. Also, it can achieve load balancing by alleviating loads from overloaded servers. Due to the importance of live VM migration as a powerful tool in managing data centers, many optimization techniques have been proposed to enhance the live migration process [14]. The total time from starting the migration until VM becomes no longer available at its original server is denoted as the total migration time, which should be reduced as much as possible to benefit from the advantages of migration. Moreover, the total transferred data should be decreased to minimize the migration time, which in turn achieves the best utilization of the network bandwidth. Downtime is also another critical time that should not be neglected. This time refers to the interval during which the service becomes unavailable due to the migration process.

For accomplishing the live migration process, VM should be migrated with all its contents including (memory pages, storage, and processor state). However, migrating the memory pages in data centers is almost the main challenge [15]. This is due to the nature of the storage data in data centers, which are almost shared between the source and the destination. Another reason is the small size of the processor state in comparison to the huge size of memory pages. The software layer which is responsible for managing VMs is the Virtual Machine Monitor (VMM) or hypervisor. Nowadays, live VM migration is supported by most current hypervisors such as Xen [16], KVM [17] and VMware [18].

Dynamic VM consolidation has become a magic solution to increase the resource utilization and reduce the energy consumption in data centers. Specifically, it depends on live VM migration to reallocate VMs from underutilized servers to other ones. It can consolidate VMs into a fewer number of hosts as depicted in Figure 1. Consequently, it allows to switch off their original hosts or change their state to a low power mode. In other words, it can save a portion of the energy consumption in data centers by minimizing the number of active hosts. Nevertheless, an aggressive VM consolidation may degrade the performance of the introduced services to clients, thereby leads to violating the Service Level Agreement (SLA). Thus, a trade-off should be taken into account to minimize the power consumption in data centers without violating the SLA. Therefore, many techniques have been proposed in the context of VM consolidation to decide overloaded hosts, underloaded hosts, VMs that need to be migrated from the overloaded hosts and finally which servers will receive the migrated VMs.

In this paper, a Power-Aware VM placement technique is proposed based on Particle Swarm Optimization algorithm (PAPS0). It aims to minimize the amount of energy consumption in data centers by consolidating VMs into the minimum number of servers, but also takes into consideration the Quality of Service (QoS) that is introduced to clients. As a meta-heuristic algorithm, Particle Swarm Optimization (PSO) is adopted in the proposed technique. It can provide the ability of both local search and global search, thereby helping to reach the near-optimal VM placement solution. PAPS0 relies on a minimization fitness function to decide the near-optimal VM placement solution. The proposed technique is implemented in CloudSim and evaluated in comparison to Power-Aware Best Fit Decreasing algorithm (PABFD) [20] with random workloads under different sizes of VMs and PMs. The main contributions of this paper are organized as follows:

- 1) A power-aware VM placement technique, PAPS0, is proposed to reduce the power consumption in data centers without violating SLA.
- 2) PAPS0 achieves the near-optimal VM placement using a discrete version of the Particle Swarm Optimization (PSO) algorithm. It adopts PSO to benefit from its ability to search for the optimal solution locally



**FIGURE 1.** An example for dynamic VM consolidation into a minimum number of servers [19].

and globally. As PSO deals with continuous problems, the proposed technique introduces an effective decimal encoding to the VM placement problem to convert PSO into a discrete version.

- 3) PAPSPO uses a feasible minimization fitness function to find the near-optimal solution. It depends on the weighted sum method to merge the number of active hosts and the number of overloaded hosts.
- 4) Experimental results demonstrate the worthiness of PAPSPO in decreasing the consumed energy, the number of VM migrations and the number of host shutdowns with about 8.01%, 39.65%, and 66.33% on average, respectively.
- 5) Energy SLA Violation (ESV), as a combined performance metric related to the trade-off between energy consumption and SLA, proves the superiority of PAPSPO over PABFD by achieving about 11.87% reduction on average.
- 6) An aggressive scenario for VM consolidation is introduced to prove the importance of PAPSPO. This aggressive scenario, Aggressive Consolidation based on PSO (ACPSO), can achieve more reduction in the consumed power, but at the expense of SLA.

The rest of the paper is organized as follows: Section II overviews a background of the proposed technique. Section III refers to the related works and provides a detailed comparison between VM placement techniques. Section IV states the VM placement problem. Section V describes the system architecture. The proposed technique, experiment results with analysis, and discussion, are presented in sections VI, VII, and VIII. Finally, conclusion and future work are explored in section IX.

## II. BACKGROUND

As the proposed work relies on the PSO algorithm, an overview of this algorithm should be provided to understand how it works. Although PSO treats with the continuous search domains, our problem search domain is discrete. Therefore, the discrete PSO has to be explored before introducing the proposed technique.

### A. PARTICLE SWARM OPTIMIZATION

PSO is a prevalent meta-heuristic population-based optimization algorithm to find the near-optimal solution in many problems. Kennedy and Eberhart [21] have developed PSO in 1995. As based on Swarm Intelligence (SI), it simulates the social behavior of living creatures like bird flocks and fish schools. They live in groups, where they can collaborate and interact with themselves or their environment. Thus, they can broaden the knowledge that helps them to achieve their sophisticated tasks.

When studying a group of animals or a flock of birds that search for its food, it will be noticed that they interact as a group. Firstly, all group members are scattered randomly to find the food. Then, they can update their directions according to the nearest one to the food. Similarly, PSO is initialized with several particles, where each one has a position and a velocity. These particles can update their velocities based on both previous knowledge and interactions with other swarm members. Therefore, positions of particles are changed in each iteration from the current location to a new one according to the new speed. This iterative process leads to the near-optimal solution that has the best fitness function. If the goal is to find the minimum value according to the proposed fitness function which evaluates each solution, it is called a minimization fitness function and vice versa. During its search for the best solution, each particle has a *pBestPosition* and a *pBestValue* represent the position of a particle that achieves the best value in the fitness function and this best value, respectively. On the swarm scale, a *gBestPosition* and a *gBestValue* refer to the best particle among the swarm and its fitness function value.

### B. DISCRETE PARTICLE SWARM OPTIMIZATION

As a Swarm Intelligence algorithm, PSO searches for the global best position through an iterative process. The position of each particle in the swarm is updated during the iterative process until reaching the near-optimal solution. These positions are updated based on the speed of particles, which in

turn is also updated over the iterations of PSO according to eq. 1. This update from the current speed  $V_t$  to the new one  $V_{t+1}$  is affected by the current particle position  $X_t$  compared to both the best particle position  $pBestPosition$  and the global best position among the whole swarm  $gBestPosition$ . Therefore, the current position of the particle  $X_t$  can be updated to  $X_{t+1}$  based on  $V_{t+1}$  as depicted in eq. 2. These previous calculations lead to real numbers that are appropriate for continuous search space problems.

On the other hand, some problems have discrete search domains, thus a discrete binary version of PSO has been introduced [22]. In the VM placement problem, the proposed technique searches for the near-optimal solution to map the migrated VMs to the available PMs. Consequently, an approximation will be applied in the proposed work to convert the real numbers into integer ones that refer to indexes of the available PMs. In other words, calculations of  $V_{t+1}$  will be rounded to the nearest integer, thereby it is added to the current position to get a new one. Additionally, a strict constraint is adopted to ensure that the new position will represent a valid PM index as shown in eq. 3, where  $i_0$  and  $i_n$  are the indexes of the first available PM and the last one, respectively.

$$V_{t+1} = (w \times V_t) + (r_1 \times c_1 \times (pBestPosition - X_t)) + (r_2 \times c_2 \times (gBestPosition - X_t)) \quad (1)$$

$$X_{t+1} = X_t + V_{t+1} \quad (2)$$

$$i_0 \leq X_t \leq i_n \quad (3)$$

### III. RELATED WORKS

Recently, consolidation of VMs has attracted many researchers to introduce effective algorithms [5], [23] that can enhance power management in data centers. These algorithms dynamically reallocate VMs among servers, where VMs can be consolidated into a fewer number of hosts. To the best of our knowledge, Nathuji and Schwan [24] introduced one of the first contributions for an effective power management architecture for virtual data centers. Their architecture, Virtual Power Management (VPM), can consolidate VMs based on global policy. In general, dynamic consolidation of VMs into a fewer number of servers can be divided into four sub-problems [20] to decide the following: overloaded servers, underloaded servers, VM selection, and VM placement.

#### A. DETECTION OF OVERLOADED AND UNDERLOADED SERVERS

To minimize amount of the consumed power in data centers without violating SLA, Beloglazov *et al.* [25] proposed a technique based on a static threshold to ensure that the total CPU utilization of hosts will remain always between two thresholds. Thus, they migrate the whole VMs from underloaded servers, if the server's CPU utilization is under the lower threshold. Thereby, they can reduce power

consumption by switching these hosts to the sleep mode. On the other hand, they avoid the violation of SLA by keeping suitable free resources through migrating some VMs from the overloaded hosts that have a CPU utilization greater than the upper threshold.

It is noticed that static thresholds are not suitable enough to manage servers with dynamic workloads. Therefore, adaptive thresholds have been used based on a statistical analysis of the historical data, which have been collected during the VMs lifetime [20]. Farahnakian *et al.* [26] proposed another technique for predicting the future utilization of CPU according to the linear regression approach. Gray-Markov model has been employed to predict the future CPU utilization [27]. Gradient descent-based regression (Gd) [28] has been proposed to detect overloaded servers based on a machine learning technique. M estimator regression (MeReg) [29] has been introduced to calculate the upper threshold for hosts based on the historical data about CPU utilization. In contrast, other techniques do not rely on thresholds to identify overloaded or underloaded hosts such as [30], [31]. Although several kinds of research take CPU utilization of servers as the only parameter in their algorithms, some studies take additional resources into their consideration such as memory, bandwidth, and disk [30], [32]–[35].

#### B. VM SELECTION

Different algorithms have been proposed to decide the set of VMs which have to be reallocated from overloaded servers [20], [25]. Random choice is based on selecting some VMs randomly to be migrated. Minimization of Migrations (MM) depends on choosing for migration the minimum number of VMs to lower the CPU utilization among overloaded hosts under an upper threshold. Highest Potential Growth (HPG) is proposed to avoid SLA violation by selecting the ones that use CPU minimally compared to the needed CPU in these VMs. Minimum Migration Time (MMT) chooses VMs that require the least time to finish the migration, thereby it is related to the utilized portion of RAM by VMs and the available network bandwidth. Additionally, a similar selection technique [36] is applied to choose VMs that have the least RAM demand, and thus least MMT.

Maximum Correlation (MC) has been proposed to migrate the most correlated VMs of CPU utilization with other VMs. Maximum Requested Bandwidth (MBW) [37] chooses VMs for migration from overloaded hosts taking into consideration the maximum requested bandwidth for VMs. Also, Minimized Square Root (MISR) [38] adopts a technique that selects VMs according to their minimal impact on the load. Maximum Utilization Minimum Size (MuMs) [39] selects those VMs with the high CPU utilization and the minimum size. Minimum Utilization Prediction (MuP) [40] collects the CPU utilization, in the overloaded hosts, for each VM at different time frames. The Median Absolute Deviation (MAD) can be applied to predict those VMs with the least workloads. On the other hand, fuzzy logic [41] has been

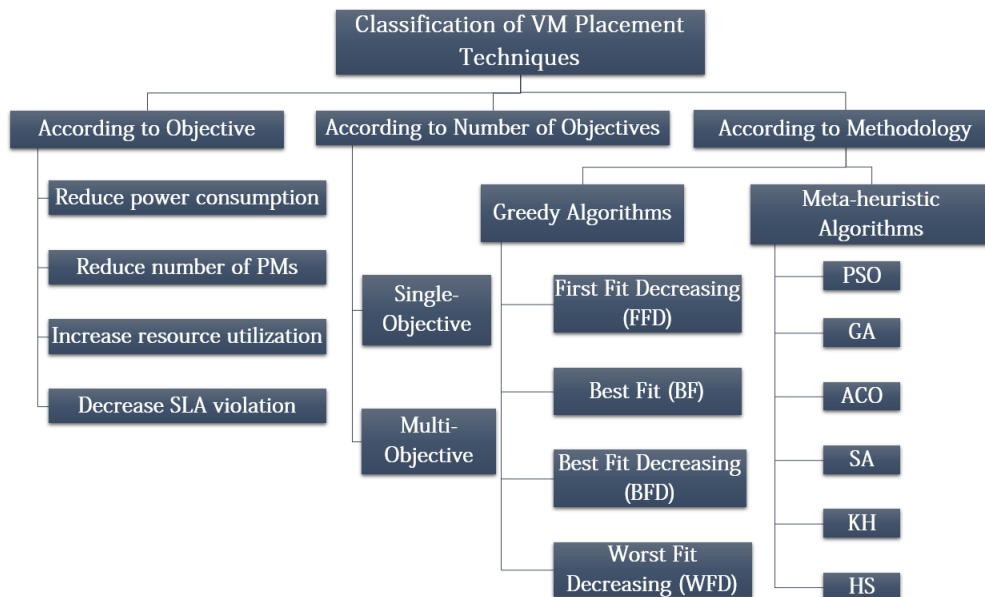


FIGURE 2. A taxonomy of VM placement techniques.

proposed for VM selection to have the advantage of taking an intelligent decision.

C. VM PLACEMENT

Deciding appropriate places to receive the migrated VMs is a significant issue for a good VM consolidation. However, VM placement problem is not related only to the VM consolidation scenarios, where the most appropriate server is selected to receive the migrated VM. There is another scenario for VM placement by finding a suitable host to receive the VM as an initial placement. As our proposed technique is related to VM placement from the VM consolidation point of view, our focus is to highlight the previous works that are related to VM consolidation.

Many studies have been proposed to introduce effective techniques based on different methodologies aiming to achieve various goals. Figure 2 provides a general taxonomy for VM placement techniques according to the adopted methodologies, objectives, and number of these objectives. According to the number of objectives, some techniques try to find the best mapping that achieves only one goal, while others have more than one objective. These objectives may be related to increasing the QoS that is provided to users, reducing costs for service providers or both. According to the adopted methodology, many techniques have employed heuristic and meta-heuristic algorithms to decide which PM will receive the migrated VM.

Different greedy heuristic algorithms can be exploited such as First Fit (FF), where each VM is mapped to the first host that fits it. Best Fit (BF) which maps VMs to the best PMs that can fit them. Best Fit Decreasing (BFD) and First Fit Decreasing (FFD) are similar to BF and FF, but they are preceded by VMs sorting in a decreasing order. On the other hand, there are also some meta-heuristic methods that

have been proposed to find the near-optimal places for VMs such as Ant Colony Optimization (ACO) [42], [43], Genetic Algorithm (GA) [44], [45], Simulated Annealing (SA) [46], Improved Lévy based on Whale Optimization Algorithm (ILWOA) [47], Glowworm Swarm Optimisation [48], Harmony Search (HS) [49], Krill Herd (KH) algorithm [50], and hybridized algorithm [51]. A recent detailed work has been introduced to review the state-of-the-art multi-objective techniques based on meta-heuristic algorithms [52]. Moreover, another recent extensive survey has been devoted to study the VM placement with a wide exploration of single and multi-objective techniques [19].

Anton Beloglazov and Rajkumar Buyya [20] relied on a modified version of the heuristic algorithm BFD to decide the appropriate VM placement, which is named as Power-Aware Best Fit Decreasing (PABFD). They sort the migrated VMs in a decreasing order according to the CPU utilization. After that, they allocate each VM to the PM that will consume the minimum amount of power due to the allocation. Another VM placement policy [53] has been proposed based on BFD, which is called Space Aware Best Fit Decreasing (SABFD). It depends also on the decreasing sorting for the migrated VMs according to CPU utilization, where the PM which has enough computing resources will receive the first VM. After placing the first VM, PM with the least available computing resources will be selected as a destination host for the second VM and so on until completing the placement of all VMs. Modified Multi-weights Best fit (MW-BF) [35] has been applied based on a modification of the BF algorithm, where they consider multi-weights for VMs which need a placement.

Malekloo et al. [43] proposed a multi-objective approach based on ACO for VM placement and consolidation. They relied on two complementary algorithms to achieve

a trade-off between energy consumption and Quality of Service (QoS). Their VM placement algorithm aimed to reduce energy consumption, communication energy cost and resource wastage. Their algorithm depends on the Pareto front method to find the non-dominated solutions that reduce their objectives. The algorithm was tested in CloudSim and compared to three single objective algorithms (FFD, DVFS, and LR) and one multi-objective algorithm (MGA). They claimed that their algorithm can achieve energy saving with about 39.19 % on average. However, increasing the number of objectives may come at the expense of time complexity and system performance.

Mohammadhosseini et al. [54] proposed an energy-aware VM placement procedure based on the Cultural Algorithm (CA). They called their proposed algorithm Balance-based Cultural Algorithm for Virtual Machine Placement (BCAVMP). The authors defined a parameter known as the server balance vector for VM placement solutions related to each server. Their fitness function relied on two parameters, which are the sum of balance vector lengths for each feasible solution and the total amount of consumed energy. Therefore, they can minimize the resource wastage and power consumption in data centers. The authors evaluated their proposed method using CloudSim simulator, they compared it with the well-known VM placement algorithm (PABFD). The obtained results for their experiments showed that BCAVMP can achieve an acceptable reduction in the consumed energy, the number of active servers and the number of VM migrations. However, this reduction came at the expense of SLA.

Fatima et. al [55] introduced a hybrid algorithm of Levy flight and Multi-Objective Grey Wolf Optimization (LMOGWO) to solve the VM placement problem. The proposed algorithm was inspired by grey wolves and mimicked their natural behaviors in hunting and leadership. It attached an archive to save the non-dominant solutions. The best three solutions are considered leader wolves that guide other wolves to attack the prey. Those leaders are known as alpha, beta, and delta, while the remaining solutions are denoted as omega wolves. The proposed algorithm uses the levy flight to calculate the step size for the wolves. Nine standard benchmark functions are used for testing the proposed algorithm.

Due to its ability to benefit from exploration and exploitation by applying the local and global searches, PSO has been adopted in more than a situation to find the optimal VM placement. Aiming to minimize the consumed power, Wang et al. [56] redefined parameters of PSO, in addition to, proposing a local fitness first strategy. Furthermore, they designed an encoding scheme for particles based on two dimensions. Dashti and Rahmani [57] reallocated the migrated VMs from the overloaded hosts, in addition to, consolidating the underloaded hosts to save power. Also, they embedded the consumed power and the migration cost in the fitness function. According to their mapping, the host and its VMs representing a particle and its dimensions, respectively. In contrast, dimensions of each particle in our proposed

mapping represent VMs that need to be migrated. A multi-objective VM placement based on PSO was proposed in [58] to increase the resource utilization and minimize the power consumption. To improve the global search ability and increase the convergence efficiency, an enhanced PSO [59] is applied by updating particle velocities using Lévy flight which is known as Particle Swarm Optimization with Lévy Flight (PSOLF). Table 1 highlights some of the related works and refers to their optimizations and limitations.

VM placement is an important issue for an efficient VM consolidation. Therefore, a power-aware VM placement technique will be introduced to reduce the consumed power in the data center. Although many VM placement techniques have been devoted to finding the most appropriate servers for hosting the migrated VMs, some techniques focused on reducing the consumed power at the expense of SLA. Moreover, some previous works have adopted heuristic techniques that can not always produce the optimal placement solution. Despite being a leading factor in the amount of consumed energy, some techniques neglected CPU utilization and took other resources in their consideration. Consequently, we propose a VM placement technique based on the particle swarm optimization algorithm to reduce power consumption in the data center without violating SLA. In its search for the near-optimal placement for VMs, the proposed technique focuses on CPU utilization in servers. It uses a discrete version of the Particle Swarm Optimization (PSO) algorithm, where PSO in its ordinary form is suitable for continuous problems. Specifically, an appropriate decimal encoding is adopted to allow PSO to deal with the VM placement problem.

#### IV. PROBLEM STATEMENT

Virtual Machine Placement (VMP) is a major issue to have an efficient consolidation of VMs. It deals with deciding the most appropriate PMs to receive the migrated VMs. This issue is considered as a bin packing optimization problem, where VMs can be considered as objects that should be consolidated into the minimum number of PMs which viewed as bins [64]. However, deciding the most appropriate manner to map the migrated VMs to the most suitable PMs is not a trivial mission [65]. Specifically, it is assumed that a set of migrated VMs  $V = \{VM_1, VM_2, VM_3, \dots, VM_m\}$  should be mapped to the set of available PMs  $P = \{PM_1, PM_2, PM_3, \dots, PM_n\}$ , where  $m$  and  $n$  are the number of migrated VMs and the number of available servers, respectively. It is also assumed that a migrated VM denoted as  $VM_{mg}$  requires the following amounts of resources  $CPU_{mg}$ ,  $RAM_{mg}$ ,  $Bw_{mg}$  for CPU, memory, and bandwidth. On the other hand, a  $PM_j$  has the following capacities of resources  $CPU_j$ ,  $RAM_j$ ,  $Bw_j$  for CPU, memory and bandwidth. A  $PM_j$  can receive a  $VM_{mg}$  if and only if the free resources of  $PM_j$  exceeds that needed by  $VM_{mg}$  as shown in equations 4, 5, and 6, where  $k$  is the number of currently allocated VMs to  $PM_j$ .

$$\sum_{i=1}^k CPU_i + CPU_{mg} < CPU_j \quad (4)$$

TABLE 1. Comparison of different VM placement techniques.

Reference	Methodology	Considered Resources	Experimental Environment	Compared To	Technique	Optimization	Limitation
[25]	BFD	CPU	CloudSim	NPA and DVFS	A modified BFD (MBFD) that allocates a VM to the PM that produces the least amount of energy.	Introducing one of the leading algorithms for VM placement (PABFD) to reduce power consumption and implementing it in CloudSim.	As a heuristic technique, it can not always produce the optimum solution.
[45]	GA	CPU and network communications	Java	FFD	Genetic Algorithm is applied with a focus on the consumed power in the hosts and network communications in a three-tier data center.	Energy consumption is decreased with 10.08% on average.	SLA violation is not considered and simulations are based on their own model.
[43]	ACO	CPU and network elements	CloudSim	FFD, DVFS, LR and MGA	It adopts a multi-objective ACO (MACO) to reduce power consumption, resource wastage and costs of communication energy.	Saving about 39.19% of energy consumption on average compared to FFD.	Increasing the number of objectives leads to time complexity and performance degradation.
[49]	HS	CPU	CloudSim	MBFD and ACS-VMC	A multi-objective VM placement based on Harmony Search Algorithm (HSA) to reduce power consumption, number of migrations and number of active hosts.	According to their results, energy consumption, SLA violation, number of migration and ESV are reduced.	Simulation Experiments depends only on a one state of the random workload.
[50]	KH	CPU	CloudSim	MBFD and GA	A single objective technique based on Krill Herd algorithm to reduce the power consumption.	Reducing about 35% of energy consumption, compared to MBFD.	SLA is violated and ESV gives bad results compared to MBFD and GA.
[54]	CA	CPU, memory and BW	CloudSim	PABFD	An energy aware method for VM placement based on cultural algorithm to reduce resource wastage and power consumption.	Reducing energy consumption with about 26.11% on average.	Simulation results showed that SLA is violated.
[60]	BFD	CPU	CloudSim	MBFD	Similar to PABFD, but binary search is used to sort the PMs list increasingly before finding the appropriate PM.	Decreasing both energy consumption and SLA violation.	The optimum solution can not be guaranteed in a greedy algorithm. Additionally, the number of host shutdowns and the number of VM migrations are worse in some models.
[61]	FFD	Memory	CloudSim	VM-static algorithm	It studies the history of memory consumption to decide the appropriate PM based on estimation of memory consumption.	Saving about 17.08% of PMs.	It focuses on memory consumption only.
[62]	PSO	CPU and memory	OpenStack	OpenStack native scheduler	It focuses on decreasing the number of active PMs by increasing CPU and memory utilization.	Decreasing 50% of PM consumption, in addition to, increasing memory utilization of PMs by two times.	SLA violation is neglected in the objective function.
[63]	BF	CPU	CloudSim	PABFD	Energy and Resource-Aware VM Placement (ERAP) allocates a VM to the PM that produces both a minimum energy and a utilization which does not exceed 80%.	Minimized about 13.12% of the energy consumption.	The optimum VM placement can not be guaranteed.

$$\sum_{i=1}^k RAM_i + RAM_{mg} < RAM_j \tag{5}$$

$$\sum_{i=1}^k Bw_i + Bw_{mg} < Bw_j \tag{6}$$

Furthermore, a VM can not be hosted by more than one PM at the same time, thus a  $VM_i$  can be mapped to only one server. The mapping of migrated VMs is represented in the proposed technique by a particle  $P_i = [7, 3, 3, 4, \dots, 1]$ . This particle is simply just a vector contains a number of elements that refers to the number of migrated VMs. Each element in the vector has a value represents the index of the PM that will host the VM. According to the values of  $P_i$ , the first migrated VM will be placed in the seventh available PM and the last one will be hosted by the first server. Similarly, the second and the third migrated VMs will be hosted by the third PM, where a PM can host several VMs but the opposite is not true.

With the growing in the number of active servers, the amount of consumed energy increases [8]. Consequently, the objective is to place the migrated VMs into the minimal number of servers, but also with a major constraint to avoid increasing the number of overloaded servers as much as possible. Therefore, the proposed technique has mainly two objectives to reduce both the number of active servers and

the number of overloaded servers as declared in equations 7 and 8, where  $x$  and  $y$  are the number of running hosts and the number of overloaded hosts, respectively. In other words, the best solution for mapping the migrated VMs is the one that decreases both the number of running PMs and the number of overloaded PMs as much as possible. Reducing the power consumption in hosts is guaranteed through increasing the CPU utilization and thereby consolidating VMs into a minimum number of servers, which allows switching the unused PMs to a sleep mode. On the other hand, reducing the number of overloaded hosts ensures to provide VMs with the required resources, thus the QoS that is provided to users can be increased by avoiding SLA violation.

$$\min \sum_{i=1}^x PM_i \tag{7}$$

$$\min \sum_{i=1}^y PM_i \tag{8}$$

## V. SYSTEM ARCHITECTURE

Cloud computing usage has expanded, based on the pay-as-you-go model. Our study focuses on the IaaS environment, where large-scale data centers consist of many heterogeneous servers. Multiple users can benefit from these servers, which can provide independent VMs to run their applications

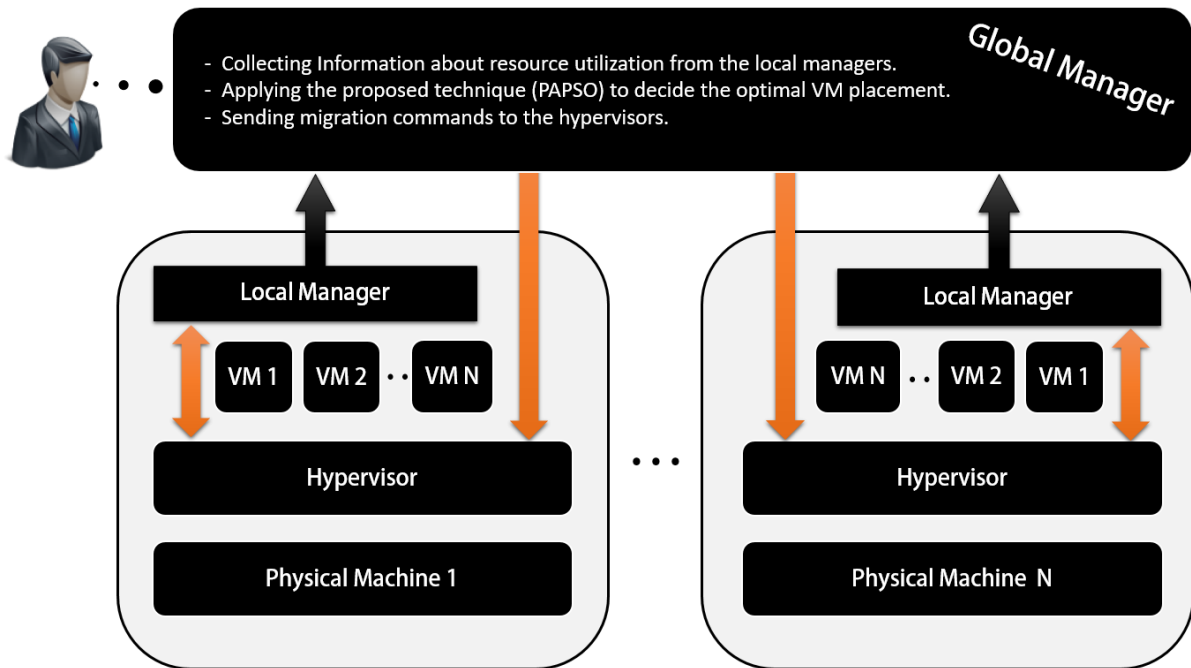


FIGURE 3. System architecture.

securely. The major server characteristics are as follows: CPU performance which declared as Million Instructions Per Second (MIPS), RAM, network bandwidth and storage which is shared between servers.

Virtualization technology has allowed each PM to host multiple VMs and provide them with the required resources. Each PM has a software layer to manage VMs, known as a hypervisor. For power management purposes, VMs are consolidated into the minimal number of active hosts without violating SLA by applying live VM migration. The resource utilization in each server is monitored continuously by a module called the local manager. Also, there is a global manager located in the master node to communicate with the local managers to have a complete view of the resource utilization. Therefore, it can decide the necessary migrations for VMs to decrease the overall power consumption in the data center as shown in Figure 3. The system architecture of our proposed technique can be organized as follows:

- 1) The local manager in each server monitors the resource utilization of VMs periodically.
- 2) The local manager decides overloaded and underloaded servers. Then, it selects the most appropriate VMs to be migrated from the overloaded hosts.
- 3) The global manager receives information about the resource utilization from the local managers.
- 4) The global manager uses the power-aware VM placement technique based on particle swarm optimization to decide the optimal placement for the migrated VMs.
- 5) The global manager sends migration commands to hypervisors, thereby VM migrations can be accomplished and idle servers are kept in a low power mode.

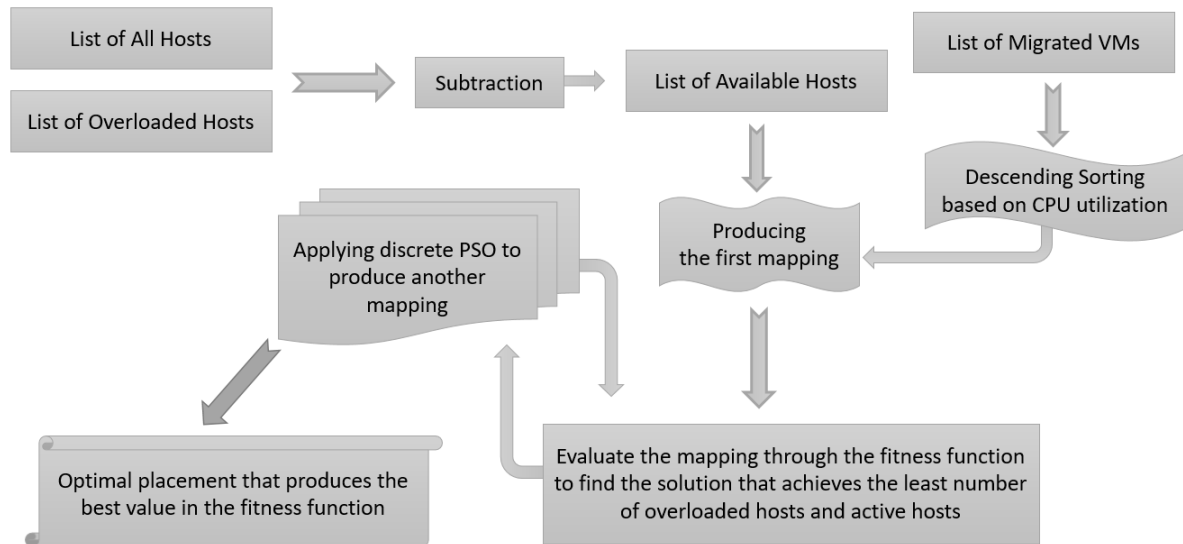
## VI. PROPOSED TECHNIQUE

After detecting overloaded and underloaded servers, and selecting some VMs to be migrated from the overloaded servers, there is a critical mission to find appropriate places for the migrated VMs. Thus, this study proposes a VM placement technique based on PSO with a purpose to find a near-optimal solution as shown in Figure 4. This technique receives a list of migrated VMs that need an appropriate placement, in addition to, another list of available hosts that can host the migrated VMs. Although PSO is considered as one of the most effective population-based algorithms, it deals with continuous problems. Therefore, an appropriate encoding has to be applied to convert the problem into a discrete one.

The pseudo-code of our proposed technique is explained in Algorithm 1, where its notations are described in Table 2. The algorithm starts with initializing the proposed technique parameters (line 1). These parameters are number of particles, number of dimensions, number of iterations, cognitive and social parameters, the minimum and maximum limits of inertia weight. Then, VMs that search for the optimal placement will be sorted in a descending order according to the CPU utilization (line 2). The initialization phase will be declared through subsection VI-A. The third step excludes overloaded servers from the list of available hosts for receiving the migrated VMs (line 3). As PSO relies on assigning positions to each particle, the minimum and maximum available values for positions and velocities are defined according to the lower and upper indexes in the list of available hosts (line 4).

The algorithm searches for the best particle among the swarm, where particles are evaluated through a function





**FIGURE 4.** The proposed technique to find the near-optimal VM placement.

called the fitness function as declared in subsection VI-C. In the studied case, the technique searches for the minimum value of the fitness function. Therefore, the best value of the swarm is initialized with the maximum value (line 5). The algorithm initializes all particles (lines 6-10), where each particle is initialized with a random position and a random speed. The best value for each particle is initialized also with the maximum value. The proposed technique mapping is discussed through subsection VI-B.

After initializing the swarm, it moves through a number of iterations until reaching the optimal solution as shown in subsection VI-D (lines 12-36). Before moving to the iterative stage, the iteration number is initialized with zero (line 11). The algorithm ensures that each VM is assigned to a valid server through the server index (lines 13-19). If the particle position exceeds the upper available position, VM will be assigned to that server with the highest index. In a similar fashion, VM will be assigned to that server with the lowest index if the particle position is lower than the minimum available index.

After ensuring the validity of positions for all particles, each solution is evaluated (line 19). If the new fitness function value of a particle is better than its best value, the particle's best value and best position will be replaced (lines 21-24). The previous process is repeated to determine the best fitness function value among the swarm and the position that leads to this best value (lines 25-28). The next step is to update particle positions (lines 29-35). The inertia weight is calculated which relies on the current iteration number (line 30). A new speed and thus a new position are calculated for each particle using the particle swarm optimization equations (lines 34-35). The iteration number is incremented before moving to the next iteration (line 36). At the end of the iterations, the best particle is determined. Thus, dimensions of the best particle can be used to decide the optimal mapping for each VM (line 37-39).

The time complexity analysis of the proposed algorithm will be introduced in subsection VI-E.

#### A. INITIALIZATION PHASE

Like any algorithm based on PSO, the proposed technique should be started by initializing the parameters. The population size and number of iterations have to be defined, where the population size refers to the number of particles and each particle indicates a mapping solution. Also, each particle contains a number of dimensions representing the number of migrated VMs that have to be placed. Besides, values of PSO parameters should be determined before applying the algorithm. These parameters include the following, an inertia weight coefficient, a cognitive parameter, and a social parameter. As CPU utilization is the major factor in the context of power consumption, the migrated VMs are sorted according to CPU utilization. Furthermore, the value of the best position for each particle and the global best value among the swarm are initialized with the maximum values.

#### B. PAPS0 MAPPING

The proposed technique takes three lists as inputs representing migrated VMs, total servers, and overloaded servers. Therefore, migrated VMs can be mapped to any PM from the available PMs, where a list of the available hosts is generated by excluding overloaded servers from the list of total hosts. PAPS0 produces an initial mapping solution which is denoted as a particle, where this particle consists of several dimensions equivalent to the number of migrated VMs. Each dimension keeps a decimal value that refers to the index of a PM in the list of available hosts. Therefore, it can be said that these values of the indexes represent the particle position. Additionally, an index can be repeated in more than one dimension in the particle, where a PM can host several VMs.

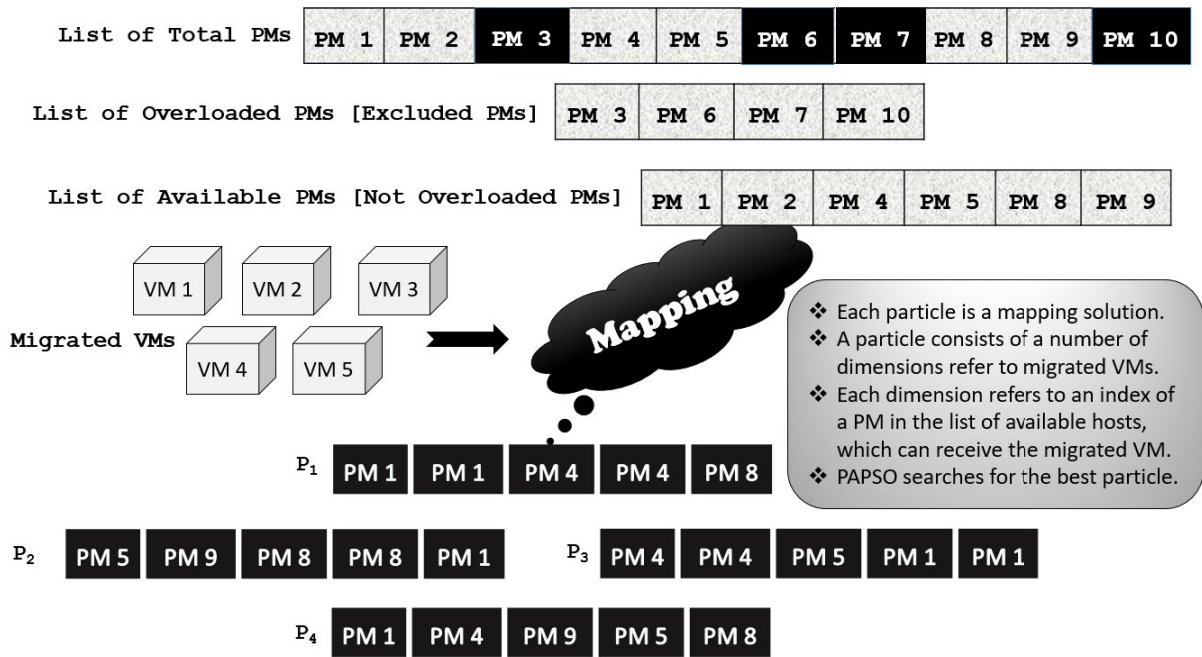


FIGURE 5. The Mapping of migrated VMs according to PAPS0.

For controlling the movement of particles, each particle has a speed which can be updated. Thus, an initial value is set for each dimension represents the particle speed, thereby applying the particle speed on its position must lead to a new position representing an index in the available host list. This initial mapping is evaluated according to our placement objective, thus a new mapping can be generated and evaluated and so on until reaching the final mapping. Figure 5 explains the mapping of migrated VMs from the PAPS0 point of view. According to the illustrated example, four PMs denoted as  $PM_3$ ,  $PM_6$ ,  $PM_7$  and  $PM_{10}$  are overloaded servers. Therefore, PAPS0 excludes these servers from the list of available servers. Consequently, particles can be generated periodically to map the migrated VMs to appropriate PMs from the list of available hosts.

### C. FITNESS FUNCTION

To evaluate the quality of solutions, swarm relies on a fitness function. Based on the values of this function, particles can update their positions repeatedly until reaching the best value. This significant function can be called a fitness function or an objective function, interchangeably. Its equation changes from one problem to another with a goal to get the optimum solution. Also, it can be a minimization function or a maximization one according to our goal. Here, a minimization function is proposed that seeks to decrease the number of active hosts as much as possible to increase the CPU utilization of servers, thereby minimize the total power consumption. Meanwhile, another factor is taken into consideration that represents the number of overloaded hosts after placement which should be also minimized.

Due to its simplicity and efficiency in the context of multi-objective problems, the weighted sum method is adopted to solve many multi-objective problems. It depends on assigning a weight to each objective, which in turn is multiplied by the cost value of this objective. Thereby, applying a cumulative addition to the multiplication processes produces the total cost of the solution as shown in eq. 9. Its goal is to find the total cost of a problem that has  $n$  objectives, where  $w_i$  and  $c_i$  refers to the weight and the cost value of the  $i_{th}$  objective, respectively. Generally, the summation of all weights has to be 1 and each weight falls in this range [0, 1]. Therefore, choosing appropriate values for weights has a significant effect on the overall cost value. By adopting the weighted sum method, fitness function can produce total cost values for solutions in a simple fashion as if it deals with a single-objective problem. In the proposed technique, the weighted sum is applied in the fitness function  $F_n$  to merge two objectives based on two weight factors  $w_1$  and  $w_2$  as shown in eq. 10.

The proposed technique tries to reduce the consumed power by mapping the migrated VMs to the minimum number of PMs, but also it avoids as possible to produce overloaded servers to keep the SLA. These goals can be achieved through the introduced fitness function that benefits from the capability of the weighted sum method to consider the problem as a composite objective function. Specifically, the proposed fitness function searches globally for the near-optimal mapping that consolidates VMs into the minimal number of PMs to reduce the power consumption without increasing the number of overloaded servers. Therefore, it can reduce both the number of underloaded servers and the number of overloaded

**Algorithm 1** Proposed PAPS0 Technique

```

Input :  $HL, Ex - HL$  and  $VML$ 
Output:  $MP$ 
1 Initializing parameters:  $N_p, N_d, N_t, W_{min}, W_{max}, c_1, c_2$ 
2 Descending sorting for the migrated VMs according to CPU utilization
3 Producing a list of the available hosts for receiving the migrated VMs
4 Initializing  $X_{min}, X_{max}, V_{min}$ , and  $V_{max}$  based on indexes of the available hosts list
5  $gBestValue \leftarrow \text{MaxValue}$ 
6 foreach  $p \in P$  do
7    $pBestValue \leftarrow \text{MaxValue}$ 
8   foreach  $vm \in VML$  do
9      $X_0 \leftarrow \text{rand}(X_{min}, X_{max})$ 
10     $V_0 \leftarrow \text{rand}(V_{min}, V_{max})$ 
11  $t \leftarrow 0$ 
12 while  $t < N_t$  do
13   foreach  $p \in P$  do
14     foreach  $vm \in VML$  do
15       if  $X_t > X_{max}$  then
16          $X_t \leftarrow X_{max}$ 
17       else if  $X_t < X_{min}$  then
18          $X_t \leftarrow X_{min}$ 
19       Calculate  $F_n$  for  $p$  based on eq. 10
20   foreach  $p \in P$  do
21     if  $F_n < pBestValue$  then
22        $pBestValue \leftarrow F_n$ 
23       foreach  $vm \in VML$  do
24          $pBestPosition \leftarrow X_t$ 
25     if  $F_n < gBestValue$  then
26        $gBestValue \leftarrow F_n$ 
27       foreach  $vm \in VML$  do
28          $gBestPosition \leftarrow X_t$ 
29   foreach  $p \in P$  do
30      $W \leftarrow W_{max} - (W_{max} - W_{min}) \times \frac{t}{N_t}$ 
31     foreach  $vm \in VML$  do
32        $r_1 \leftarrow \text{random}(0, 1)$ 
33        $r_2 \leftarrow \text{random}(0, 1)$ 
34        $V_{t+1} \leftarrow [(w \times V_t) + (r_1 \times c_1 \times (pBestPosition - X_t)) + (r_2 \times c_2 \times (gBestPosition - X_t))]$ 
35        $X_{t+1} \leftarrow X_t + V_{t+1}$ 
36    $t \leftarrow t + 1$ 
37 foreach  $vm \in VML$  do
38    $MP \leftarrow \text{map}(vm, gBestPosition)$ 
39 return  $MP$ 

```

**TABLE 2.** Notations of PAPS0.

Notation	Description
$HL$	List of hosts
$Ex - HL$	List of excluded hosts in the VM placement
$VM$	A virtual machine
$VML$	List of migrated VMs that need appropriate hosts
$MP$	Migration Plan
$N_p$	Population size (Particles number)
$N_d$	Number of dimensions (Number of migrated VMs)
$N_t$	Number of iterations
$W_{min}$	Minimum value of inertia weight coefficient
$W_{max}$	Maximum value of inertia weight coefficient
$W$	Inertia weight coefficient
$c_1$	Cognitive parameter
$c_2$	Social parameter
$r_1$	A random value between 0 and 1
$r_2$	A random value between 0 and 1
$X_{min}, X_{max}$	Limits of the position for each particle
$V_{min}$	Minimum speed for the particle
$V_{max}$	Maximum speed for the particle
$p$	A specified particle in the swarm
$P$	Swarm of particles
$F_n$	Value of the fitness function for a particle
$w_1, w_2$	Applied weights in the fitness function
$pBestValue$	Best $F_n$ value for a particle
$gBestValue$	Best $F_n$ value among all particles
$pBestPosition$	Position of particle which produce the best $F_n$ value
$gBestPosition$	Best position for VMs among the swarm
$t$	The current iteration
$X_0$	Initial position for the particle (generated randomly)
$V_0$	Initial speed for the particle (generated randomly)
$X_t$	Current position of a particle
$X_{t+1}$	Particle position in the next iteration
$V_t$	Current speed of a particle
$V_{t+1}$	Particle speed in the next iteration

reasons, overloaded servers and underloaded ones.

$$TotalCost = \sum_{i=1}^n w_i \times c_i \tag{9}$$

$$F_n = \min\left\{w_1 \times \frac{\text{Number Of Active Hosts}}{\text{Total Number of Hosts}} + w_2 \times \frac{\text{Number Of Overloaded Hosts}}{\text{Total Number Of Hosts}}\right\} \tag{10}$$

**D. ITERATIVE PROCESS OF PAPS0**

To reach a near-optimal VM placement, PAPS0 passes through several iterations. Each particle updates its speed in each iteration, thus it can update its position. The speed modification for particles takes into consideration the current position compared to both the best particle position and the best position among the whole swarm. Calculation results for the new position and the new speed are rounded to the nearest integer to be translated to an index in the list of available hosts. According to the new positions, the fitness function is calculated for each particle. Produced values from these calculations help particles in their exploration process for the best solutions. Therefore, the swarm can exploit this situation to find the optimum solution.

**E. TIME COMPLEXITY ANALYSIS**

The time complexity of our proposed algorithm will be introduced in this subsection, but let us assume that:  $n$  is the total

ones. Consequently, the proposed technique can decrease the number of migrations drastically through avoiding its two

**TABLE 3.** The evaluated test cases.

Test Case	$C_1$	$C_2$	$C_3$
Number of PMs	50	75	100
Number of VMs	50	75	100

number of hosts;  $m$  is the number of migrated VMs;  $N_p$  is the number of particles;  $N_t$  is the total number of iterations. The time complexity of each part in the algorithm is defined as follows:

- Initializing PAPSO parameters:  $O(1)$ .
- Producing the list of available hosts:  $O(n)$ .
- Sorting the VMs:  $O(m \times m)$ .
- Initializing  $X_{min}$ ,  $X_{max}$ ,  $V_{min}$ , and  $V_{max}$ :  $O(1)$ .
- Initializing  $gBestValue$ :  $O(1)$ .
- Initializing  $X_0$  and  $V_0$ :  $O(N_p \times m)$ .
- Initializing the iteration number:  $O(1)$ .
- Ensuring that particle positions within the valid range:  $O(N_t \times N_p \times m)$ .
- Calculating the fitness function for each particle:  $O(N_t \times N_p)$ .
- Updating  $pBestValue$  for each particle:  $O(N_t \times N_p)$ .
- Updating  $pBestPosition$  for each particle:  $O(N_t \times N_p \times m)$ .
- Updating  $gBestValue$ :  $O(N_t \times N_p)$ .
- Updating  $gBestPosition$ :  $O(N_t \times N_p \times m)$ .
- Calculating the inertia weight:  $O(N_t \times N_p)$ .
- Updating the new speeds and positions:  $O(N_t \times N_p \times m)$ .
- Incrementing the iteration number:  $O(N_t)$ .
- Producing the final mapping:  $O(m)$

Consequently, the total time complexity for the proposed algorithm is  $O(N_t \times N_p \times m)$

## VII. SIMULATION RESULTS AND ANALYSIS

All simulation experiments are performed on a PC that has the following characteristics, Intel Core i7-2670QM CPU @ 2.20 GHz processor and 6 GB of RAM based on Ubuntu 18.04 LTS with Eclipse IDE. Experiments are performed based on three test cases that include different sizes of both VMs and PMs using random workloads as shown in Table 3. To ensure the trustiness of results, each experiment is executed 10 times. Thus, an accurate result can be provided based on the average of these results, then compared to the well-known VM placement algorithm PABFD. The parameter values, that were applied in the experiments, are tabulated in Table 4. As our experiments are based on a large number of PMs and VMs, it is not applicable or cost-effective to apply our repeated experiments, to test the proposed technique, on a large-scale real environment. Instead, a simulation environment for testing the proposed VM placement technique is used. Specifically, our choice was the simulation platform CloudSim toolkit, the commonly used simulator in these similar scenarios.

CloudSim [66] is an open-source toolkit, developed by CLOUDS Laboratory in the University of Melbourne, that written in Java. It provides a suitable environment that

**TABLE 4.** Parameter values for PAPSO technique.

$W_{min}$	$W_{max}$	$N_p$	$N_t$	$c_1$	$c_2$
0.4	0.9	30	50	2.05	2.05

**TABLE 5.** Characteristics of servers.

Server Type	Frequency of core (MIPS)	No. of cores	Memory (GB)	Bandwidth (Gb/s)	Storage (GB)
HP ProLiant ML110 G4	1860	2	4	1	1000
HP ProLiant ML110 G5	2660	2	4	1	1000

**TABLE 6.** Characteristics of VMs.

VM Type	Frequency (MIPS)	No. of cores	Memory (MB)	Bandwidth (Mb/s)	Storage (GB)
High-CPU VM	2500	1	870	100	2.5
Extra Large VM	2000	1	1740	100	2.5
Small VM	1000	1	1740	100	2.5
Micro VM	500	1	613	100	2.5

includes all required entities of cloud computing IaaS to apply our repeatable experiments. Therefore, a method called *getNewVmPlacement*, located in a class known as *PowerVmAllocationPolicyMigrationAbstract*, is extended to implement our proposed technique. Two types of servers and four types of VMs are simulated in the experiments, whose characteristics are described in Table 5 and 6, respectively. According to [20], the combination of Local Regression (LR) and MMT has proved its superiority to decide the overloaded hosts and select VMs for migration from these overloaded hosts, respectively. Consequently, the LrMMT combination is adopted in our experiments. Performance metrics have shown that PAPSO outperforms PABFD as follows.

### A. ENERGY CONSUMPTION

Although CPU, memory, disk storage, network interfaces, and cooling system of servers consume power, CPU is considered the leading component in consuming power in data centers. According to [8], [67], the power consumption in servers can be computed based on a linear relationship between CPU utilization of servers and the consumed power.

Our simulation experiments relied on two types of servers, where the power consumption of these two hosts can be calculated based on data of the power model provided by the SPECpower benchmark for HP ProLiant ML110 G4 [68] and HP ProLiant ML110 G5 [69] as declared in Table 7. Therefore, the total consumed energy by a typical server can be calculated based on the integral relationship of the power consumed over a time interval as shown in eq. 11. According to the objective function of PAPSO which decreases the number of active hosts, it consolidates the migrated VMs into the minimal number of hosts. As a result, it increases the CPU utilization in the running servers, where other hosts can be switched to a sleep mode. Thus, the total energy consumption can be minimized as shown in Figure 6. It can be noticed that the proposed technique can reduce the consumed power with 8.01% on average, in comparison to PABFD. The proposed technique can achieve an acceptable reduction

TABLE 7. Power consumption in (Watt) at different levels of utilization.

Server Type	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
HP ProLiant ML110 G4	86	89.4	92.6	96	99.5	102	106	108	112	114	117
HP ProLiant ML110 G5	93.7	97	101	105	110	116	121	125	129	133	135

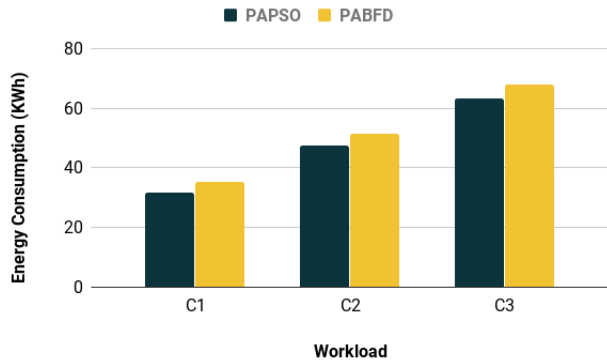


FIGURE 6. Energy consumption.

in the consumed power, taking into consideration that it is constrained on the other hand by not violating SLA.

$$Energy = \int_{t_0}^{t_1} P(u(t))dt \tag{11}$$

**B. NUMBER OF VM MIGRATIONS**

Live VM migration can degrade the performance of the system, thereby violate the SLA. In other words, the more you migrate VMs between hosts, the more you degrade the performance. PAPS0 adopts a technique to decrease the number of active servers, in addition to, minimize the number of overloaded hosts. Therefore, reducing the number of active servers increases CPU utilization of hosts that leads to a minimum number of underloaded servers. On the other hand, it reduces the number of overloaded servers. Thus, the two reasons for migration are taken into consideration which leads to reducing the number of VM migrations as noticed in Figure 7. The proposed technique reduces the number of VM migrations with about 39.65%, compared to PABFD. Although VM consolidation techniques are based on applying many VM migration processes between servers seeking to minimize the total power consumption in data centers, PAPS0 can achieve its aim with a remarkable reduction in the number of VM migrations. This minimization in the amount of VM migration processes has a great effect on the quality of introduced services to users.

**C. SLA VIOLATION**

To ensure an acceptable QoS provided by IaaS environments, SLA should not be violated. Therefore, an independent metric to workloads is introduced to assess SLA that provided to a VM’s user in an IaaS environment [20]. For measuring the level of SLA violation, two metrics are defined based on the following reasons: time during which servers are

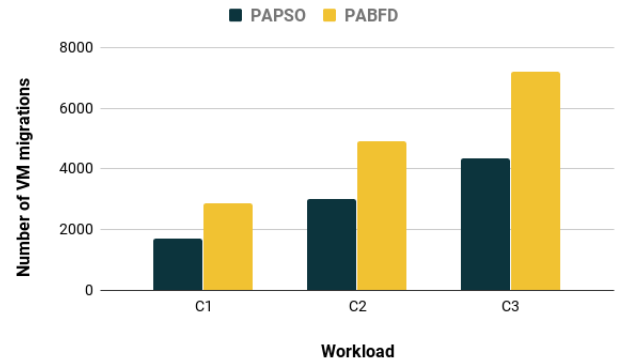


FIGURE 7. Number of VM migrations.

fully utilized and service degradation due to migrations. The first metric is defined as SLA violation Time per Active Host (SLATAH) and can be calculated through eq.12, where N is the total number of servers, T<sub>s<sub>i</sub></sub> is the time during which a server is fully utilized, and T<sub>a<sub>i</sub></sub> is the overall time during which the server state is active. The other metric is defined as Performance Degradation due to Migrations (PDM). It can be measured through eq.13, where M is the number of VMs, C<sub>d<sub>j</sub></sub> is an estimation of the performance degradation of VM<sub>j</sub> due to migration which is estimated by 10% of CPU utilization during all migrations of that VM, and C<sub>r<sub>j</sub></sub> is the total requested CPU capacity of VM<sub>j</sub> during its operation. Consequently, a combined metric called SLA Violation (SLAV) is provided based on these two previous independent metrics and can be calculated through eq.14.

$$SLATAH = \frac{1}{N} \sum_{i=1}^N \frac{T_{s_i}}{T_{a_i}} \tag{12}$$

$$PDM = \frac{1}{M} \sum_{j=1}^M \frac{C_{d_j}}{C_{r_j}} \tag{13}$$

$$SLAV = SLATAH \cdot PDM \tag{14}$$

Experimental results proved that the proposed work does not violate SLA, compared to PABFD as declared in Figure 8. It can be noticed that PAPS0 does not only avoid increasing SLA violation, but it can reduce SLA violation with the increase in the number of VMs and PMs (Such as C3). Although the main goal for this work is to consolidate the migrated VMs into the least number of hosts to save power consumption, it can avoid SLA violation by avoiding its two main sources. Specifically, it makes PMs not overloaded as much as possible and decreases also the number of VM migrations. Therefore, QoS for users can be protected from the side effects of the VM consolidation process.

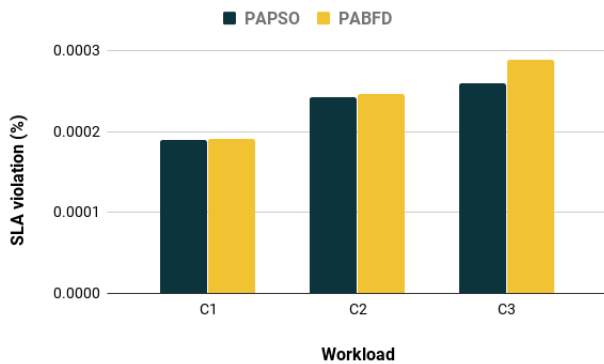


FIGURE 8. SLA violation.

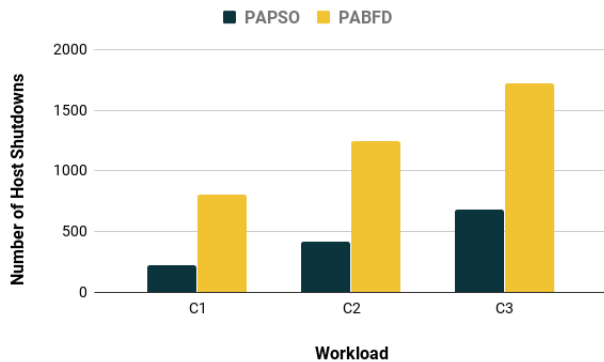


FIGURE 9. Number of host shutdowns.

#### D. NUMBER OF HOST SHUTDOWNS

Another effective factor on QoS is the number of host shutdowns. When CPU utilization in a server becomes under a specific limit, VMs have to be migrated from this server to other servers. Therefore, this underloaded server can be switched off or put in a low power mode. The server can host migrated VMs in the future, and it can be expected to migrate those VMs again if the server is underloaded. Therefore, its state can be change again to the low power mode, to reduce the power consumption. This scenario not only degrades the performance for users by repetitive VM migrations, but also has a bad effect on the power consumption. It is not cost-effective to operate a host with light loads in a repeated fashion. According to avoidance both overloaded and underloaded scenarios in PAPSO, it can decrease the number of host shutdowns as depicted in Figure 9. The proposed technique minimizes the number of host shutdowns with 66.33% on average, in comparison to PABFD.

#### E. ENERGY SLA VIOLATION (ESV)

When studying the VM placement problem, it will be noticed that it is a trade-off problem. Specifically, cloud service providers consolidate VMs into the minimal number of servers, as they care about costs of operation and cooling for the servers. On the other hand, customers focus on the service performance, that should not be affected by the consolidation process. Thus, cloud service providers seek to reduce energy consumption without violating SLA.

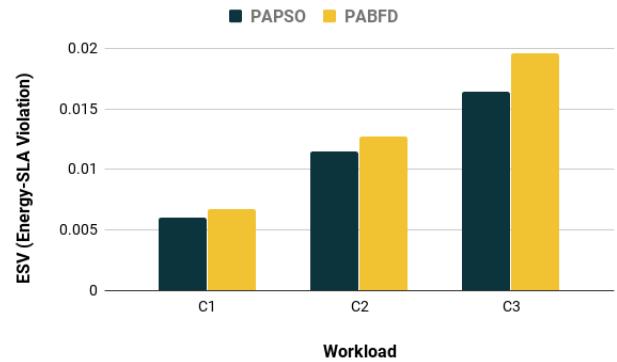


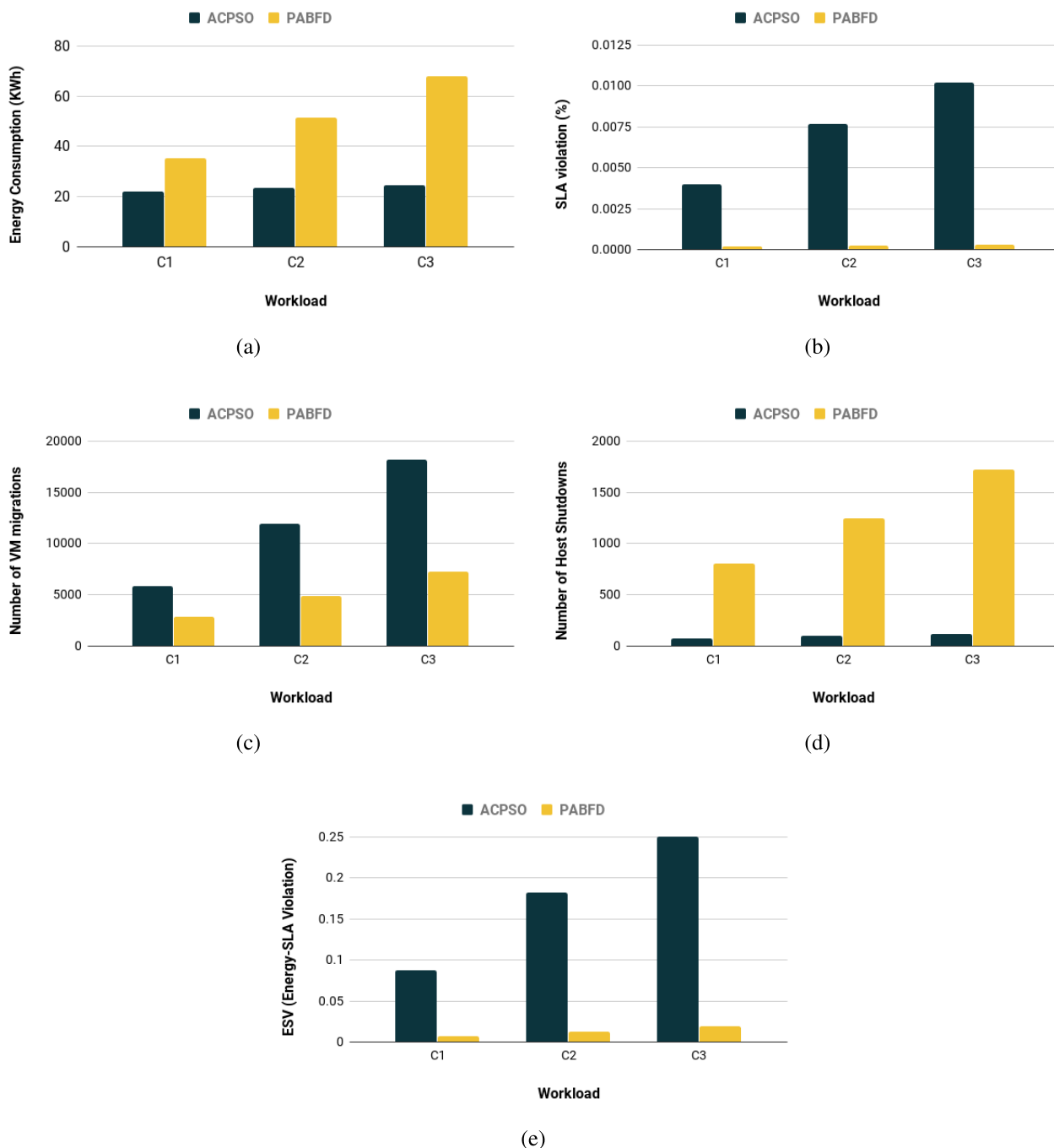
FIGURE 10. The combined metric Energy-SLA violation (ESV).

Therefore, a combined trade-off metric [20], which merges energy consumption and SLA violation, has been introduced to evaluate the VM placement. In our proposed technique, PAPSO can reduce energy consumption without violating SLA. Consequently, PAPSO outperforms PABFD by reducing ESV as declared in Figure 10. The amount of reduction is estimated with 11.87% on average due to taking into consideration the consumed energy in data centers, in addition to, the SLA violation.

#### VIII. DISCUSSION

The proposed technique, PAPSO, has been evaluated through three test cases under random workloads as declared in the previous section. It exploits the capabilities of PSO to search for the best VM placement solution locally and globally. Its objective is to manage power consumption in data centers by introducing an effective power-aware technique to map the migrated VMs to the most appropriate servers. Specifically, it tries to keep servers in the normal mode by reducing the number of overloaded servers and underloaded ones. Overloaded servers have bad effects on the introduced services to users. On the other hand, underloaded servers waste resources and leverage the overall power consumption in data centers, which necessitate to perform a number of VM migrations that in turn violate the SLA. Experimental results, compared to PABFD, have proved that PAPSO can avoid violating SLA by keeping servers away from the overload or underload state. PAPSO can also reduce the consumed power in servers by consolidating VMs into a minimal number of servers. Simulation results have shown that PAPSO can reduce on average about 8.01% of power consumption, 39.65% of the number of VM migrations, and 66.33% of the number of host shutdowns. Finally, the efficiency of PAPSO is clarified through the combined metric Energy SLA Violation (ESV), where it can achieve a reduction of about 11.87% on average.

Although the main objective of PAPSO is to reduce the power consumption, it preserves SLA against violation. To illustrate the efficiency of PAPSO, another scenario is applied using Aggressive Consolidation based on PSO (ACPSO). This scenario adopts the same technique, but without considering the number of overloaded hosts in the fitness function. It focuses only on increasing the utilization



**FIGURE 11.** Comparison between ACPSO and PABFD; (a) Energy consumption, (b) SLA violation, (c) Number of VM migrations, (d) Number of host shutdowns, (e) ESV.

and reducing the number of active hosts. Therefore, it can achieve a better reduction in the consumed power, compared to PAPS0. However, this reduction comes at expense of the performance, thereby violates the SLA completely. Figure 11 shows that ACPSO outperforms PABFD in decreasing energy consumption drastically, but with a heavy cost in the context of SLA violation. Additionally, it shows that ACPSO performs a large number of VM migrations. This is due to the growing number of the overloaded servers that necessitates

to migrate VMs from the overloaded hosts continuously, which in turn violates the SLA. On the other hand, ACPSO reduces the number of host shutdowns dramatically, where this performance metric is related to the number of underloaded servers which disappears almost in ACPSO. As a conclusion to this scenario, results of the trade-off performance metric ESV show that decreasing the power consumption should not be the only objective to have a good VM placement.

## IX. CONCLUSION

Cloud computing and its services have become widespread all over the world, therefore the construction of large-scale data centers has been doubled. Consequently, there was a need to manage power consumption in data centers. Dynamic consolidation of VMs into the minimum number of servers is a significant issue in the context of reducing the power consumption. It deals with migrating VMs from underloaded servers to other hosts, where their original hosts can be switched to a sleep mode for the sake of energy saving. It focuses also on migrating VMs from the overloaded servers to avoid SLA violation. Thus, finding appropriate hosts to receive the migrated VMs is a vital mission. Therefore, we introduced in this paper, PAPS0, a power-aware VM placement technique based on PSO to reduce power consumption without violating SLA. A decimal encoding for PSO was introduced to suit the VM placement problem, where PSO deals with the continuous problems. Moreover, an effective fitness function was employed to reduce the number of active servers and the number of overloaded ones. The proposed technique was implemented in CloudSim and simulation results, compared to PABFD, showed its efficiency in terms of consumed energy, number of host shut-downs, number of VM migrations, and ESV. As future work, PAPS0 efficiency can be tested through implementation in a real environment. Additionally, more optimization can be achieved by taking more resources into consideration such as memory, bandwidth, and the network factors.

## REFERENCES

- [1] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *J. Netw. Comput. Appl.*, vol. 34, no. 1, pp. 1–11, Jan. 2011.
- [2] J. M. Kaplan, W. Forrest, and N. Kindler, "Revolutionizing data center energy efficiency," McKinsey & Company, Chennai, India, Tech. Rep., 2008. [Online]. Available: [https://www.sallan.org/pdfs/docs/McKinsey\\_Data\\_Center\\_Efficiency.pdf](https://www.sallan.org/pdfs/docs/McKinsey_Data_Center_Efficiency.pdf)
- [3] J. Koomey, "Growth in data center electricity use 2005 to 2010," Analytical Press, Oakland, CA, USA, Tech. Rep., 2011, vol. 9, p. 161.
- [4] A. Shehabi, S. J. Smith, D. A. Sartor, R. E. Brown, M. Herrlin, J. G. Koomey, E. R. Masanet, N. Horner, I. L. Azevedo, and W. Lintner. *United States Data Center Energy Usage Report*. Accessed: Apr. 23, 2020. [Online]. Available: <https://eta.lbl.gov/publications/united-states-data-center-energy>
- [5] A. Varasteh and M. Goudarzi, "Server consolidation techniques in virtualized data centers: A survey," *IEEE Syst. J.*, vol. 11, no. 2, pp. 772–783, Jun. 2017.
- [6] M. Uddin, Y. Darabdarabkhani, A. Shah, and J. Memon, "Evaluating power efficient algorithms for efficiency and carbon emissions in cloud data centers: A review," *Renew. Sustain. Energy Rev.*, vol. 51, pp. 1553–1563, Nov. 2015.
- [7] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, Dec. 2007.
- [8] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control," *Cluster Comput.*, vol. 12, no. 1, pp. 1–15, Mar. 2009.
- [9] C.-M. Wu, R.-S. Chang, and H.-Y. Chan, "A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters," *Future Gener. Comput. Syst.*, vol. 37, pp. 141–147, Jul. 2014.
- [10] S. Wang, Z. Qian, J. Yuan, and I. You, "A DVFS based energy-efficient tasks scheduling in a data center," *IEEE Access*, vol. 5, pp. 13090–13102, 2017.
- [11] M. Masdari and H. Khezri, "Efficient VM migrations using forecasting techniques in cloud computing: A comprehensive review," *Cluster Comput.*, pp. 1–30, Jan. 2020, doi: [10.1007/s10586-019-03032-x](https://doi.org/10.1007/s10586-019-03032-x).
- [12] C. Li, D. Feng, Y. Hua, and L. Qin, "Efficient live virtual machine migration for memory write-intensive workloads," *Future Gener. Comput. Syst.*, vol. 95, pp. 126–139, Jun. 2019.
- [13] A. V. Kumar, V. Krishnakumar, and A. N. Kumar, "Efficient performance upsurge in live migration with downturn in the migration time and downtime," *Cluster Comput.*, vol. 22, no. S5, pp. 12737–12747, Sep. 2019.
- [14] M. Noshay, A. Ibrahim, and H. A. Ali, "Optimization of live virtual machine migration in cloud computing: A survey and future directions," *J. Netw. Comput. Appl.*, vol. 110, pp. 1–10, May 2018.
- [15] A. Bhardwaj and C. Rama Krishna, "Improving the performance of pre-copy virtual machine migration technique," in *Proc. 2nd Int. Conf. Commun., Comput. Netw.*, C. R. Krishna, M. Dutta, and R. Kumar, Eds. Singapore: Springer, 2019, pp. 1021–1032.
- [16] P. Barham, "Xen and the art of virtualization," *ACM SIGOPS Oper. Syst. Rev.*, vol. 37, no. 5, pp. 164–177, Dec. 2003.
- [17] A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori, "Kvm: The linux virtual machine monitor," in *Proc. Linux Symp.*, vol. 1, 2007, pp. 225–230. [Online]. Available: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.488.2278>
- [18] *Vmware*. Accessed: Apr. 23, 2020. [Online]. Available: <http://www.vmware.com>
- [19] H. Talebian, A. Gani, M. Sookhak, A. A. Abdelatif, A. Yousafzai, A. V. Vasilakos, and F. R. Yu, "Optimizing virtual machine placement in IaaS data centers: Taxonomy, review and open issues," *Cluster Comput.*, pp. 1–42, Jul. 2019, doi: [10.1007/s10586-019-02954-w](https://doi.org/10.1007/s10586-019-02954-w).
- [20] A. Beloglazov 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.*, vol. 24, no. 13, pp. 1397–1428, Sep. 2012.
- [21] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, vol. 4, Nov. 1995, pp. 1942–1948.
- [22] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proc. IEEE Int. Conf. Syst., Man, Cybernetics. Comput. Cybern. Simul.*, Oct. 1997, pp. 4104–4108.
- [23] M. A. Khan, A. Paplinski, A. M. Khan, M. Murshed, and R. Buyya, *Dynamic Virtual Machine Consolidation Algorithms for Energy-Efficient Cloud Resource Management: A Review*. Cham, Switzerland: Springer, 2018, pp. 135–165.
- [24] R. Nathuji and K. Schwan, "Virtualpower: Coordinated power management in virtualized enterprise systems," *SIGOPS Oper. Syst. Rev.*, vol. 41, no. 6, pp. 265–278, Oct. 2007.
- [25] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Gener. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, May 2012.
- [26] F. Farahnakian, P. Liljeberg, and J. Plosila, "LIRCUP: Linear regression based CPU usage prediction algorithm for live migration of virtual machines in data centers," in *Proc. 39th Euromicro Conf. Softw. Eng. Adv. Appl.*, Sep. 2013, pp. 357–364.
- [27] 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.*, vol. 139, pp. 99–109, May 2020.
- [28] R. Yadav, W. Zhang, O. Kaiwartya, P. R. Singh, I. A. Elgendy, and Y. Tian, "Adaptive energy-aware algorithms for minimizing energy consumption and sla violation in cloud computing," *IEEE Access*, vol. 6, pp. 55923–55936, 2018.
- [29] R. Yadav and W. Zhang, "MeReg: Managing energy-SLA tradeoff for green mobile cloud computing," *Wireless Commun. Mobile Comput.*, vol. 2017, Dec. 2017, Art. no. 6741972.
- [30] R. Nasim, J. Taheri, and A. J. Kessler, "Optimizing virtual machine consolidation in virtualized datacenters using resource sensitivity," in *Proc. IEEE Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Dec. 2016, pp. 168–175.
- [31] A. Marcel, P. Cristian, P. Eugen, P. Claudia, T. Cioara, I. Anghel, and S. Ioan, "Thermal aware workload consolidation in cloud data centers," in *Proc. IEEE 12th Int. Conf. Intell. Comput. Commun. Process. (ICCP)*, Sep. 2016, pp. 377–384.
- [32] F. Farahnakian, R. Bahsoon, P. Liljeberg, and T. Pahikkala, "Self-adaptive resource management system in IaaS clouds," in *Proc. IEEE 9th Int. Conf. Cloud Comput. (CLOUD)*, Jun. 2016, pp. 553–560.



- [33] F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila, N. T. Hieu, and H. Tenhunen, "Energy-aware VM consolidation in cloud data centers using utilization prediction model," *IEEE Trans. Cloud Comput.*, vol. 7, no. 2, pp. 524–536, Apr. 2019.
- [34] A. Abdelsamea, A. A. El-Moursy, E. E. Hemayed, and H. Eldeeb, "Virtual machine consolidation enhancement using hybrid regression algorithms," *Egyptian Informat. J.*, vol. 18, no. 3, pp. 161–170, Nov. 2017.
- [35] L. Xie, S. Chen, W. Shen, and H. Miao, "A novel self-adaptive VM consolidation strategy using dynamic multi-thresholds in IaaS clouds," *Future Internet*, vol. 10, no. 6, p. 52, 2018.
- [36] A. Mosa and R. Sakellariou, "Virtual machine consolidation for cloud data centers using parameter-based adaptive allocation," in *Proc. 5th Eur. Conf. Eng. Computer-Based Syst. - ECBS*, New York, NY, USA, 2017, p. 16.
- [37] D. A. Alboaneen, B. Pranggono, and H. Tianfield, "Energy-aware virtual machine consolidation for cloud data centers," in *Proc. IEEE/ACM 7th Int. Conf. Utility Cloud Comput.*, Washington, DC, USA, Dec. 2014, pp. 1010–1015.
- [38] Y. Chang, C. Gu, F. Luo, G. Fan, and W. Fu, "Energy efficient resource selection and allocation strategy for virtual machine consolidation in cloud datacenters," *IEICE Trans. Inf. Syst.*, vol. E101.D, no. 7, pp. 1816–1827, Jul. 2018.
- [39] R. Yadav, W. Zhang, H. Chen, and T. Guo, "MuMs: Energy-aware VM selection scheme for cloud data center," in *Proc. 28th Int. Workshop Database Expert Syst. Appl. (DEXA)*, Aug. 2017, pp. 132–136.
- [40] R. Yadav, W. Zhang, K. Li, C. Liu, M. Shafiq, and N. K. Karn, "An adaptive heuristic for managing energy consumption and overloaded hosts in a cloud data center," *Wireless Netw.*, vol. 26, no. 3, pp. 1905–1919, Apr. 2020.
- [41] M. A. H. Monil and R. M. Rahman, "VM consolidation approach based on heuristics, fuzzy logic, and migration control," *J. Cloud Comput.*, vol. 5, no. 1, p. 8, Dec. 2016.
- [42] E. Feller, L. Rilling, and C. Morin, "Energy-aware ant colony based workload placement in clouds," in *Proc. IEEE/ACM 12th Int. Conf. Grid Comput.*, Washington, DC, USA, Sep. 2011, pp. 26–33.
- [43] M.-H. Malekloo, N. Kara, and M. El Barachi, "An energy efficient and SLA compliant approach for resource allocation and consolidation in cloud computing environments," *Sustain. Comput., Informat. Syst.*, vol. 17, pp. 9–24, Mar. 2018.
- [44] A. C. Adamuthe, R. M. Pandharpatte, and G. T. Thampi, "Multiobjective virtual machine placement in cloud environment," in *Proc. Int. Conf. Cloud Ubiquitous Comput. Emerg. Technol.*, Nov. 2013, pp. 8–13.
- [45] G. Wu, M. Tang, Y.-C. Tian, and W. Li, "Energy-efficient virtual machine placement in data centers by genetic algorithm," in *Proc. Int. Conf. Neural Inf. Process.* Berlin, Germany: Springer, 2012, pp. 315–323.
- [46] A. Marotta and S. Avallone, "A simulated annealing based approach for power efficient virtual machines consolidation," in *Proc. IEEE 8th Int. Conf. Cloud Comput.*, Jun. 2015, pp. 445–452.
- [47] M. Abdel-Basset, L. Abdle-Fatah, and A. K. Sangaiah, "An improved Lévy based whale optimization algorithm for bandwidth-efficient virtual machine placement in cloud computing environment," *Cluster Comput.*, vol. 22, no. S4, pp. 8319–8334, Jul. 2019.
- [48] D. A. Alboaneen, H. Tianfield, and Y. Zhang, "Glowworm swarm optimisation algorithm for virtual machine placement in cloud computing," in *Proc. Intl IEEE Conferences Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput., Commun., Cloud Big Data Comput., Internet People, Smart World Congr. (UIC/ATC/ScalCom/CBDCCom/IoP/SmartWorld)*, Jul. 2016, pp. 808–814.
- [49] M. H. Fathi and L. M. Khanli, "Consolidating VMs in green cloud computing using harmony search algorithm," in *Proc. Int. Conf. Internet E-Business (ICIEB)*, New York, NY, USA, 2018, pp. 146–151.
- [50] M. Soltanshahi, R. Asemi, and N. Shafiei, "Energy-aware virtual machines allocation by krill herd algorithm in cloud data centers," *Heliyon*, vol. 5, no. 7, Jul. 2019, Art. no. e02066.
- [51] C. C. T. Mark, D. Niyato, and T. Chen-Khong, "Evolutionary optimal virtual machine placement and demand forecaster for cloud computing," in *Proc. IEEE Int. Conf. Adv. Inf. Netw. Appl.*, Mar. 2011, pp. 348–355.
- [52] N. Donyagard Vahed, M. Ghobaei-Arani, and A. Souri, "Multiobjective virtual machine placement mechanisms using nature-inspired Meta-heuristic algorithms in cloud environments: A comprehensive review," *Int. J. Commun. Syst.*, vol. 32, no. 14, Sep. 2019, Art. no. e4068, doi: 10.1002/dac.4068.
- [53] H. Wang and H. Tianfield, "Energy-aware dynamic virtual machine consolidation for cloud datacenters," *IEEE Access*, vol. 6, pp. 15259–15273, 2018.
- [54] M. Mohammadhosseini, A. Toroghi Haghigat, and E. Mahdipour, "An efficient energy-aware method for virtual machine placement in cloud data centers using the cultural algorithm," *J. Supercomput.*, vol. 75, no. 10, pp. 6904–6933, Oct. 2019.
- [55] A. Fatima, N. Javaid, A. Anjum Butt, T. Sultana, W. Hussain, M. Bilal, M. Hashmi, M. Akbar, and M. Ilahi, "An enhanced multi-objective gray wolf optimization for virtual machine placement in cloud data centers," *Electronics*, vol. 8, no. 2, p. 218, 2019.
- [56] S. Wang, Z. Liu, Z. Zheng, Q. Sun, and F. Yang, "Particle swarm optimization for energy-aware virtual machine placement optimization in virtualized data centers," in *Proc. Int. Conf. Parallel Distrib. Syst.*, Dec. 2013, pp. 102–109.
- [57] S. E. Dashti and A. M. Rahmani, "Dynamic VMs placement for energy efficiency by PSO in cloud computing," *J. Experim. Theor. Artif. Intell.*, vol. 28, nos. 1–2, pp. 97–112, Mar. 2016.
- [58] K. Braiki and H. Youssef, "Multi-objective virtual machine placement algorithm based on particle swarm optimization," in *Proc. 14th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2018, pp. 279–284.
- [59] R. Jensi and G. W. Jiji, "An enhanced particle swarm optimization with levy flight for global optimization," *Appl. Soft Comput.*, vol. 43, pp. 248–261, Jun. 2016.
- [60] M. Abdullah, K. Lu, P. Wieder, and R. Yahyapour, "A heuristic-based approach for dynamic VMs consolidation in cloud data centers," *Arabian J. for Sci. Eng.*, vol. 42, no. 8, pp. 3535–3549, Aug. 2017.
- [61] Z. Tang, Y. Mo, K. Li, and K. Li, "Dynamic forecast scheduling algorithm for virtual machine placement in cloud computing environment," *J. Supercomput.*, vol. 70, no. 3, pp. 1279–1296, Dec. 2014.
- [62] J. Yan, H. Zhang, H. Xu, and Z. Zhang, "Discrete pso-based workload optimization in virtual machine placement," *Pers. Ubiquitous Comput.*, vol. 22, no. 3, pp. 589–596, Jun. 2018.
- [63] N. Garg, D. Singh, and M. S. Goraya, "Power and resource-aware VM placement in cloud environment," in *Proc. IEEE 8th Int. Advance Comput. Conf. (IACC)*, Dec. 2018, pp. 113–118.
- [64] S. Lee, R. Panigrahy, V. Prabhakaran, V. Ramasubramanian, K. Talwar, L. Ueda, and U. Wieder, "Validating heuristics for virtual machines consolidation," Microsoft Research, Redmond, WA, USA, Tech. Rep. MSR-TR-2011-9, 2011, pp. 1–14. [Online]. Available: <https://www.microsoft.com/en-us/research/wp-content/uploads/2011/01/virtualization.pdf>
- [65] Z. Á. Mann and M. Szabó, "Which is the best algorithm for virtual machine placement optimization?" *Concurrency Comput., Pract. Exper.*, vol. 29, no. 10, May 2017, Art. no. e4083, doi: 10.1002/cpe.4083.
- [66] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Softw. Pract. Exper.*, vol. 41, no. 1, pp. 23–50, Jan. 2011.
- [67] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *Proc. 34th Annu. Int. Symp. Comput. Archit.*, New York, NY, USA, 2007, pp. 13–23.
- [68] *Power Model of an Hp Proliant ml110 G4 Provided by Specpower Benchmark*. Accessed: Apr. 23, 2020. [Online]. Available: [http://www.spec.org/power\\_ssj2008/results/res2011q1/power\\_ssj2008-20110124-00338.html](http://www.spec.org/power_ssj2008/results/res2011q1/power_ssj2008-20110124-00338.html)
- [69] *Power Model of an Hp Proliant Ml110 G5 Provided by Specpower Benchmark*. Accessed: Apr. 23, 2020. [Online]. Available: [http://www.spec.org/power\\_ssj2008/results/res2011q1/power\\_ssj2008-20110124-00339.html](http://www.spec.org/power_ssj2008/results/res2011q1/power_ssj2008-20110124-00339.html)



**ABDELHAMEED IBRAHIM** received the bachelor's and master's degrees in engineering from the Computer Engineering and Control Systems Department, in 2001 and 2005, respectively, and the Ph.D. degree in engineering from the Faculty of Engineering, Chiba University, Japan, in 2011. He was with the Faculty of Engineering, Mansoura University, from 2001 to 2007. He is currently an Assistant Professor of computer engineering with the Faculty of Engineering, Mansoura University, Egypt. His research interests include computer vision, pattern recognition, optimization, cloud computing, and virtual machine migration. He serves as a Reviewer for the *Journal of Electronic Imaging*, *Optical Engineering*, the *IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS*, *IEEE ACCESS*, *Computer Standards & Interfaces*, the *Journal of Healthcare Engineering*, *IET Image Processing*, *Multimedia Tools and Applications*, and other respected journals.



**MOSTAFA NOSHY** received the B.Sc. degree (Hons.) from Mansoura University, in 2014, and the M.Sc. degree from Mansoura University, in 2020. He is currently a Teaching Assistant with the Computer Engineering and Control Systems Department, Faculty of Engineering, Mansoura University, Egypt. His research interests include cloud computing, virtualization, and live virtual machine migration. He serves as a Reviewer for IEEE ACCESS.



**MAHMOUD BADAWY** received the B.Sc., M.Sc., and Ph.D. degrees from the Computer Engineering and Control Systems Department, Mansoura University, Egypt. He is currently an Associate Professor with the Computer Engineering and Control Systems Department, Faculty of Engineering, Mansoura University. His major research interests are computer networking, distributed control systems, database management systems, mobile robots, and cloud computing.

• • •



**HESHAM ARAFAT ALI** was a Visiting Professor with the Computer Science Department, University of Connecticut, from January 2000 to September 2001. From 2002 to 2004, he was the Vice Dean for student affair with the Faculty of Computer Science and Information, Mansoura University. He is currently a Professor with the Computer Engineering and Control Systems Department, Faculty of Engineering, Mansoura University, Egypt. He has many book chapters

published by international press and about 150 published articles in international (conferences and journals). His interests are in the areas of network security, mobile agent, network management, search engine, pattern recognition, distributed databases, and performance analysis. He is a Founder Member of the IEEE SMC Society Technical Committee on Enterprise Information Systems (EIS). He was awarded with the Highly Commended Award from Emerald Literati Club 2002 for his research on the network security. He has served as a Reviewer for many high quality journals, including the *Journal of Engineering Mansoura University*.