



# Optimal Meta-Heuristic Elastic Scheduling (OMES) for VM selection and migration in cloud computing

Krishan Tuli<sup>1</sup> · Manisha Malhotra<sup>1</sup>

Received: 27 July 2022 / Revised: 17 August 2023 / Accepted: 31 August 2023 /  
Published online: 26 September 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

## Abstract

Virtualization is a powerful technique that allows numerous applications can execute on a single cloud server. The process is carried out by cramming software into Virtual Machines (VMs), so that many programs may execute in parallel which leads to an increase in speed. It reduces the overall cost of the cloud data centers by applying migration, and load balancing techniques on the virtual machines. However, the associated energy consumption and Service Level Agreement (SLA) breaches have been extremely high because of increased network traffic and the bandwidth requirements of the applications. To address this issue, the current study presented a novel approach based on the food selection technique used by honey bees to allocate and utilize resources to the VMs. The proposed Optimal Meta-Heuristic Elastic Scheduling (OMES) integrates the Artificial Bee Colony algorithm with flower pollination to select VMs for specific clusters. The simulation is applied on 1000 VMs and analyzed based on VM migration, energy consumption, and SLA violation performance metrics. The comparative analysis performed against existing studies demonstrates highest unit improvement of 0.47 for VM migrations, 0.485 for power consumption, and 0.305 for SLA-V.

**Keywords** Artificial Bee Colony (ABC) · Cloud computing · Energy consumption · Service Level Agreements (SLAs) · Virtual Machine (VM)

## 1 Introduction

Cloud computing was designed to perform in the most efficient manner any processing engine could have ever imagined to be implemented. Considering the fast-working abilities of the Data-centre (Dc), it was no wonder that complex computation tasks like Google Navigation, fast-speed video editing, and high-resolution image processing of the hyper-structure images became a reality. Considering that one data center may consume power

---

✉ Krishan Tuli  
mca.krishantuli@gmail.com

Manisha Malhotra  
mmanishamalhotra@gmail.com

<sup>1</sup> University Institute of Computing, Chandigarh University, Mohali, Punjab, India

equivalent to the power consumption of 25,000 households, a massive amount of power, green computing has gained popularity among cloud developers and researchers [1]. Hence, modern-day computing environments consider creating a power-balanced processing system rather than creating a system that can speed up the computation. A Dc contains Physical Machines (PMs) used to execute the different tasks provided by users at a specific layer, especially service. The PMs are assisted by Virtual Machines (VMs) at the cost of hardware resources. The VMs uses PM's hardware resources until a VM is associated with the PM. It becomes the responsibility of the scheduler to arrange VMs for the PM to complete a particular set of processing queries. The scheduler sits in between the user layer and the processing.

The Dc takes requests from the user, passes them on to the scheduler with Turn Around Time (Tat), and is expected to complete them on time. Generally, the process is speeded up by arranging the scheduler to VMs for the PMs based on the VM's working capabilities and the user's demand requirements. When a VM is supplied to a PM, the process is referred to as VM allocation. A VM is available over the PM for a specific time interval and then migrated to another PM to support scalability [2]. Overall, the VM to PM association process is a subset of 4 steps.

1. VM is allocated to the desired PM.
2. Identification of the PM to migrate the VMs.
3. Identification of the PM to supply the migrated VMs.
4. Migration of the VM.

The allocation process was introduced in 2010 by Beloglazov and Buyya, considering power consumption as a dependent variable over allocation strategies and policies (Fig. 1). To allocate a VM to a PM, an allocation process was introduced to the world based on the bin packing algorithm by the researchers using the Modified Best Fit Decreasing (MBFD). MBFD identifies every possible PM with enough resources to compensate the considering VM. To allocate, MBFD computes the least likely power consumption and supplies the VM to the most petite consumption holder. The power consumption calculation has been observed to be evaluated by many methods [3].

## 1.1 DVFS (dynamic voltage and frequency scheduling)

The frequency of DVFS strategies changes in response to the use of complex workloads. Since dynamic power is frequency-dependent, these policies help to minimize resource consumption dynamically. Underutilized resource power has been reduced using the DVFS to avoid SLA violations. To avoid performance degradation due to overprovisioning in dynamic workload environments, policies must manage the workload needs and adapt the operating server collection. Such a scheduling technique is the most efficient and energy-viable available today. This technique reduces operating voltage and frequency to scale power according to the device's changing workload. The switching operation is slowed by lowering the operating frequency and voltage, resulting in energy savings while also lowering efficiency. The implementation of DVFS has been done as there is a direct relationship between the CPU power and its frequency, but in the case of using the DVFS for CPU, resources such as voltage and frequency are limited [4].

DVFS has primarily been used to improve the scheduling workload for the CPU' having less energy used for servers. It is also viable for systems where tasks are noncritical (Wu

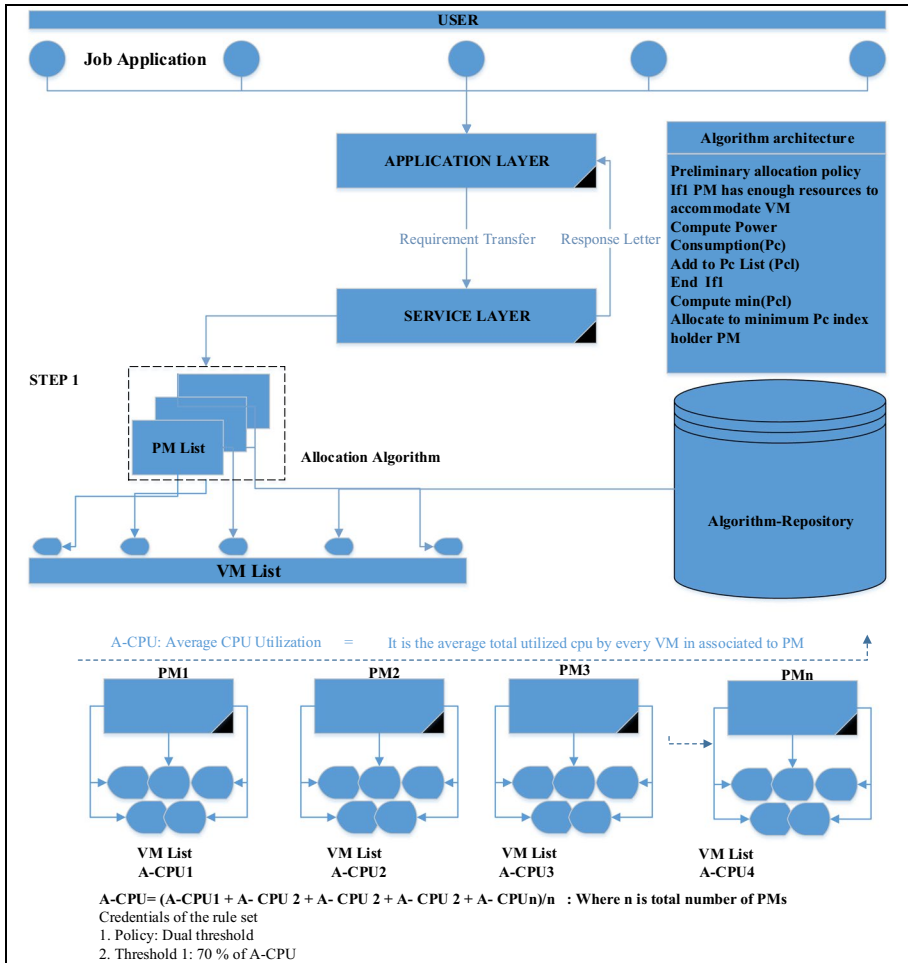


Fig. 1 Service Allocation

et al. 2014). Consequently, reducing voltage and CPU operating frequency slows down the switching operation process, which saves energy but negatively affects the system’s efficiency. As a result, implementing DVFS-aware consolidation policies in Cloud Dc have enough potential and energy to shrink the energy consumed by the machines on which the workload is very high [5].

The term migration comes into action when the data center has to reschedule the VMs to save the PMs from overloading. The most efficient way to judge a PM for the overload parameter is to check the utilization factor and energy consumed by the PM. The cloud does not want to produce the best result. But instead, it would be happy to create an efficient result that can be used to calculate the practical solution to a given problem set. The evaluation policy is a dual-threshold policy and has been opted for by many researchers and practitioners [6].

The famous migrations of VMs are live and non-live migrations. Before the migration process, the VM has to be switched OFF, and the various operating services

provided to the machines have been operated on demand. The knowledge of which VM is running depends upon the energy utilization, and the encapsulation process has been carried out to the target site if it is suspended.

Figure 2 shows that no open network connections are held during the migration, and all connections are rebuilt after the VM is restarted (a). Migration of Memory data and continuity of the network link are the two issues that must be resolved in live migration to prevent service interruption. Data migration is needed when the end-to-end sites do not share the exact storage mechanism. The process operated on the running VMs that are migrated would be disrupted by non-live migration. The most efficient way to judge a machine on the side of utilization is to check the voltage and energy consumed by the PM. Consequently, implementing a live machine reduces operating voltage and frequency to scale power according to the workload in a device. Since several applications in a cloud data center run 24 h a day, this significantly limits its application field. As a result, most research is concentrated on live migration [7].

In contrast, to bundle the VMs into similar group architectures, Nishaat et al. presented the Smart Elastic Scheduling Algorithm (SESA), which intends to bind the VMs as per their location preferences and the RAM associated along with the CPU utilization of the VMs. The work architecture of SESA uses enhanced k-means which are divided into the following significant steps as depicted below: -

1. Calculation of the total number of centroids required to accommodate the migrating VMs.
2. Initial placement of the VMs based on the Euclidean Distance between the attribute set utilized to analyze the VMs for the migration.
3. It is shifting the cluster group based on the changed centroid.

Allocating the VM group with the highest cluster density to the PMs, Where the allocation process is considered only by MBFD.

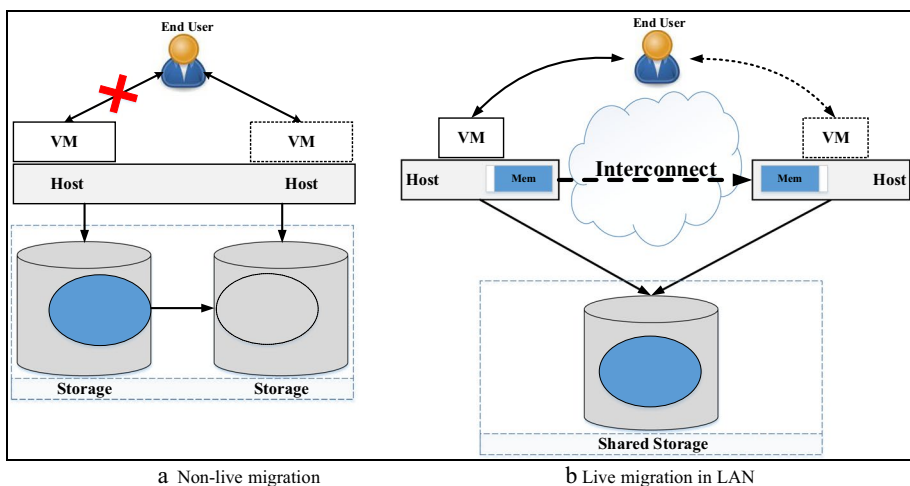


Fig. 2 Migration types

## 1.2 Problem formulation

VM allocation and migration is one of the vital steps that must be implemented to support the elasticity in the network. When the load increases over a PM that becomes relatively high compared to the other PMs in the list, it becomes essential to migrate the VMs from one end to another, viz., from one PM to another PM. Migrating the VMs consumes power from the data center, so choosing the appropriate VM for the migration becomes essential. The proposed VM allocation and migration model is an extended version of the Bin-packing problem, and good conditions have been expressed to avoid inequalities. The main objective of the underlying problem is to determine the number of nodes used to host the VM, and then power consumption and appropriate selection policy have been selected. The requested VMs by the user has been characterized as several servers or Physical Machines (PM) available to provide services in the data center as represented ‘PM.’ The consumption power of the servers is assumed to have a limit as follows:

$$PM_{i,max}, \{i = 1, 2, \dots, m\}$$

At each run, each PM is used to host several VMs and characterized by the current power consumed by the machines as  $PM_{i,current}$ .

Since the main objective of this study is the minimum consumption of power by the data centers and the appropriate selection of VM policy, therefore key decision variables (i) have been defined for each server ( $ES_i$ ) which approaches 1 if server ‘i’ is applicable for VMs, and 0 if not selected. The main function is to place all the VMs within a minimum count of machines or servers expressed as follows:-

$$\min K = \sum_{i=1}^n ES_i \quad (1)$$

The optimization using the efficient technique has been done considering the linear constraints having the capacity limit for the respective server and VM can only be assigned to the server having resources not utilized efficiently. VMs allocation based on the remaining energy usage is illustrated as follows:-

- Each physical machine or server has a certain power limit ( $PM_{i,max}$ ) that is not exceeded at a certain level when hosting a VM and this will be used to determine the remaining capacity of the machine.

$$\sum_{r=1}^j pm_r y_{ri} \leq PM_{i,max} ES_i - PM_{i,current}, \forall i = 1, \dots, m \quad (2)$$

All the requests by the user have been fulfilled by the cloud within an SLA and VM is assigned to the requested one-to-one server:-

$$\sum_{i=1}^m y_{ri} = 1, \forall i = 1, \dots, n \quad (3)$$

For servers, the condition has been verified as

$$(PM_{i,max}) > PM_{i,current}, \text{ and } PM_{i,current} \neq 0,$$

The total count of servers is under the lower bound represented as

$$\text{Lower bound} \rightarrow \left\lceil \frac{\sum_{i=1}^m PM_{i,current}}{PM_{i,max}} \right\rceil \quad (4)$$

Thus, model inequality has been avoided by adding the following equation as follows: -

$$\sum_{i=1}^m ES_i > \left\lceil \frac{\sum_{i=1}^m PM_{i,current}}{PM_{i,max}} \right\rceil \quad (5)$$

Further, the following set of conditions has been considered to determine the objective function as follows: -

$$\min K = \sum_{i=1}^n ES_i \quad (6)$$

This is subjected to the following equations: -

$$\sum_{r=1}^j pm_r y_{ri} \leq PM_{i,max} ES_i - PM_{i,current}, \forall i = 1, \dots, m \quad (7)$$

$$\sum_{i=1}^m y_{ri} = 1, \forall i = 1, \dots, n \quad (8)$$

$$\begin{aligned} \sum_{i=1}^m ES_i &\geq \left\lceil \frac{\sum_{i=1}^m PM_{i,current}}{PM_{i,max}} \right\rceil \\ ES_i &= \begin{cases} 1, & \text{if the server } i \text{ is used} \\ 0, & \text{otherwise} \end{cases} \\ y_{ri} &= \begin{cases} 1, & \text{if the } VM_r \text{ is placed in server } i; \\ 0, & \text{otherwise} \end{cases} \end{aligned} \quad (9)$$

For easy reference, all the reference constants and variables are explained as follows: -

- In the given equation,  $j$  is the request size for requested VMs by the user, and  $n$  is the servers count in the data center.
- $pm_r$  represents the power consumption of  $y_{ri}$
- $y_{ri}$  is a bivalent variable that shows  $VM_r$  is assigned to the server.
- $ES_i$  indicate whether the server 'i' is used or not.
- $PM_{i,max}$  indicates maximum power consumption of server  $i$ .
- $PM_{i,current}$  indicates the current power consumption of server  $i$ .

The problem of the presented work is to reduce energy consumption and optimize the VM selection policy to reduce overall power consumption. A lot of evidence of the usage of SI is presented to select the appropriate VMs in the literature and hence the problem extends to designing a novel fitness function utilizing the SI behavior.

### 1.3 Contribution of the paper

This paper is inspired by the work presented by Nishaat et al. [8]. Though a good enough structure has been presented already, the contribution of the proposed work is listed as follows [8].

1. The proposed work introduces a novel fitness function for adjusting the VM in the current group and evaluating the possibilities of the VM being placed in other clusters.
2. The reallocation policy is based on MBFD as it is in SESA. Still, it gets sustainable improvements in allocating the VM to the PM based on the utilized slots and history of the PM while the VM was allocated to the PM. To utilize the previous experience of the PM after the reallocation of the VM, a monitoring unit is placed, underutilization model.
3. This paper outlines the issues related to VMP and illustrates the techniques for the appropriate placement of the VM.

## 1.4 Organization of the paper

Rest the organization of the paper is structured into 5 sections. Section 2 discusses the distinguishing work done to improve the energy efficiency of cloud data centers based on virtualization. The integration of the Swarm Intelligence (SI) approach is discussed in Section 3 to design a novel fitness function proposed in the paper. Section 4 is dedicated to the simulation analysis and discussion of the results. The derived conclusion and the future work are presented in Section 5. This is followed by the list of references that are cited in the paper.

## 2 Related work

The last decade has witnessed the scientific community being focussed on achieving energy-efficient resource utilization for the cloud environment. Virtualization technology has been one of the most critical aspects of achieving this goal. Therefore, many researchers have also taken advantage of the optimization concept to achieve efficient resource allocation with minimal CPU and power consumption [9].

Various approaches have been proposed in the literature to address the VM migration issues. The metaheuristic techniques based on the intelligent behavior of the species and best fit decreasing algorithms have been developed. The optimization techniques are based on the social behavior of the ants using the Ant Colony Optimization (ACO) approach, and bees, namely Artificial Bee Colony (ABC), particles namely Particle Swarm Optimization (PSO), firefly technique (FFA) also added. The past study also includes ML, deep learning, best fit decreasing (BFD) techniques, round robin, and Modified BFD (MBFD).

A BP-based heuristic is used to investigate a concrete method for controlling the number of VMs migrated [10]. The approach prevents the migration of VMs with consistent workloads, which decreases the number of migrations. Beloglazov et al. [11] propose a threshold-based VM migration optimization after a first fit BP process [11]. A multi-dimensional space partition model coordinates different resource loads [12]. The authors divided the resource space into three categories: acceptance, protection, and forbidden domain. From the former to the latter, VM takes precedence. The given solution will effectively reduce the number of physical machines (PMs) required by maximizing the complementary resource requirement. Song et al. [13] present a robust technique that worked on the migration of VMs. Specifically, VM is categorized into four parts ranging from allocation of small to big. For the one-dimension case, and then packed according to the categories. It has been established that the algorithm is approximate. Only the algorithm can reduce the number of VM migrations batches. However, it cannot guarantee the number of VMs that

will be migrated [13]. A multi-capacity stochastic solution was proposed that was different from the existing multi-dimensional BP because it believed that that particular VM should only use a part of one resource allocated to one VM. Multi-capacity BP more realistically captures server resource characteristics. To reduce the number of stochastic resources used, a heuristic algorithm working on two stages is provided [14]. Heuristics based on BP are often widely used to save resources by reducing the number of services needed.

Similar, complementary, and remaining together are used to achieve a better result in maximizing nodes and bandwidth capital [15]. Shi et al. [16] want to increase CSP sales from an economic standpoint. They organize VM collections based on capacity and placement constraints, such as security, anti-location, and co-location. A hybrid heuristic is proposed based on BP and a spanning tree [16].

Beloglazov et al. [11] proposed an MBFD technique in which the VMs are sorted in decreasing order, and the sorting of PMs was performed according to the power. FFD is used to complete the work after the sorting is completed. The drawback of MBFD is that it only considers a single target and cannot handle scalable data center situations. The heuristic approach, which is based on the greedy algorithm to search and use fewer additional details, is typically used in conventional solution methods [11].

As previously mentioned, BP-based algorithms are unable to profile VM communication. It cannot also represent networked networks. Meta-heuristic (MH) algorithms, apart from the other general-purpose techniques, are worked on by shrinking the solution space linked to the universe population. MH algorithms will achieve efficiency as inspired by nature, as illustrated above. These algorithms' work depends upon the fitness value generated during the complete delivery of food which is directly linked to the maximum output with static input. The species searching the food is linked to the determination of a solution by being kept and carried over to the next.

According to the literature review, a significant number of evolutionary methods have been explored to address the optimization problems of cloud data centers.

## 2.1 Genetic algorithm

It is a classic heuristic algorithm that is inspired by generic evolutionary facts such as crossover, genetic mutation, and other operations. The solution's consistency is assessed using a fitness function.

Mi et al. [17] developed a GA-based technique based on the reallocation of VMs on heterogeneous PMs with evolving workloads. The method was based on using the request forecasting module to analyze the workloads assigned to the specific PMs. The proposed technique is a multi-objective optimization strategy that results in power conservation [17].

In their paper, Xu et al. [18] proposed a two-level control scheme for handling workload and VMPM mappings. This control system architecture was created to reduce overall resource waste and power usage while preventing hotspots. They used an enhanced GA with a fuzzy multi-objective assessment to achieve the goals mentioned earlier. The paper is aimed to use fuzzy logic, and the solution is further accessed using the fitness value. Using an optimization algorithm in conjunction with the GA avoids the problem of random crossover; thus, the scheme for handling workload is viable and provides better results. Fuzzy logic is used to measure the performance of the VM and PM mappings. The proposed algorithm uses less energy, produces less heat, and wastes fewer resources [18].

Wang et al. [19] modeled the VM problem using the elastic strategy. Their approach maximizes resource utilization, balances resource use across dimensions, and reduces



contact traffic. The proposed technique uses the GA to target the problem handled in the first two objectives and was converted to restriction in the last objective. Furthermore, they enhanced the conventional crossover approach by integrating the proposed technique with the GA, and thus the optimal solution was obtained using the fitness value. The solution is feasible for the next generation. Owing to the use of a single objective formula, the suggested solution got better efficiency and the lowest contact with balance resources. The paper is limited to balancing resource use across different dimensions and reducing contact for viable resources. [19].

Wang et al. [20] proposed the two upper-level objective functions that occur in bi-level programming: reducing the gap problem, and it was solved using MOGA-based multi-objective GA. The server's energy usage and efficiency were taken into account. Two upper-level objective functions occur in their bilevel programming: reducing the gap. The proposed solution is also feasible for lower-level programming functions, which are used to minimize the number of differences in server resource usage before and after scheduling [20].

The NSGGA technique was introduced by Liu et al. [21] to solve the VMP problem. It was designed by introducing a novel approach in which the fitness function adjusted of the VM in the current group and the evaluation of the possibilities of the VM to be placed with a contact flow and balancing the resources. The authors used to merge NSGAI's no dominated sorting function with GA's grouping feature [21]. Sofia et al. [22] suggested a method for reducing energy use while maintaining consumer satisfaction. To find a collection of no dominated solutions, they used the DVFS method and the NSGAI optimization method [22].

Riahi and Krichen [23] suggested a VM migration approach using the Bernoulli simulation in conjunction with the GA. Their main goal is to introduce a novel fitness function to adjust the VM to minimize the wastage of resources. GA's fitness function is based on different weights, combining previously described goals into a single plan. The suggested solution has been successfully used to reduce the wastage of resources due to this work [23]. Yousefipour et al. [24] proposed a mathematical model to lower costs and reduce power usage. Then, to solve the issue, they suggested an algorithm that is inspired by the genetic facts as a classic evolutionary heuristic technique that attempts to approximate the optimum solution using a series of operators on the population, such as crossover, mutation, and selection. After that, a GA-based approach was used to solve the formulated problem. The amount of consumed energy at PMs was calculated to compute the consumed energy in cloud data centers [24].

## 2.2 Particle swarm optimization (PSO)

PSO has the advantage of requiring fewer parameters and achieving much quicker convergence. PSO's use of cloud resource provision has been the subject of numerous publications.

Guo et al. [25] proposed a robust solution using the PSO technique and interaction graphs to optimize time and cost. The data placement problem entails mapping all job data to all data centers. They attempted to allocate all task data to the Dc, where data is realized using the graphs and labeled, indicating its significance [25]. Xu et al. [26] proposed IMOPSO, a multi-objective evaluation model for dynamic VM deployment that is used to optimize the time of migration from VM to PM and the optimal use of resources. The proposed work realizes the CloudSim platform and promising results obtained [26].

Wang et al. [27] developed the PSO-based algorithm for the placement of VM. The algorithm is designed for data-intensive workloads in National Cloud Dc (NCDCs). The authors make no assumptions about server compatibility. The MBFD and BF algorithms are compared to the proposed PSO-based algorithm. In tree-like-topology-networked data-centers, the algorithm is energy efficient. One drawback of this research is that it assumes that each VM runs a single operation [27].

To boost energy efficiency, Dashti and Rahmani [28] suggested a solution to allocate migrated VMs from overloaded PMs. They proposed a hierarchical architecture to meet the needs of consumers and providers, using QoS parameters and performance metrics have been measured [28]. Furthermore, Li et al. [29] suggested a solution for managing resource utilization to cause the migration of VMs, reduce data center energy consumption and improve resource utilization. The overloaded and under-loaded PMs are identified by avoiding the remigration process and thus attaining the proposed goal of reducing data center energy consumption and improving resource utilization [29].

### 2.3 Ant Colony Optimization (ACO)

ACO algorithm has gotten much attention from researchers in the last few years for solving various NP-hard problems [43].

ACO-based VM placement algorithm is proposed in [33] to obtain a non-dominated solution set while reducing energy consumption. The results show that VMPACS can scan the solution space more effectively to find a solution that uses the fewest servers and makes the most available resources. As a result of this solution's increased overall performance, less power is used [30]. Ferdous et al. [31] use the ACO metaheuristic to VM to the PM based on the utilized slots and history of the PM while the VM was allocated to the PM. To utilize the previous experience of the PM after the reallocation of the VM, a monitoring unit is placed, underutilization model. This is a practical solution with an excellent computation time [31].

Wen et al. [32] developed an ACO-based distributed VM migration strategy. The goals of the developed solution are for the VM to be allocated to the PM. To utilize the previous experience of the PM after the reallocation of the VM, a monitoring unit is placed under underutilization model usage that is rational, as well as minimizing the number of migrations. They independently track each host's resource use, overcoming the limitations of other ACO approaches' more uncomplicated trigger strategy and pheromone misuse. The authors identified positive and negative pheromones. When a host gets overloaded, all the VMs are sorted according to the average load. The tasks are migrated when the VMs are found with high load values. [32].

Tan et al. [33] suggested a solution ACO-based VM placement algorithm using the threshold policy. They used the CPU use of physical nodes to calculate the SLA violation rate and set the threshold at 90%. They used the weighted coefficient approach to solve this problem, the proposed solution used the weighted function to avoid the M remigration problem, and the MinMax ACO system was used to improve performance. The findings show that the solution has a positive reception since it uses fewer resources than BFD [33].

Malekloo et al. [34] suggested a new method focused on the ACO metaheuristic, which was coupled with a probabilistic decision rule and a heuristic knowledge formula. To find the best solutions, they used the Pareto front process. They compare MOGA to three single-objective algorithms, namely FFD, DVFS, and linear regression, for evaluation (LR). Because of its simplicity, the FFD solution obtained the highest power consumption while

taking the shortest execution time. In comparison to MOGA, MACO consumes less energy and generates better performance. However, neither of these algorithms achieved a lower energy communication rate [34].

A multi-resource overload scheme that is energy conscious has been suggested. ECVMC focused on consolidated placement, is a term coined by Li et al. [35]. Its main goal is to increase physical server resource usage, and thereby technique is energy conscious, and ECVMC realized to consolidate the placement. The simulation process was accurately carried out using the base model, and the performance of the different algorithms was tested considering the other performance metrics. The consolidation process was carried out in different phases, and the migration model was accurately used to minimize the energy of the proposed solution.

Liu et al. [36] suggested an ACO algorithm-based solution in conjunction with the Machine learning algorithm as Extreme Learning Machine (ELM). Because of its success in regression and large dataset classification applications, the ELM is recognized as an emerging learning algorithm. ELM is used to forecast the state of servers, and the process is speeded up by arranging the scheduler to VMs for the PMs based on the VM's working capabilities and the user's demand requirements [36].

## 2.4 Ant Bee Colony (ABC)

The ABC algorithm, which is a swarm-based meta-heuristic algorithm, is inspired by bee actions. It considers potential solutions such as bee food to find a suitable food supply as an optimal solution [42].

$$\sum_{r=1}^j pm_r ABC \leq PM_{i,max} ES_i - PM_{i,current}, \forall i = 1, \dots, m \quad (10)$$

The given equation represents the power utilized during the application of ABC technique in which it is assumed that employed bees ( $Bees_{employed}$ ) having a valid solution with maximum information. However, scout bees still swarming to acquire information about the food source. Such swarming behavior has been linked with the VM allocation. An iteration round has been simulated to determine the server having the least use of resources. Once the requirement is fulfilled then VM is allocated [50].

$$\sum_{i=1}^m Bees_{employed} = 1, \forall i = 1, \dots, n \quad (11)$$

**Table 1** Ordinal Measures employed during the implementation of ABC for the selection of VMs

Maximum Number of Bees related to VMs	10,000
Minimum Number of Bees employed relative to VM's	200
Load supplied for the VMs	$10^6 MIPS$
Total number of simulations per VM set	100
Maximum Number of simulations	$10,000 \times 10,000$
Implementation tool	Python
Platform	Anaconda

The condition is applicable for all VMs and the server allocation process is carried out accordingly. The ordinal measures are illustrated in Table 1.

In the past, several studies presented work on VM allocation and minimum SLA violation.

Jiang et al. [37] presented a model for evaluating energy performance that includes two controlling factors: GPU and CPU utilization rates. They chose the VMs that resulted in the most significant reduction in energy consumption using an ABC-based VM selection algorithm. They proposed a VM migration policy that is live using the ABC, in which the primary goal is to consolidate to perform live VM migration. The outcomes depict that the developed approach efficiently controls the VM migrations while using fewer resources than the current policy. Despite this, its poor scalability and long execution time remain a significant concern. Furthermore, Li et al. [35] formulated that VM migration needs to be considered by demonstrating the various constraints such as energy consumption, violations, CPU utilization (CU), and efficient resource utilization. The study includes the algorithm that finds the best map between the machines by ensuring minimum energy consumption. The results are promising but limited to mapping the VMs as fewer PMs are needed to serve requests. Moreover, the algorithm ensures service quality and maps VMs, reducing energy consumption [51].

## 2.5 Firefly Algorithm (FFA)

The flashing behavior of fireflies inspired the FA, and another MH approach focused on swarm intelligence. It has a wide range of applications in different fields in a short period. The literature has shown to minimize the number of used PMs and achieve an optimum placement solution with the least amount of resource waste and energy use.

Li et al. [38] proposed a viable solution using the FA to place the VM at an appropriate PM. The authors had been used the weighted sum approach to achieve the desired objectives, such as energy consumption and wastage of resources[38]. According to Kansal et al. [39], an FA optimization technique should be used. They try to find the best method to optimize energy efficiency by minimizing migrations [39].

Perumal and Murugaiyan [40] address the VMP by integrating the FFA with the fuzzy technique to solve the VM migration problem. They planned to address two critical issues in cloud data centers: virtual machine allocation and server consolidation. They also attempted to minimize the active PMs and obtain the optimal solution with the least amount of resource waste and energy use. As a result, all of the above priorities were considered simultaneously while determining the best location for the VMs. They set a 90 percent memory and CPU power threshold for each PM to prevent resource consumption from exceeding 100 percent [40].

Yavari et al. [1] developed the VM consolidation technique using the FA, which is temperature aware. The evaluation was done considering the five different metrics: reduction in energy consumption, violations of services, and migrations [1].

## 2.6 Hybrid

The hybrid approach incorporates the metaheuristic techniques described above. Cho et al. [41] proposed an ACOPS for load balancing in the cloud. Its goals are to maximize resource usage balance while receiving as many requests as possible. To schedule, the authors took into account both CPU and memory resources (Table 2). This algorithm

uses an accelerating phase called preselect, which checks each server's remaining memory before scheduling to minimize the solution's overall processing time and dimensions. The algorithm then uses PSO to boost the performance by generating a better solution using the global best solution. Here a fitness solution is used to resolve the local and international challenges. In each iteration round, all the ants find out only the global best to update the best solution's paths. This adds to the solution's effectiveness instead of finding local and international gests. Finally, ACOPS, like other meta-heuristics, is terminated when the iteration exceeds a predetermined number of iterations or the global best solution remains constant across time. The authors cause the time complexity of ACOPS as a supplement to other ACO and PSO algorithms. Furthermore, the results show that the algorithm is good at balancing loads [41, 49].

In summary below (see Table 3), most studies target to minimize energy consumption only in terms of CPU, while storage is an important component of energy consumption in cloud-host machines. Some of the proposed approaches use VM migration mechanisms from overloaded host machines to underloaded hosts. However, these approaches do not define appropriate thresholds (set at runtime) to detect whether host machines are overloaded or underloaded. In addition, existing approaches do not consider the types of applications running on the VMs, which can lead to incorrect migration decisions that result in fewer energy savings and more SLA violations.

### 3 Proposed work algorithm

This paper introduces the novel fitness function to place the VM in the respective clusters, and the function is derived utilizing the attribute architecture of Swarm Intelligence (SI). The usage of SI has been observed in many recent articles [11–14]. The proposed algorithm identified that among most of the research cited articles, colony algorithms have significantly impacted the selection and migration policy. The selection procedure gets incorporated at the IaaS layer when a user submits the request to the server. The server handler passes the requests to the scheduler, and to support the elasticity, the VMs are migrated from one PM to another PM, as shown in the overall system model. SI intelligence architecture is known for its up-bringing solution behavior for small data sets. Using natural computing could also have been another solution, but the computation complexity for a small group of values could be high. This is because the learning abilities of natural computing algorithm depends upon the co-relation for a bulk amount of data. Here, in the proposed solution system, one host could be assigned with few, many, or no VMs. Hence the moderation in the sample population leads us to use SI rather than a natural computing algorithm [52].

The proposed algorithm is based on Artificial Bee Colony (ABC) to continue the legacy. ABC algorithm comprises three bee phases: the employed bee, the onlooker bee, and the scout bee. The proposed algorithm Enhanced Artificial Bee Colony (E-ABC) sequence divided the entire colony into  $k$  number of hives.

The algorithm is executed in the following steps.

- Step 1. Initiate a number of VMs and PMs to start the selection and placement process of VMs.
- Step 2. Apply MBFD algorithm for allocation of VMs.
- Step 3. Based on MBFD outcomes, identify the hotspot VMs.

**Table 2** Comparative analysis of existing work based on nature-inspired optimization algorithms

Paper	Optimization Algorithm	Objective	Optimized Resources
Yousefipour et al. [24]	GA	Power consumption has been reduced with a reduction in active PMs	CPU and memory
Guo et al. [25]	PSO	Time and communication cost has been minimized	CPU, RAM, and bandwidth
Xu et al. [26]	PSO	Maximize the usage of resources and reduction in migration time	CPU, RAM, and bandwidth
Wang et al. [27]	PSO	The power consumed by the machines has been minimized, Maximize the usage of resources and reduction in migration time	CPU, memory, and bandwidth
Dashti and Rahmani [28]	PSO	Resource wastage, VM migration, and power consumption have been minimized	CPU, memory, and bandwidth
Li et al. [29]	PSO	Resource utilization has been improved and power consumption has been minimized	CPU, memory, and bandwidth
Gao et al. [30]	ACO	Improving overall utilization of resources	CPU, Memory, Network I/O
Ferdaus et al. [31]	ACO	Improving energy efficiency, minimum resource wastage	CPU, Network and Storage
Wen et al. [32]	ACO	Minimizing SLA Violations	CPU, memory, and disk
Tan et al. [33]	ACO	Resource utilization has been improved, Improvement in SLA violation and power consumption has been minimized	CPU, memory, and disk
Malekloo et al. [34]	ACO	Maximize the usage of resources and reduction in communication cost	CPU and networking elements
Li et al. [35]	ACO	Optimization, improvement in violations of services, and reduction in energy consumption	CPU
Liu et al. [36]	ACO	Reduction in energy consumption, and improvement in SLA violations	CPU
Jiang et al. [37]	ABC	Reduction in energy consumption, and improvement in SLA violations, migrations, and Resource utilization has been improved	CPU, memory, and bandwidth
Li et al. [35]	ABC	Optimizing energy efficiency and reducing the number of migrations	CPU, memory, and bandwidth
Li et al. [38]	FA	Reduction in power consumption, minimizing the number of VM migrations, and improvement in overload probability	Memory utilization, Bandwidth
Kansal et al. [39]	FA	Reduction in communication cost with time also	CPU, RAM, and bandwidth
Perumal et al. [40]	FA	Reduction in communication cost and power consumption	CPU, memory, and bandwidth
Cho et al. [41]	ACO, PSO	Minimizing the number of migrations	CPU
Sutr et al. [43]	ACO	Minimizing the number of migrations	CPU, power consumption

**Table 2** (continued)

Paper	Optimization Algorithm	Objective	Optimized Resources
Anurag et al. [44]	Crow search Algorithm	VM placement using the live migration	CPU, Memory usage
Verma et al. [45]	Optimization model	Reduction in energy consumption, and improvement in SLA violations	CPU
Talwani et al. [46]	Machine Learning with ABC	Minimizing the number of migrations and Energy Efficiency	CPU
Singh et al. [47]	Whale Optimization Algorithm	Migration Cost, Energy utilized, and Resource Abundance	CPU and RAM
Arshad et al. [48]	–	Reduction in energy consumption, and improvement in SLA violations	CPU

**Table 3** Summary of literature review techniques

Heuristics	Strengths	Weaknesses
Optimal Online Deterministic Algorithms [7]	Minimize the total number of active hosts, live migrations, and performance degradation	Minimize energy base only CPU utilization whereas neglects memory, and disk storage
Dynamic Virtual Machines Consolidation Algorithm [8]	Meet end-to-end performance requirements, energy efficient, cost-effective	CPU, memory and I/O intensive tasks are not considered running on the VM
Energy-Efficient Task Scheduling Algorithm [11]	Improve resource utilization, consider heterogeneous tasks	SLA violations and temperature of host increase
Resource Aware Load Balancing Algorithm [12]	Improve resource utilization, minimal scheduling overhead, reduced makespan	SLA is not considered, not efficient for dependent tasks
Adaptive Energy-Aware Algorithm [13]	Minimize SLA and consider migration overhead	They do not consider the total amount of resource cost
Energy-Efficient, QoS-Aware and Cost-Effective Scheduling [15]	Maintain QoS, maximize resource, energy and cost Savings	Bandwidth and network traffic is also increasing
Energy Efficiency heuristic of VM Migrations [16]	CPU and memory are considered for migration Purpose	Cooling system, network traffic, and migration cost are not considered
Space Aware Best Fit Decreasing [17]	Minimize resource waste, energy optimize	VM placement base on CPU utilization whereas memory is not considered
Energy-Aware Data Centre Resource Allocation [22]	CPU and memory are considered energy efficiency Heuristic	Operational cost increase, neglect the application type running in the VMs
Energy-Efficient and QoS DVM Consolidation [26]	Reduce the amount of VMs migrations, low operating costs, meet SLA	Neglects the essential factors like workload, type of host, and temperature



- Step 4. Aggregate the hotspot VMs from the overutilized PM.
- Step 5. The division of VMs is done into “k” number of clusters based on the architecture utilized in [8]
- Step 6. Initialize ABC algorithm and for each employed bee a onlooker bee is defined.
- Step 7. Call ABC fitness function. If fitness is satisfied, reject the migration otherwise migrate the VM.
- Step 8. Repeat the process until all the VMs are covered.
- Step 9. Create and store the information in a defined repository
- Step 10. Information stored in the repository is further used for reducing the computation complexity of the system.

The High-Level Architecture of Optimal Meta-Heuristic Elastic Scheduling is further illustrated using Fig. 3. The first step is to initialize an empty allocation table and divide the data into “k” number of clusters. As nectar or the food collection behaviour of honey bees is utilized in the proposed work. The information of various hosts for VM’s is updated using flower pollination. An index is selected from flower pollination and VM is migrated to the flower index.

The position of VM is validated and information is stored in a defined repository. Information stored in the repository is further used for reducing the computational complexity of the system. The proposed algorithm uses specifically E-ABC to reflect optimized metaheuristic elastic scheduling. In cloud computing, virtualization is done to manage the underlying resources of the workstation and hosts. The tasks are distributed among the

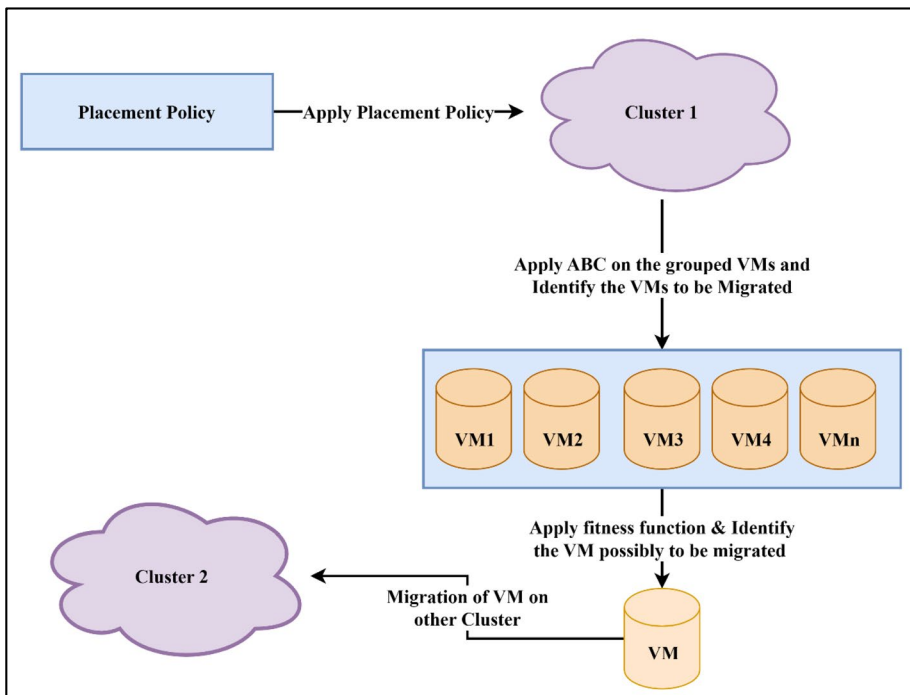


Fig. 3 High Level Architecture of Optimal Meta-Heuristic Elastic Scheduling (OMES) [52]

VMs so as to maintain the optimal performance of the PM by avoiding them getting overloaded. The process starts with the initialization of the number of VMs and PMs. This is followed by the sorting of the VMs using MBFD based on load, service quality etc.

It incorporates the requirements from the VMs and allocates the VM to the best possible PM that allocates the VM on minimum cost. The process holds unique significance as it helps in the identification of the hotspot VMs. The hotspot VMs are then aggregated from the overutilized PM and divided into  $k$  groups. At this stage ABC is applied on each group and for each employed bee, an onlooker bee is defined. In case the bee fitness is satisfied, the VMs are not migrated and stay on the same PM otherwise VM is migrated to other PM. The process of refinement using ABC continues until all the VMs are covered.

The outcomes of the process are further validated for service quality, energy consumption, CPU utilization in the next chapter. The Fig. 4 depicts the flowchart for High Level Architecture of Optimal Meta-Heuristic Elastic Scheduling.

**Algorithm OMES:**

Input: Host List ( $H_L$ ), VM List ( $V_L$ )

Output: Re-allocated List (RaL)

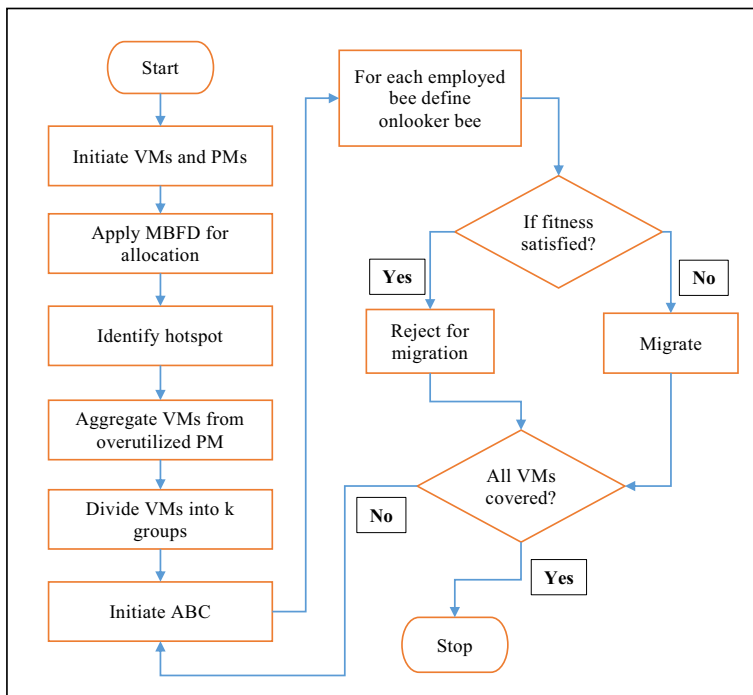
Initialize RaL = [ ]

**Step1:** Arrange Clusters based on the CPU utilization in decreasing order.

**Step2:**  $Tfg = k$ ;

Where Tfg is the total number of fly groups  $k$  is the total number of groups.

**Step3:** For each Tfg in Tfg:



**Fig. 4** High level architecture flowchart of omes algorithm

- a. Arrange VMs in the decreasing order of utilization.
- b. LI required (required flight intensity).

$$F_{structure.append} = fx(wL, tms, PC) \tag{12}$$

$fx$  is the function that aggregates the three components of ABC i.e. the total amount of workload as ‘wL’, total passed memory set instruction as ‘tms’, and total power consumption as ‘PC’.

$F_{structure}$  is the structure of fire-flies of every fly group.

For the normalization of the attributes using Eq. (11)

$$I. f_1^{out} = \{wL_n, tms, PC_n\} \tag{13}$$

$$II. f_2^{out} = Grouped - Host \tag{14}$$

$f_2$  Assembles the host out of  $H_L$  to accommodate all VMs of one group to one host.

/\* Arrangement of Onlooker Hives

**Step 4:** Activate  $f_2$

- a) Gather information for available hosts.
- b) Calculate  $f_{21}$  by applying Eqs. (15), (16), and (17).

$$f_{21} = \frac{\int_{ni=0}^{hL.host\_count} WL_{ni}}{\sum_{j=1}^{hL.host\_count} WL_j} \tag{15}$$

$$f_{22} = \frac{\int_{ni=1}^{hL.host\_count} ts_{ni}}{\sum_{j=1}^{hL.host\_count} tms_{ni}} \tag{16}$$

$$f_{33} = \frac{\int_{ni=0}^{hL.host\_count} PC_{ni}}{\sum_{j=0}^{hL.host\_count} PC_{ni}} \tag{17}$$

The role of  $f_3$  is to segregate the hosts based on the information collected by  $F_{21}$  light intensity.

**Step 5:**  $f_2.input = f_2.Output$

$$f_1.Output = \left\{ \frac{f_{21}}{f_{22} + f_{33}} \right\} \tag{18}$$

The proposed algorithm utilizes the flower Pollination concept and takes a time interval of  $(f_0-f_n)$ .

Where  $n$  is the total number of intervals.

Which gives the actual behavior of the PMs and thieves of  $n$  other light attained intensities which could be from the current intensity values to program to the intensity a random

probability. If the probability is below 5, the fly intensity at the next time slot will be reduced by 10% and vice-versa.

**Step 6:**  $f_{2_{oi}} = \text{Prepare } \{f_{2_{i_0}, \dots, f_{i_n}}\}$

**Step 7:** Total intensity value desired = k.

**Step 8:** Apply the Nashaat rule over  $f_{2_{oi}}$

Where  $f_{2_{oi}}$  is the preliminary output of  $f_2$ .

**Step 9:** For each intensity group calculates the requirement of the group.

$$Min_{PCg} = \max(PC); most_{suitable} = 0$$

**Step 10:** For each hosts group in hL:

**Step 11:** If Costgroup.Satisfied (VM group, CPU demand).

Compute  $PCg = \text{Allocated on group consumption}$ .

Where  $PCg$  power consumption of group to accommodate VMs.

$$\begin{aligned} & \text{if } PCg < Min_{PCg} \\ & \quad Min_{PCg} = PCg \\ & \quad Most_{suitable} \end{aligned}$$

End if

**Step12 (a):** End for

**Step 12 (b):** Embed to RaL.

**Step 13:** Prepare RaL.

**Step 14:** Return RaL.

The proposed algorithm uses specifically Artificial Bee Colony. To validate the results, the evaluations over service parameters have been done and the illustration is given in the next section.

## 4 Results and validations

Based on the analysis, the following parameters have been evaluated.

SLA-V: SLA-V has a lot of ways of measurement. In most of the research articles, the evaluation has been made on the base of power consumption. The proposed algorithm uses Eq. (19) to evaluate the SLA-V.

$$SLA - V = \frac{TPC_{UnderutilizedState} + TPC_{OverutilizedState}}{Total Consumed Power} \tag{19}$$

where,  $TPC_{UnderutilizedState}$  represents Total Power Consumed for underutilized states and  $TPC_{OverutilizedState}$  represents Total Power Consumed for over utilized states.

Migration count: It is the count of total number of migrations that took place in a given time interval.

Total consumed power: It is the total consumption of power in all states.

The illustration of the ordinal measures and the results are given by Tables 4 and 5.

The proposed work shows significant improvement due to the selection of appropriate VMs on the appropriate host by utilizing the idle time of the VMs at the hosts. As a result, more power is preserved as shown in Table 5 when analyzing over 10,000

**Table 4** Ordinal measures

Maximum Number of VMs	10,000
Minimum Number of VMs	200
Supplied Load	10 <sup>6</sup> MIPS
Total number of simulations per VM set	100
Maximum Number of simulations	10,000×100
Implementation tool	Python
Toolset	Spyder
Platform	Anaconda

VMs. The migration analysis shows that the average VM migrations observed for Nashaat et al. are 2802.18 which is 2123.18 using Masdari and Khezri. In comparison to these two works, the OMES achieved only 1637.41 migrations. This exhibits an average improvement of 1164.76 and 485.76 observed for OMES concerning Nashaat et al. and Masdari and Khezri, respectively. The unit improvement concerning the increase in the number of VMs is further illustrated in Fig. 5. It is observed that the unit improvement in terms of VM migrations lies between 0.3838 and 0.4766 with an average unit improvement of 0.4275.

As the number of VMs increases, there is a considerable increase in power consumption using existing works or the OMES. However, it is observed that the average power consumption of OMES is only 286.62 in comparison to 512.20 using Nashaat et al. and 378.29 using Masdari and Khezri. In other words, this shows that the OMES exhibits 225.57 and 910.66 units less power consumption than Nashaat et al. and Masdari and Khezri, respectively. The unit-wise improvement observed against two existing works is illustrated in Fig. 6, which shows that the unit improvement for power consumption ranges from 0.376 to 0.485 for Nashaat et al. and from 0.202 to 0.294 for Masdari and Khezri's work. The average unit improvement demonstrated by OMES against Nashaat et al. and Masdari and Khezri is 0.435 and 0.243, respectively.

The quality of service delivered on the client side is determined by the number of SLA-V. The SLA-V analysis of the OMES shows that it exhibits the least SLA-V with an average value of 0.7011 followed by Masdari and Khezri of 0.759 and Nashaat et al. of 0.769. This shows that OMES exhibits 0.058 and 0.068 less SLA-V as compared to the average SLA-V of Masdari and Khezri and Nashaat et al., respectively when analyzed over 10,000 VMs.

The unit improvement in SLA-V observed for OMES against each variation in the number of VMs used for the experimentation is shown in Fig. 7. It is observed that the unit improvement lies between 0.0025 and 0.3059 when compared to Nashaat et al. and between 0.0072 and 0.234 when compared to Masdari and Khezri. In other words, OMES exhibits an average unit improvement of 0.091 and 0.792 units against Nashaat et al. and Masdari and Khezri, respectively. The detailed simulation analysis of three critical criteria shows that the OMES successfully reduced the number of VM migrations even with an increase in the number of VMs in each step. Moreover, the comparative feature analysis shows that the proposed work exhibited maximum improvement against Nashaat et al.. Therefore, a glimpse of the feature analysis of the proposed work performed against Nashaat et al. work is also summarised in Table 6.

**Table 5** Result illustration

Total Number of VMs	Migrations Nashaat et al. [8]	Migrations Masdari and Khezri [6]	Migrations OMES [8]	Power Consumption Nashaat et al. [8]	Power Consumption Masdari and Khezri [6]	Power Consumption OMES	SLA-V Nashaat et al. [8]	SLA-V Masdari and Khezri [6]	SLA-V OMES
200	157	120	85	27	21.195	16.2	0.692	0.628	0.480
400	283	206	149	54	38.205	27.81	0.659	0.670	0.529
600	469	374	289	81	63.315	50.49	0.779	0.790	0.665
800	609	473	373	108	82.215	63.855	0.780	0.745	0.700
1000	757	554	425	135	102.195	74.79	0.759	0.726	0.687
1500	1144	882	629	202.5	154.44	119.07	0.773	0.808	0.670
2000	1484	1116	789	270	200.34	150.66	0.776	0.740	0.689
2500	1884	1345	986	337.5	254.34	181.575	0.773	0.740	0.734
3000	2246	1673	1178	405	303.21	225.855	0.797	0.750	0.676
3500	2615	2014	1586	472.5	353.025	271.89	0.802	0.767	0.744
4000	2854	2228	1725	540	385.29	300.78	0.758	0.770	0.731
5000	3721	2627	2017	675	502.335	354.645	0.781	0.775	0.745
6000	4681	3631	2815	810	631.935	490.185	0.777	0.808	0.762
7000	5107	3865	3066	945	689.445	521.775	0.791	0.784	0.766
8000	5616	4266	3306	1080	758.16	575.91	0.797	0.772	0.767
9000	6980	5254	4183	1215	942.3	709.29	0.773	0.785	0.771
10,000	7030	5466	4235	1350	949.05	737.91	0.810	0.846	0.795

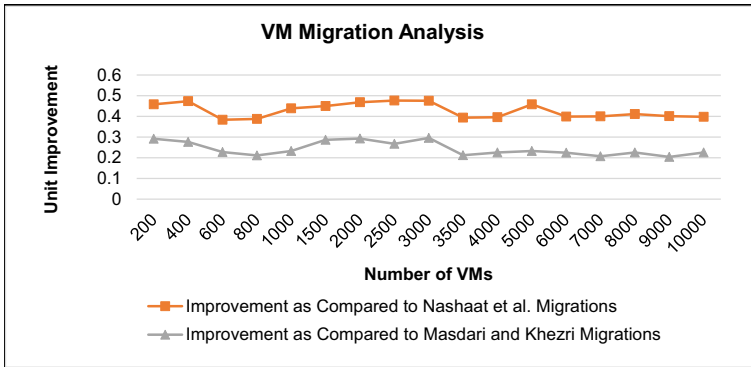


Fig. 5 Unit improvements for migrations

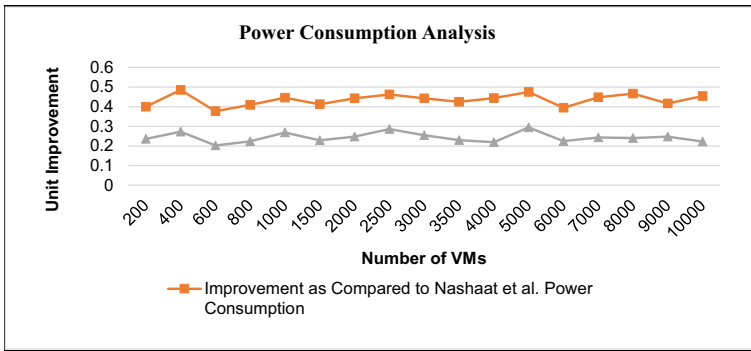


Fig. 6 Unit improvements for power consumption

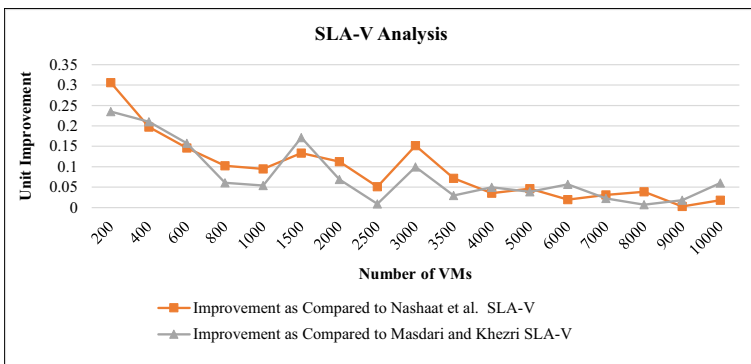


Fig. 7 Unit improvements for SLA-V

Further, it is observed that in comparison to the existing studies, OMES exhibits the least power consumption and SLA-Vs. when analyzed using 10,000 VMs. Table 7 illustrates the comparative analysis of CPU utilization by different techniques. The proposed technique shows the least CPU utilization.

**Table 6** Comparative feature analysis of the proposed work against existing work

Feature	Proposed OMES	Nashaat et al
CPU Utilization	√	√
RAM	√	√
Processing Algorithm (k-means)	√	√
SI Algorithm	ABC	√

**Table 7** Comparative Analysis of CPU utilization

Total Number of VMs	ABC + BA [42]	ACO [43]	CSA[44]	DC-MFA [45]	CPU-OMES
10	935	1323	1254	886.2	884.1
20	1679	1535	1203	1228	1179
30	1599	1095	1254	965	897
40	1599	1614	1502	1135	1132
50	985	1311	1140	1390	1299

## 5 Conclusion

The Swarm Intelligence (SI) is aimed to find the best optimal solution in terms of VMs to minimize the need for a large number of migrations. In the paper, Enhance ABC is used as an SI technique that guides the selection of VM using the ABC fitness function and concept of flower pollination. The simulation studies were performed using 10,000 VMs evaluated in terms of SLA violations, energy consumption, and several VM migrations. The extensive experimentation and analysis show that OMES requires reduced VM migrations that conserve unnecessary power consumption and exhibit the least SLA-Vs. It is concluded that OMES indicates the most considerable unit improvement of 0.476 in terms of SLA-V, 0.485 for power consumption, CPU utilization least, and 0.305 for VM migrations compared with two existing studies. In the future, the author looks forward to integrating other SI techniques to design a hybrid optimization architecture to further reduce energy consumption by improving the VM selection process and minimizing VM migrations.

**Authors' contributions** The author declares that both the authors have same contribution.

**Data availability** No Datasets are used in this research.

## Declarations

**Ethical approval** The author declares that manuscript in part or in full has not been submitted or published anywhere.

**Competing interests** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Conflict of interest** The authors declare that they have no conflict of interest.

Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.



## References

1. Yavari M, Rahbar AG, Fathi MH (2019) Temperature and energy-aware consolidation algorithms in cloud computing. *J Cloud Comput* 8(1):1–16
2. Zhang P, Zhou M, Wang X (2020) An intelligent optimization method for optimal virtual machine allocation in cloud data centers. *IEEE Trans Autom Sci Eng* 17(4):1725–1735
3. Beloglazov A, Buyya R (2010) Energy efficient resource management in virtualized cloud data centers. In: 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing. IEEE, pp 826–831
4. Le Sueur E, Heiser G (2010) Dynamic voltage and frequency scaling: the laws of diminishing returns. In: Proceedings of the 2010 international conference on power aware computing and systems, pp 1–8
5. Arroba P, Moya JM, Ayala JL, Buyya R (2017) Dynamic voltage and frequency scaling-aware dynamic consolidation of virtual machines for energy efficient cloud data centers. *Concurr Comput* 29(10):e4067
6. Masdari M, Khezri H (2020) Efficient VM migrations using forecasting techniques in cloud computing: a comprehensive review. *Clust Comput* 23(4):2629–2658
7. Zhang F, Liu G, Fu X, Yahyapour R (2018) A survey on virtual machine migration: challenges, techniques, and open issues. *IEEE Commun Surv Tutor* 20(2):1206–1243
8. Nashaat H, Ashry N, Rizk R (2019) Smart elastic scheduling algorithm for virtual machine migration in cloud computing. *J Supercomput* 75(7):3842–3865
9. Zhang J, Huang H, Wang X (2016) Resource provision algorithms in cloud computing: a survey. *J Netw Comput Appl* 64:23–42
10. Ferreto TC, Netto MA, Calheiros RN, De Rose CA (2011) Server consolidation with migration control for virtualized data centers. *Futur Gener Comput Syst* 27(8):1027–1034
11. Beloglazov A, Buyya R (2012) Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints. *IEEE Trans Parallel Distrib Syst* 24(7):1366–1379
12. Li X, Qian Z, Lu S, Wu J (2013) Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in a data center. *Math Comput Model* 58(5–6):1222–1235
13. Song W, Xiao Z, Chen Q, Luo H (2013) Adaptive resource provisioning for the cloud using online bin packing. *IEEE Trans Comput* 63(11):2647–2660
14. Hwang I, Pedram M (2013) Hierarchical virtual machine consolidation in a cloud computing system. In: 2013 IEEE sixth international conference on cloud computing. IEEE, pp 196–203
15. Zhang J, He Z, Huang H, Wang X, Gu C, Zhang L (2014) SLA aware cost efficient virtual machines placement in cloud computing. In: 2014 IEEE 33rd International Performance Computing and Communications Conference (IPCCC). IEEE, pp 1–8
16. Shi L, Butler B, Botvich D, Jennings B (2013) Provisioning of requests for virtual machine sets with placement constraints in IaaS clouds. In: 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013). IEEE, pp 499–505
17. Mi H, Wang H, Yin G, Zhou Y, Shi D, Yuan L (2010) Online self-reconfiguration with performance guarantee for energy-efficient large-scale cloud computing data centers. In: 2010 IEEE International Conference on Services Computing. IEEE, pp 514–521
18. Xu J, Fortes JA (2010) Multi-objective virtual machine placement in virtualized data center environments. In: 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing. IEEE, pp 179–188
19. Wang S, Gu H, Wu G (2013) A new approach to multi-objective virtual machine placement in virtualized data center. In: 2013 IEEE Eighth International Conference on Networking, Architecture and Storage. IEEE, pp 331–335
20. Wang X, Wang Y, Cui Y (2014) A new multi-objective bi-level programming model for energy and locality aware multi-job scheduling in cloud computing. *Futur Gener Comput Syst* 36:91–101
21. Liu C, Shen C, Li S, Wang S (2014) A new evolutionary multi-objective algorithm to virtual machine placement in virtualized data center. In: 2014 IEEE 5th International Conference on Software Engineering and Service Science. IEEE, pp 272–275
22. Sofia AS, GaneshKumar P (2018) Multi-objective task scheduling to minimize energy consumption and makespan of cloud computing using NSGA-II. *J Netw Syst Manag* 26(2):463–485
23. Riahi M, Krichen S (2018) A multi-objective decision support framework for virtual machine placement in cloud data centers: a real case study. *J Supercomput* 74(7):2984–3015
24. Yousefipour A, Rahmani AM, Jahanshahi M (2018) Energy and cost-aware virtual machine consolidation in cloud computing. *Softw: Pract Exp* 48(10):1758–1774

25. Guo L, He Z, Zhao S, Zhang N, Wang J, Jiang C (2012) Multi-objective optimization for data placement strategy in cloud computing. In: International conference on information computing and applications. Springer, Berlin, pp 119–126
26. Xu B, Peng Z, Xiao F, Gates AM, Yu JP (2015) Dynamic deployment of virtual machines in cloud computing using multi-objective optimization. *Soft Comput* 19(8):2265–2273
27. Wang S, Zhou A, Hsu CH, Xiao X, Yang F (2015) Provision of data-intensive services through energy-and QoS-aware virtual machine placement in national cloud data centers. *IEEE Trans Emerg Top Comput* 4(2):290–300
28. Dashti SE, Rahmani AM (2016) Dynamic VMs placement for energy efficiency by PSO in cloud computing. *J Exp Theor Artif Intell* 28(1–2):97–112
29. Li H, Zhu G, Cui C, Tang H, Dou Y, He C (2016) Energy-efficient migration and consolidation algorithm of virtual machines in data centers for cloud computing. *Computing* 98(3):303–317
30. Gao Y, Guan H, Qi Z, Hou Y, Liu L (2013) A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. *J Comput Syst Sci* 79(8):1230–1242
31. Ferdaus MH, Murshed M, Calheiros RN, Buyya R (2014) Virtual machine consolidation in cloud data centers using ACO metaheuristic. In: European conference on parallel processing. Springer, Cham, pp 306–317
32. Wen WT, Wang CD, Wu DS, Xie YY (2015) An ACO-based scheduling strategy on load balancing in cloud computing environment. In: 2015 Ninth international conference on frontier of computer science and technology. IEEE, pp 364–369
33. Tan M, Chi C, Zhang J, Zhao S, Li G, Lü S (2017) An energy-aware virtual machine placement algorithm in cloud data center. In: Proceedings of the 2nd international conference on intelligent information processing, pp 1–9
34. Malekloo MH, Kara N, El Barachi M (2018) An energy efficient and SLA compliant approach for resource allocation and consolidation in cloud computing environments. *Sustain Comput: Inform Syst* 17:9–24
35. Li Z, Yan C, Yu L, Yu X (2018) Energy-aware and multi-resource overload probability constraint-based virtual machine dynamic consolidation method. *Futur Gener Comput Syst* 80:139–156
36. Liu F, Ma Z, Wang B, Lin W (2019) A virtual machine consolidation algorithm based on ant colony system and extreme learning machine for cloud data center. *IEEE Access* 8:53–67
37. Jiang J, Feng Y, Zhao J, Li K (2017) DataABC: a fast ABC based energy-efficient live VM consolidation policy with data-intensive energy evaluation model. *Futur Gener Comput Syst* 74:132–141
38. Li XK, Gu CH, Yang ZP, Chang YH (2015) Virtual machine placement strategy based on discrete firefly algorithm in cloud environments. In: 2015 12th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP). IEEE, pp 61–66
39. Kansal NJ, Chana I (2016) Energy-aware virtual machine migration for cloud computing—a firefly optimization approach. *J Grid Comput* 14(2):327–345
40. Perumal B, Murugaiyan A (2016) A firefly colony and its fuzzy approach for server consolidation and virtual machine placement in cloud datacenters. *Adv Fuzzy Syst* 2016
41. Cho KM, Tsai PW, Tsai CW, Yang CS (2015) A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing. *Neural Comput Appl* 26(6):1297–1309
42. Karthikeyan K, Sunder R, Shankar K, Lakshmanaprabu SK, Vijayakumar V, Elhoseny M, Manogaran G (2018) Energy consumption analysis of VM migration in cloud using hybrid swarm optimization (ABC–BA). *J Supercomput* 76(5):3374–3390
43. Sutar SG, Mali PJ, More AY (2020) Resource utilization enhancement through live VM migration in cloud using ant colony optimization algorithm. *Int J Speech Technol* 23:79–85
44. Satpathy A, Addya SK, Turuk AK, Majhi B, Sahoo G (2018) Crow search based virtual machine placement strategy in cloud data centers with live migration. *Comput Electric Eng* 69:334–350
45. Verma G (2022) Secure VM migration in cloud: multi-criteria perspective with improved optimization model. *Wirel Pers Commun* 1–28
46. Talwani S, Alhazmi K, Singla J, Alyamani HJ, Bashir AK (2022) Allocation and migration of virtual machines using machine learning. *Comput Mater Contin* 70(2):3349–3364
47. Singh S, Singh D (2023) A bio-inspired vm migration using re-initialization and decomposition based-whale optimization. *ICT Express* 9(1):92–99
48. Arshad U, Aleem M, Srivastava G, Lin JCW (2022) Utilizing power consumption and SLA violations using dynamic VM consolidation in cloud data centers. *Renew Sustain Energy Rev* 167:112782
49. Singh G, Malhotra M, Sharma A (2022) An adaptive mechanism for virtual machine migration in the cloud environment. *Int J Cloud Appl Comput* 12(1):1–10

50. Barthwal V, Rauthan MS, Varma R (2022) SMA-LinR: an energy and SLA-aware autonomous management of virtual machines. *Int J Cloud Appl Comput* 12(1):1–24
51. Kumar S, Singh SK, Aggarwal N, Gupta BB, Alhalabi W, Band SS (2022) An efficient hardware supported and parallelization architecture for intelligent systems to overcome speculative overheads. *Int J Intell Syst* 37(12):11764–11790
52. Tuli K, Malhotra M (2023) Novel framework: meta-heuristic elastic scheduling approach in virtual machine selection & migration. *Int J Eng Trends Technol* 71(4):436–452

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.