International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V71I4P237
Volume 71 Issue 4, 436-452, April 2023

Original Article

Novel Framework: Meta-Heuristic Elastic Scheduling Approach in Virtual Machine Selection & Migration

K. Tuli[1], M. Malhotra[2]

[1,2]*University Institute of Computing, Chandigarh University, Mohali, Punjab, India.*

[1]*Corresponding Author : mca.krishantuli@gmail.com*

Received: 16 January 2023 Revised: 31 March 2023 Accepted: 13 April 2023 Published: 25 April 2023

Abstract - *Virtualisation is a powerful technique allowing numerous applications to 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, leading to an increase in speed. It reduces the overall cost of cloud data centers by applying migration and load-balancing techniques on 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 utilise 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 analysed based on VM migration, energy consumption, and SLA violation performance metrics. The comparative analysis performed against existing studies demonstrates the 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 equivalent to the power consumption of 25000 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.

- VM is allocated to the desired PM.
- Identification of the PM to migrate the VMs.
- Identification of the PM to supply the migrated VMs.
- 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. An allocation process was introduced to the world based to allocate a VM to a PM 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].

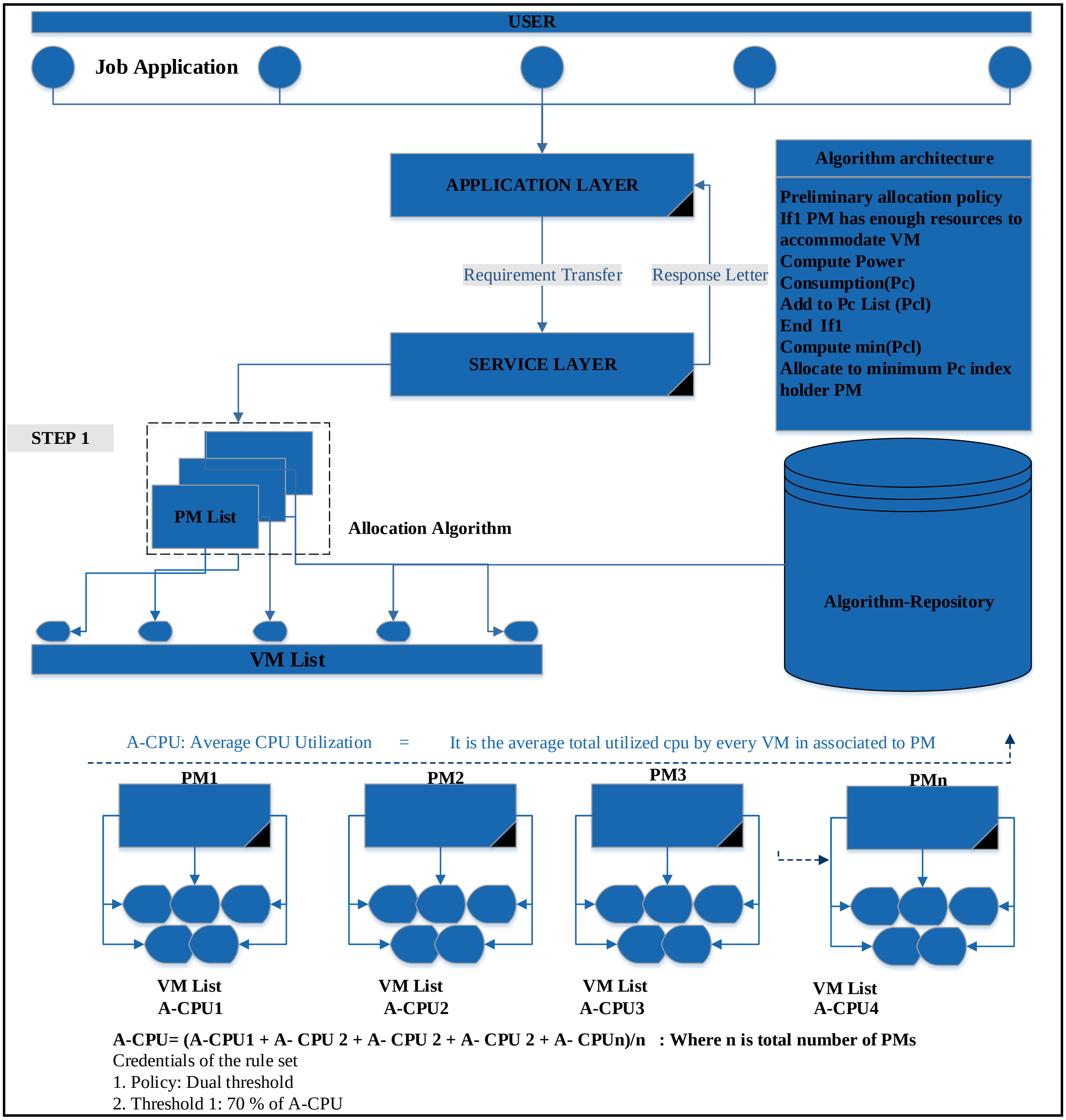
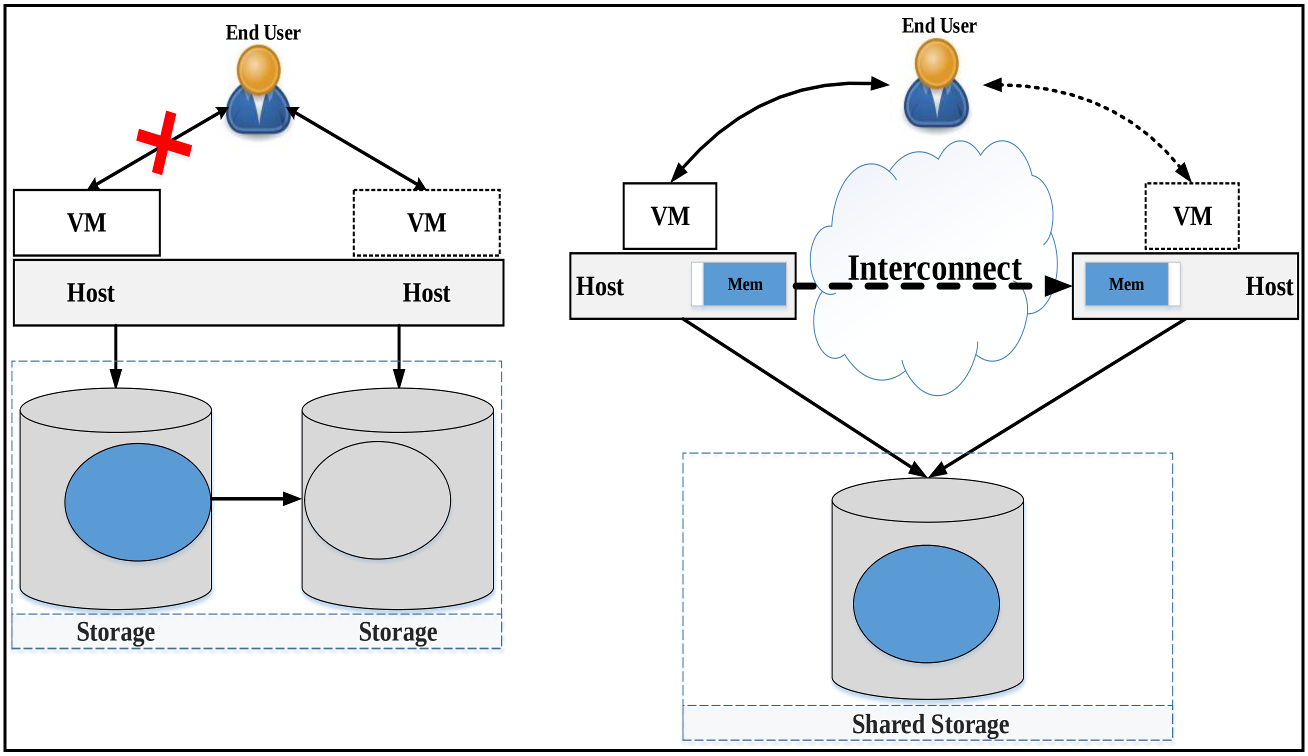


Fig. 1 Service Allocation

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 minimise resource consumption dynamically. Underutilised 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 CPU power and its frequency, but in the case of using DVFS for CPU, resources such as voltage and frequency are limited [4].



a. Non-live migration **b. Live migration in LAN**

Fig. 2 Migration types

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 et al. 2014). Consequently, reducing voltage and CPU operating frequency slows the switching operation process, saving energy but negatively affecting the system's efficiency. As a result, implementing DVFS-aware consolidation policies in Cloud Dc has 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 utilisation 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 dual-threshold evaluation policy 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 utilisation, and the encapsulation process has been carried out to the target site if it is suspended.

Fig. 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 utilisation is to check the voltage and energy consumed by the PM.

Consequently, implementing a live machine reduces the operating voltage and frequency to scale power according to the workload in a device. Since several applications in a cloud data center run 24 hours a day, this significantly limits its application field. As a result, most research is concentrated on live migration [48].

In contrast, to bundle the VMs into similar group architectures, Nishaat et al.[8] presented the Smart Elastic Scheduling Algorithm (SESA), which intends to bind the VMs as per their location preferences and the RAM associated with the CPU utilisation 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 utilised to analyse 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.

1.2. Research Gaps

The VM allocation and migration are crucial in terms of energy consumption, execution time and meeting the service quality of the users. In existing work, VM selection policies were not implemented effectively. False allocation and reallocation are raised, directly impacting resource utilisation and computational time Nashaat et al. [8]. Hence, energy consumption is reduced by compromising the quality services Masdari and Khezri [6]. An effective strategy is required to fill the gap that addresses issues like energy consumption, migration minimisation and SLA violation to provide enhanced service quality to the user end.

1.3. Problem Formulation

VM allocation and migration are vital steps that must be implemented to support the elasticity in the network. In over-utilised conditions, when load increases over a relatively high PM than 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. Sometimes the problem of false allocation also arises, and reallocation is initiated, which consumes more energy and time; hence an intelligent system is required for choosing the appropriate VM for the migration becomes essential. To address this issue, an efficient strategy is required to minimise energy consumption and optimally use the VM selection policy to reduce overall power consumption by considering the SLA constraints and providing better quality services to existing and new users. A lot of evidence for using SI is presented to select the appropriate VMs in the literature. Hence, the problem extends to designing a novel fitness function utilising the SI behavior with the proposed approach.

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 characterised 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 and is illustrated as $PM_{i,max,}\{i = 1,2, \ldots\ldots m\}$. At each run, each PM is used to host several VMs and characterised 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 optimisation 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 utilised 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.

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

The cloud has fulfilled all requests by the user within an SLA, and VM is assigned to the requested one-to-one server:-

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

For servers, the condition has been verified as $\left(PM_{i,max}\right) > PM_{i,current}$), and $PM_{i,current} \ne 0$,
The total count of servers is under the lower bound represented as

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

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:

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

This is subjected to the following equations: -

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

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

$$\sum_{i=1}^{m} ES_i \geq \left\lceil \frac{\sum_{i=1}^{m} PM_{i,current}}{PM_{i,max}} \right\rceil \quad (9)$$

Where, $ES_i = \begin{cases} 1, if\ the\ server\ i\ is\ used \\ 0, otherwise \end{cases}$ while the other notation $y_{ri} = \begin{cases} 1, if\ the\ VM_r\ is\ placed\ in\ server\ i; \\ 0, otherwise \end{cases}$.

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.

1.4. 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 utilised slots and history of the PM while the VM was allocated to the PM. To utilise the previous PM experience after the VM reallocation, a monitoring unit is placed, an underutilisation model.
3. This paper outlines the issues related to VMP and illustrates the techniques for appropriate VM placement.

1.5. Organisation of the Paper

Rest the organisation 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 virtualisation. 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 cited in the paper.

2. Related Work

In the recent decade, researchers have concentrated on making the cloud environment more efficient at optimally using its resources and minimising energy consumption. Virtualisation technologies have been an essential part of reaching this objective. Thus, many academics have used the optimisation approach to allocate resources effectively while minimising the use of computational resources and energy [9]. Various approaches have been discussed in the literature to address VM migration issues. The metaheuristic techniques based on the intelligent behavior of the species and best fit decreasing algorithms have been developed. The optimisation techniques are based on the social behavior of the ants, bees, particles and flies using "Ant Colony Optimization" (ACO), "Artificial Bee Colony" (ABC), "Particle Swarm Optimisation" (PSO), and "Firefly algorithm: (FFA) respectively. The past study also includes ML, deep learning, best fit decreasing (BFD) techniques, round robin, and Modified BFD (MBFD) Nashaat et al. [8], Masdari and Khezri [6].

A BP-based heuristic is employed to examine a realistic strategy for regulating the number of transferred VMs [10]. The technique eliminates the transfer of VMs with constant workloads, hence reducing the total migrations. After a preliminary fit BP procedure, Beloglazov et al. [11] suggested a threshold-based VM migration optimisation, which 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. Conventional solution methods typically use the heuristic approach based on the greedy algorithm to search and use fewer additional details. A multi-dimensional space division model coordinates diverse resource demands and has been discussed by Li. et al. [12]. The authors separated the resource area into three categories: permissible, protected, and prohibited. From the first to the second, VM has priority. The proposed strategy would successfully minimise the number of necessary physical machines (PMs) by optimising the complementary resource demand. Song et al. [13] provide a dependable migration approach for VMs.

Further, VM is divided into four sections ranging from minimal to large allocation sizes. For the one-dimensional example, items are then categorised and packaged accordingly. It has been determined the accuracy of the algorithm. The algorithm may decrease the number of rounds of VM migrations.

Nevertheless, it is unable to ensure the quantity of migrated VMs. The proposed multi-capacity stochastic strategy differed from the current multi-dimensional BP since it was assumed that the VM is concerned with energy consumption by allocating the resource optimally. Multi-capacity BP captures server resource parameters more accurately. A two-stage heuristic method [14] is offered to limit the number of stochastic resources utilised. Heuristics based on BP are frequently employed to preserve resources by decreasing the number of required services.

Similar, complementary, and remaining together are used to achieve a better result in maximising nodes and bandwidth capital [15]. Shi et al. [49] want to increase CSP sales from an economic standpoint. They organise 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.

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 determining a solution by being kept and carried over to the next. In the literature review, several nature-inspired optimisation strategies have been discussed to address the issue of VM allocation and migration and energy consumption in cloud data centers.

2.1. GA

It is a classic heuristic algorithm 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 analyse the workloads assigned to the specific PMs. The proposed technique is a multi-objective optimisation strategy that results in power conservation.
Xu et al. [18] proposed a two-level control scheme for handling workload and VMPM mappings in their paper. 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 optimisation 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.

Wang et al. [19] modeled the VM problem using the elastic strategy. Their approach maximises resource utilisation, balances resource use across dimensions, and reduces contact traffic. The proposed technique used 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.

Wang et al. [20] proposed the two upper-level objective functions that occur in bi-level programming: reducing the gap problem, which 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 minimise the number of differences in server resource usage before and after scheduling.

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 VM in the current group and evaluation of possibilities of the VM to be placed with a contact flow and balancing the resources. The authors used to merge NSGAII's no dominated sorting function with GA's grouping feature. 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 NSGAII optimisation method.

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 minimise 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. Yousefipour et al. [24] proposed a mathematical model to lower costs and reduce power usage. Then, to solve the issue, they suggested an algorithm 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.

2.2. 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 optimise 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 realised using the graphs and labeled, indicating its significance. Xu et al. [26] proposed IMOPSO, a multi-objective evaluation model for dynamic VM deployment that optimised the migration time from VM to PM and the optimal use of resources. The proposed work realises the CloudSim platform and promising results obtained.

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 datacenters, the algorithm is energy efficient. One drawback of this research is that it assumes that each VM runs a single operation.

Dashti and Rahmani [28] suggested a solution to allocate migrated VMs from over-utilised PMs to enhance energy efficiency. A hierarchical architecture was proposed to fulfil users' requirements of users and providers, using QoS parameters and performance metrics. Furthermore, Li et al. [29] suggested a solution for effectively utilising resources to cause the migration of VMs and reduce data center energy consumption. The over and underutilised PMs are identified by avoiding the remigration strategy and thus attaining the proposed goal of reducing data center energy consumption and utilising the resources optimally utilisation.

2.3. ACO

ACO algorithm has gotten much attention from researchers in the last few years for solving various NP-hard problems. An optimal placement of VM in order to minimise the energy consumption and utilise the resources effectively, an ACO-based strategy was employed to obtain a non-dominated solution set. 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]. Ferdaus et al. [31] use the ACO metaheuristic to VM to the PM based on the utilised slots and history of the PM while the VM was allocated to the PM. To utilise the previous PM experience after the VM reallocation, a monitoring unit is placed, an underutilisation model. This is a practical solution with an excellent computation time.

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 utilise the previous experience of the PM after the reallocation of the VM, a monitoring unit is placed under underutilisation model usage that is rational, as well as minimising 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.

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.

Malekloo et al. [34] suggested a new method focused on the ACO metaheuristic coupled with a probabilistic decision criterion and a heuristic knowledge formula. The Pareto front process was employed for an optimum solution—the proposed technique was compared with other techniques to obtain the lowest power consumption with minimum 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.

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 realised 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 minimise 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 recognised 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.

2.4. ABC

Applying the ABC technique can address the issue of VM allocation and migration to the best fit available PM. The ABC strategy helps to minimise migration time and energy consumption. The ABC algorithm considers potential solutions such as bee food to find a suitable food supply as an optimal solution [37].

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

Maximum Number of Bees related to VMs	10000
Minimum Number of Bees employed relative to VM's	200
Load supplied for the VMs	$10^6 MIPS$
Total simulations per VM set	100
Maximum number of simulations	10000 ×10000
Implementation tool	Python
Platform	Anaconda

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

The given equation represents the power utilised during the application of the ABC technique in which it is assumed that employed bees ($Bees_{employed}$) having a valid solution with maximum information. However, scout bees are 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.

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

The condition applies to 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. [38] presented a model for evaluating energy performance that includes two controlling factors: GPU and CPU utilisation 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 live VM migration policy 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, poor scalability and long execution time remain a significant concerns.

Furthermore, Li et al. [39] formulated that VM migration needs to be considered by demonstrating the various constraints such as energy consumption, violations, CPU utilisation (CU), and efficient resource utilisation. 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.

2.5. 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 minimise the number of used PMs and achieve an optimum placement solution with the least amount of resource waste and energy use.

The viable solution is using the FA to place the VM at an appropriate PM by addressing the issue of energy consumption, violations of services, and migrations using the FFA.According to Kansal et al.[40], an FA optimisation technique should be used. They try to find the best method to optimise energy efficiency by minimising migrations. Perumal and Murugaiyan [41] address the VMP by integrating the FFA with the fuzzy technique to address the VM migration problem. This work attempted to minimise the active PMs and obtain an effective solution by utilising the resource optimally. As a result, 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.

2.6. Hybrid

The hybrid approach incorporates metaheuristic techniques. Cho et al. [42] proposed an ACOPS for load balancing in the cloud. It addressed the problem of effective resource utilisation and unwanted energy wastage during the VM migration process. It checked each server's remaining memory before scheduling to minimise 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.

Further, ACO was employed, and in each iteration round, all the ants found 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. Furthermore, the results show that the algorithm is good at balancing loads.

Table 2. Comparative analysis of existing work based on nature-inspired optimisation algorithms

Paper	Optimisation Algorithm	Objective	Optimised Resources
Mi et al. [17]	GA	Improvement in the utilisation of CPU and reduction in PM's number.	CPU
Xu et al. [18]	GA	Resource wastage and power consumption has been reduced to avoid the hotspot problem.	CPU, memory
Wang et al. [19]	GA	Resource utilisation maximised and balanced the usage of resources with minimum communication traffic.	CPU cycles, memory, and bandwidth
Wang et al. [20]	GA	Improving Resource utilisation, power consumption, and data locality in a viable manner.	CPU and hard disk utilisation
Liu et al. [21]	GA	Reduction in host count and balancing the usage of resources.	CPU cycles, storage space, and bandwidth
Sofia et al. [22]	GA	Minimising makespan and energy.	CPU
Riahi and Krichen [23]	GA	Reduction in wastage of resources and power consumption.	CPU and memory
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 minimised.	CPU, RAM, and bandwidth
Xu et al. [26]	PSO	Maximise 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 minimised, maximising 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 minimised.	CPU, memory, and bandwidth
Li et al. [29]	PSO	Resource utilisation has been improved, and power consumption has been minimised.	CPU, memory, and bandwidth
Gao et al. [30]	ACO	Improving overall utilisation of resources.	CPU, Memory, Network I/O
Ferdaus et al. [31]	ACO	Improving energy efficiency and minimum resource wastage.	CPU, Network and Storage
Wen et al. [32]	ACO	Minimising SLA Violations.	CPU, memory, and disk
Tan et al. [33]	ACO	Resource utilisation has been improved, Improvement in SLA violation and power consumption has been minimised.	CPU, memory, and disk
Malekloo et al. [34]	ACO	Maximise the usage of resources and reduction in communication costs.	CPU and networking elements
Li et al. [35]	ACO	Optimisation, improvement in violations of services, and reduction in energy consumption.	CPU
Liu et al. [36]	ACO	Minimise the energy consumption and SLA-V.	CPU
Karthikeyan [37]	ABC-BA	Minimise energy consumption and migrations.	CPU
Jiang et al. [38]	ABC	Minimising energy consumption, SLA-V migrations, and Resource utilisation has been improved.	CPU, memory, and bandwidth
Li et al. [39]	FA	Reduction in power consumption, minimising the total VM migrations, and improvement in overload probability.	Memory utilisation, bandwidth
Kansal et al. [40]	FA	Reduction in communication cost with time also.	CPU, RAM, and bandwidth
Perumal et al. [41]	FA	Reduction in communication cost and power consumption.	CPU, memory, and bandwidth
Cho et al. [42]	ACO, PSO	Minimising the number of migrations.	CPU
Sutr et al. [43]	ACO	Minimising the number of migrations.	CPU, power consumption
Satpathy et al. [44]	Crow search Algorithm	VM placement using the live migration.	CPU, Memory usage
Verma et al. [45]	Optimisation model	Reduction in energy consumption and improvement in SLA violations.	CPU

In the existing literature work, most of the research focused on the reduction in execution time and meeting the service quality at the user end [46], energy consumption, Nashaat et al. [8], Masdari and Khezri [6]. Nevertheless, the present research considers energy consumption, migration minimisation and SLA violation to minimise the false allocation, save energy during the VM allocation, and adopt the intelligent migration process. The proposed algorithm is integrated with the swarm intelligence-based ABC algorithm, which provides highly efficient results without degrading the service quality compared to existing works.

3. Proposed Work Algorithm

This paper introduces the novel fitness function to place the VM in the respective clusters, and the function is derived utilising the attribute architecture of Swarm Intelligence (SI) [47]. The proposed algorithm identified that colony algorithms have significantly impacted the selection and migration policy among most research-cited articles. The overall architecture is illustrated using fig. 3. The selection procedure gets incorporated at the IaaS layer when a user submits the request to the server. The server handler passes requests to the scheduler. To support 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. 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. One host could be assigned in the proposed solution system with few, many, or no VMs. Hence the moderation in the sample population leads us to use SI rather than a natural computing algorithm. 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.

1. Initialise allocation table to empty
2. Divide data into k number of clusters utilising Nashaat et al.[8]
3. Use flower pollination for the up gradation of hosts containing the VMs
4. Choose flower index
5. Migrate VM to flower index
6. Validate VM
7. Store information in the repository
8. Use the information to reduce further computation complexity.

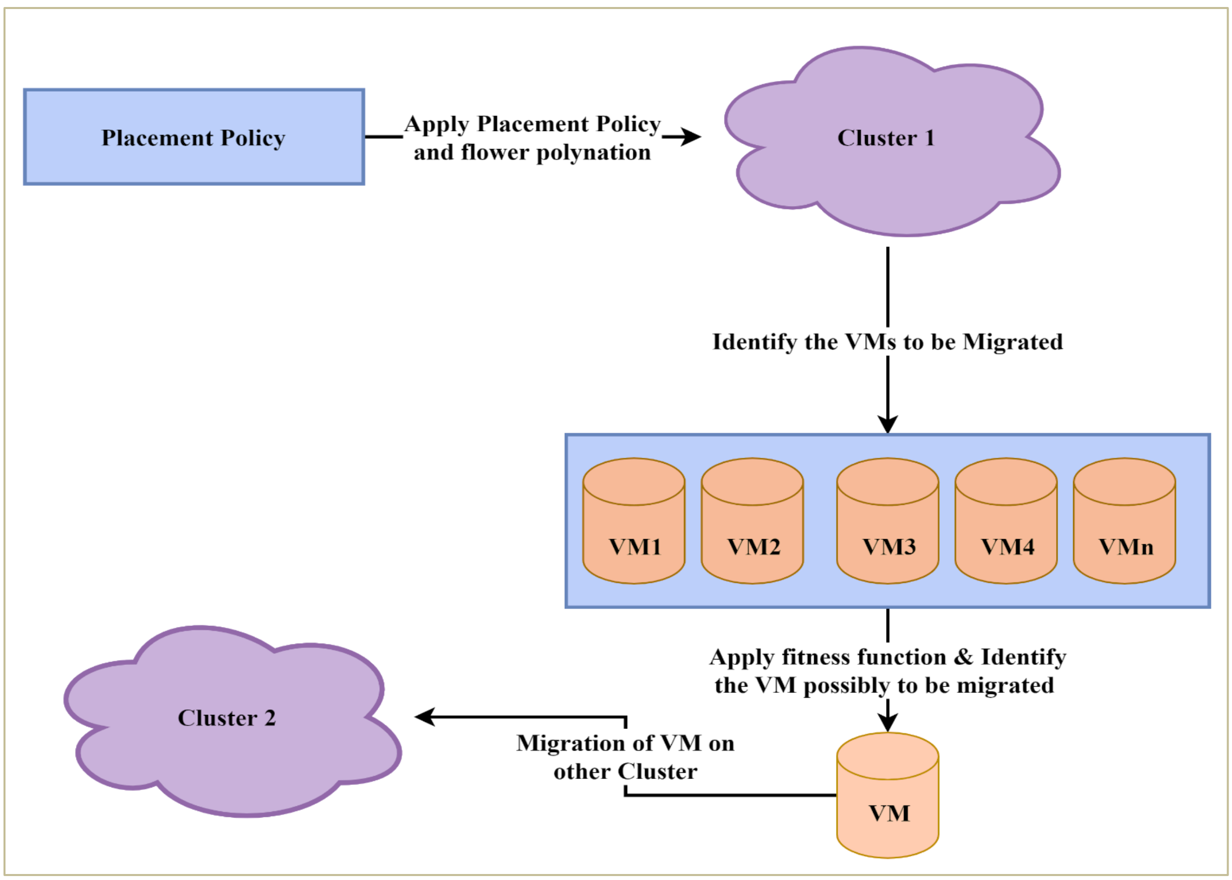


Fig. 3 Optimal Meta Heuristic Elastic Scheduling Architecture

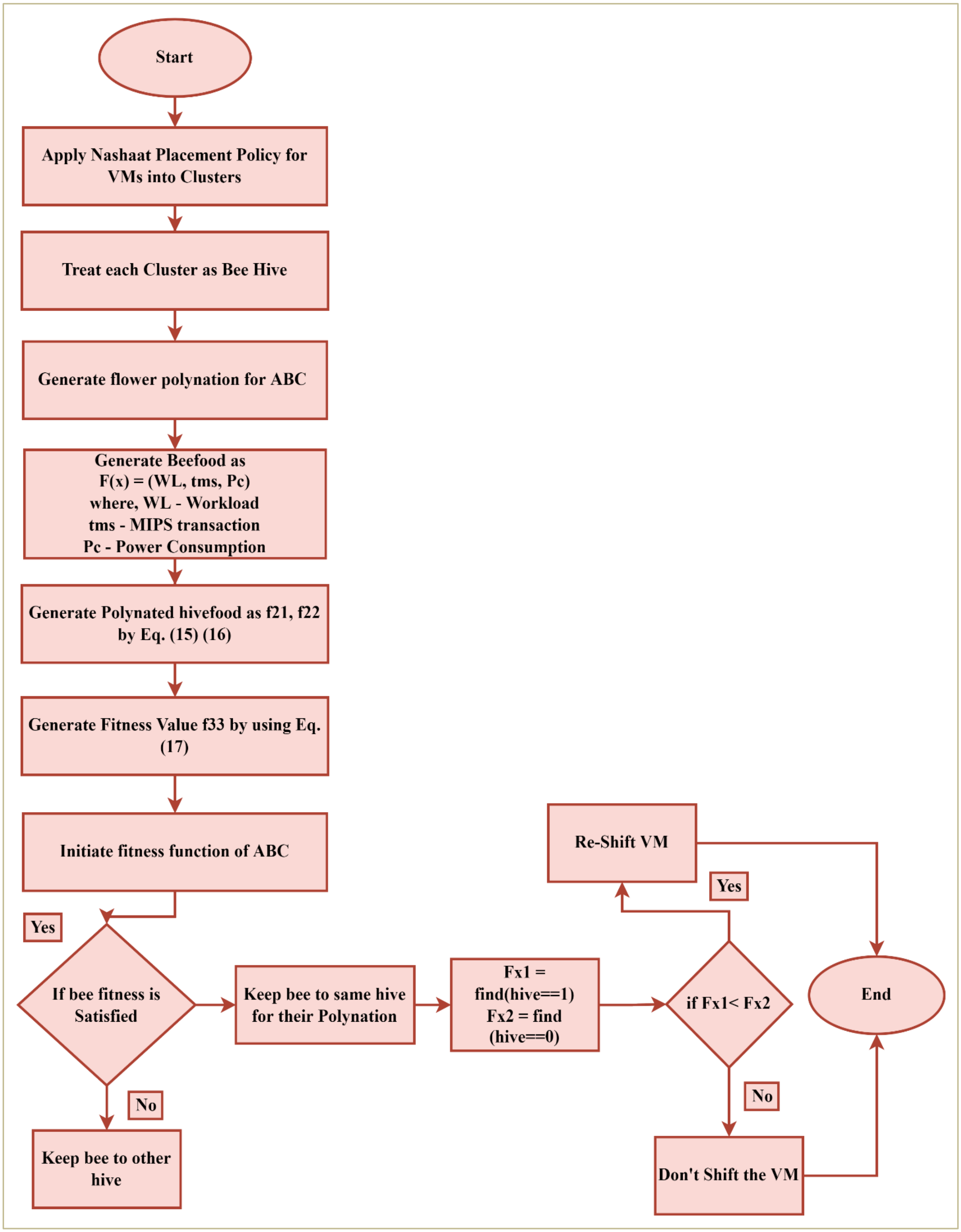


Fig. 4 Flowchart for Proposed Algorithm

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 utilisation in decreasing order.
Step2: Tfg = k;
Where Tfg is the total number of fly groups, and k is the total number of groups.
Step3: For each Tfg in Tfg:
a. Arrange VMs in the decreasing order of utilisation.
b. LI required (required flight intensity).

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

fx is the function aggregating the three components of ABC, i.e. the total amount of workload as 'wL', totally passed memory set instruction as 'tms', and total power consumption as 'PC'. $F_{structure}$ is the structure of fireflies of every fly group.

For the normalisation of the attributes using eq. (11)
I. $f_1^{out} = \{wL_n, tms, PC_n\}$ (13)
II. $f_2^{out} = Grouped - Host$ (14)

$f2$ 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 eq. (15), (16), and (17).

$$f_{21} = \frac{\int_{ni=0}^{hL.host\_count} WL_{ni}}{\sum_{j=1}^{hL.host\_count} WLj} \quad (15)$$

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

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

The role of $f3$ is to segregate the hosts based on the information collected by F_{21} light intensity.

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

$$f_1.Output = \{\frac{f_{21}}{f_{22}+f_{33}}\} \quad (18)$$

The proposed algorithm utilises the flower Pollination concept and takes a time interval of (f_0-f_n).
Where, n is the total number of intervals.

This gives the actual behavior of the PMs and thieves of n other light attained intensities which could be from the current intensity values to the program to the intensity a random probability of the firefly to be tired is created. If the probability is below 5, the fly intensity at the next time slot will be reduced by 10% and vice-versa.
Step 6: $f_{2_{Oi}}$ = Prepare $\{f_{2_{t_0 \ldots\ldots t_n}}\}$
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, the group calculates the group's requirements.
$Min_{PCg} = max(PC)$; $most_{suitable} = 0$
Step 10: For each hosts group in hL:
Step 11: If Costgroup.Satisfied (VM group, CPU demand)
Compute PCg =Allocated on group consumption.
Where PCg power consumption of group to accommodate VMs.
$if\ PCg <$ $\mathrm{Min_{PCg}}$
$Min_{PCg} = PCg$
$Most_{suitable}$
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. The evaluation of service parameters has been done to validate results, and an illustration is given in the next section.

5. Results and Validations

Based on the analysis, the following parameters have been evaluated.
SLA-V: SLA-V has many ways of measurement. In most of the research articles, the evaluation has been made based on power consumption. The proposed algorithm uses equation (13) to evaluate the SLA-V.

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

Where, $TPC_{UnderutilizedState}$ represents Total Power Consumed for underutilised states and $TPC_{OverutilizedState}$ represents Total Power Consumed for over-utilised states.
Migration count: It is the count of the total number of migrations that took place in a given time interval.
Total consumed power: It is the total power consumption in all states. The ordinal measures and the results are presented in Tables 3 and 4, respectively.

Table 3. Ordinal Measures

Maximum Number of VMs	10000
Minimum Number of VMs	200
Supplied Load	$10^6 MIPS$
Total simulations per VM set	100
Maximum number of simulations	10000 ×100
Implementation tool	Python
Tool Set	Spyder
Platform	Anaconda

Table 4. Result Illustration

Total Number of VMs	Migrations Nashaat et al. [8]	Migrations Masdari and Khezri [6]	Migrations OMES	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
10000	7030	5466	4235	1350	949.05	737.91	0.810	0.846	0.795

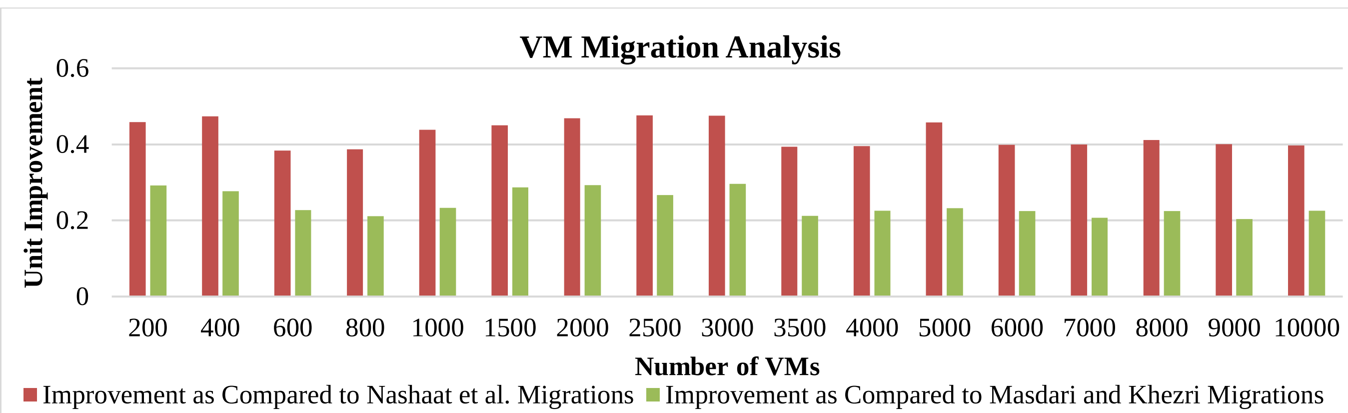


Fig 5. Unit Improvements for Migrations

The suggested approach significantly improves VM selection using intelligent VM placement of VMs on hosts. As indicated in Table 4, more power is conserved as a consequence of evaluating over 10000 VMs. The migration study reveals the average number of VM migrations recorded for Nashaat et al.[8] is 2802.18, compared to 2123.18 for Masdari and Khezri. Compared to these two endeavours, the OMES only accomplished 1637.41 migrations.

This is an average improvement of 1164.76 for OMES compared to Nashaat et al. and 485.76 for Masdari and Khezri. Fig. 3 depicts the unit improvement in the rise in the number of virtual machines. Unit improvement for VM migrations ranges between 0.3838 and 0.4766, with an average unit improvement of 0.4275.

Existing infrastructure or the OMES consumes much more energy as the number of VMs rises. Nevertheless, the average power usage of OMES is just 286.62 watts, compared to 512.20 watts when using Nashaat et al.[8] Moreover, 378.29 watts when using Masdari and Khezri[6]. This indicates that the OMES consumes 225.57 and 910.66 fewer units of energy than Nashaat et al. [8]. and Masdari and Khezri[6], respectively. Fig. 5 depicts the unit-wise improvement found in comparison to two previous studies. The power consumption unit improvement spans from 0.376 to 0.485 for Nashaat et al.[8] and from 0.202 to 0.294 for Masdari and Khezri's[6] work. OMES showed the average unit improvement in comparison to Nashaat et al.[8] and Masdari and Khezri[6] is 0.435% and 0.24%, respectively.

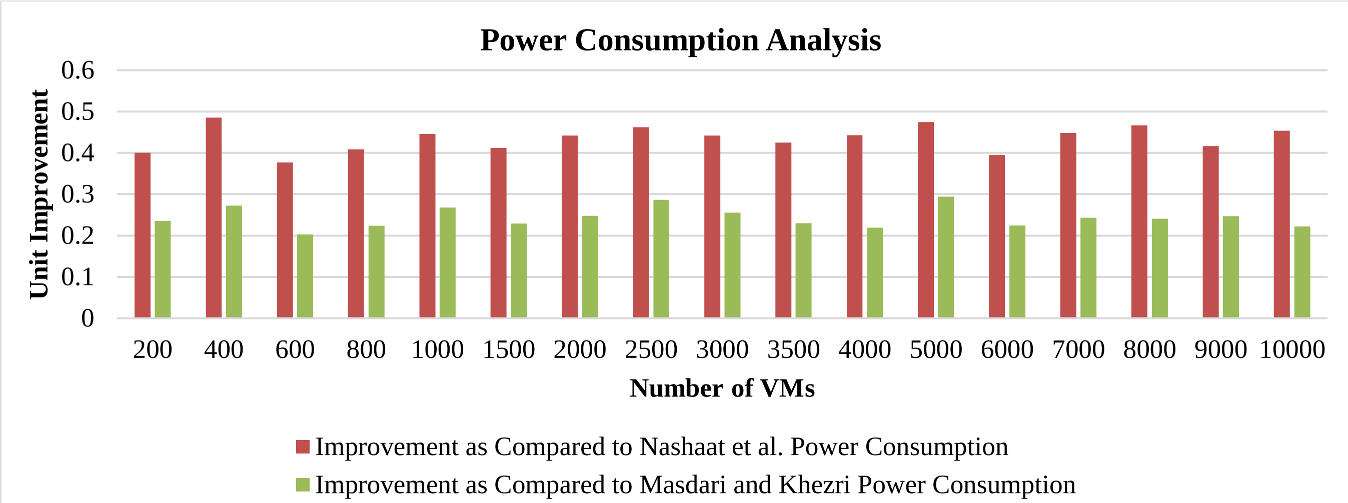


Fig 6. Unit Improvements for Power Consumption

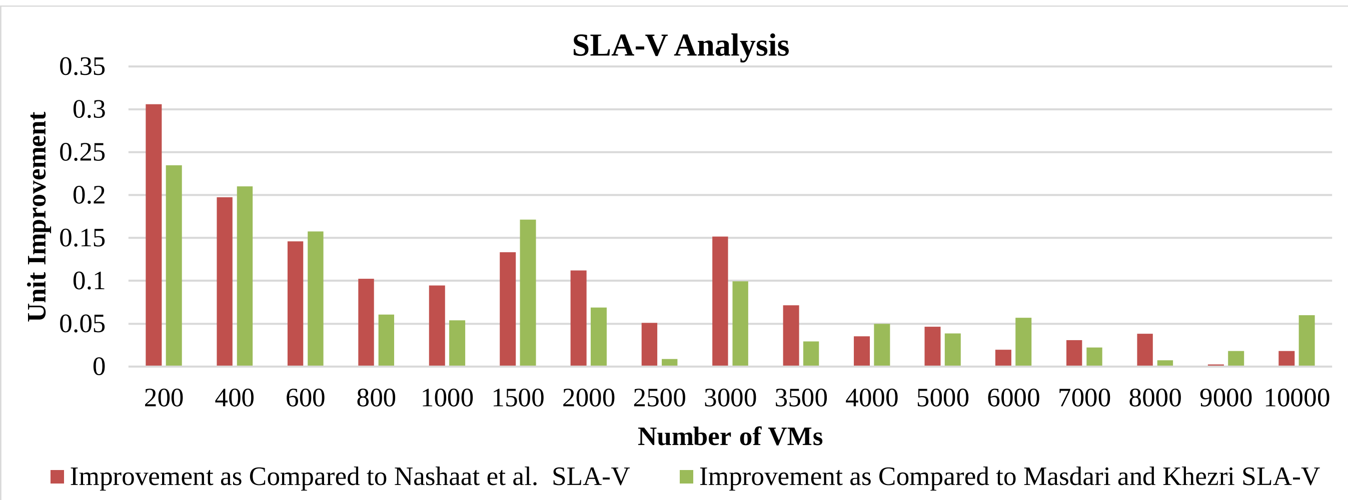


Fig. 7 Unit Improvements for SLA-V

The QoS delivered on the user end is determined by the number of SLA-V.

According to the SLA-V assessment, the OMES has the lowest SLA-V, i.e. 0.7011, compared to other Masdari and khezri [6], i.e. 0.759 and Nashaat et al. [8], i.e., 0.769. This demonstrates that OMES displays 0.058 and 0.068 less SLA-V in comparative analysis with other mentioned in [6] and [8] over 10,000 VMs.

Fig. 7 displays the SLA-V improvement per unit observed for OMES as a function of the number of VMs deployed during the trial. When comparing to Nashaat et al.[8], the observed unit improvement ranges from 0.0025 to 0.3059, and when comparing to Masdari and Khezri [6], it ranges from 0.0072 to 0.234. As a result, OMES outperforms Nashaat et al.[8] by an average of 0.091 units, and Masdari and Khezri [6] by an average of 0.792 units. Despite an increase in the total number of VMs, the OMES can minimise the total VM migrations by conducting a rigorous simulation study based on three essential criteria such as energy consumption, total migrations and SLA-V. Moreover, in the overall comparative analysis, the proposed work shows highly remarkable outcomes regarding energy consumption(Watt), minimisation of migrations and SLA violations. The proposed work effectively utilised the bandwidth by allocating the VM optimally by considering the VM selection policies on a priority basis. The proposed work exhibited benchmark improvement over the Nashaat et al.[8]. Therefore, a glimpse of the feature analysis of the proposed work performed against Nashaat et al.[8] work is also summarised in Table 5.

Further, it is observed that in comparison to the existing studies. OMES exhibits the least power consumption and SLA-Vs. When analysed using 10000 VMs, Table 6 illustrates the comparative analysis of CPU utilisation by different techniques. The proposed technique shows the least CPU utilisation.

Table 5. Comparative feature analysis of the proposed work against existing work

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

Table 6. Comparative Analysis of CPU utilisation

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

Swarm Intelligence (SI) aims to find the best optimal solution for VMs to minimise the need for many 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 regarding 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 utilisation 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 optimisation architecture to further reduce energy consumption by improving the VM selection process and minimising VM migrations.

Declaration

Ethical Approval

The author declares that the 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.

Authors' Contributions

The author declares that both authors have contributed equally to designing the paper.

References

[1] Maede Yavari, Akbar Ghaffarpour Rahbar, and Mohammad Hadi Fathi, “Temperature and Energy-aware Consolidation Algorithms in Cloud Computing,” *Journal of Cloud Computing*, vol. 8, 2019. [CrossRef] [Google Scholar] [Publisher link]

[2] Peiyun Zhang, Mengchu Zhou, and Xuelei Wang, “An Intelligent Optimization Method for Optimal Virtual Machine Allocation in Cloud Data Centers,” *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1725–1735, 2020. [CrossRef] [Google Scholar] [Publisher link]

[3] Anton Beloglazov, and Rajkumar Buyya, “Energy Efficient Resource Management in Virtualized Cloud Data Centers,” *2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, 2010. [CrossRef] [Google Scholar] [Publisher link]

[4] Longhao Shu, and Xi Li, “Temperature-aware Energy Minimization Technique through Dynamic Voltage Frequency Scaling for Embedded Systems,” *2010 2nd International Conference on Education Technology and Computer*, 2010. [CrossRef] [Google Scholar] [Publisher link]

[5] Patricia Arroba et al., “Dynamic Voltage and Frequency Scaling-aware Dynamic Consolidation of Virtual Machines for Energy Efficient Cloud Data Centers,” *Concurrency and Computation Practice and Experience*, vol. 29, no. 10, 2017. [CrossRef] [Google Scholar] [Publisher link]

[6] Mohammad Masdari, and Hemn Khezri, “Efficient VM Migrations using Forecasting Techniques in Cloud Computing: A Comprehensive Review,” *Cluster Computing*, vol. 23, pp. 2629-2658, 2020. [CrossRef] [Google Scholar] [Publisher link]

[7] G. Senthil Kumar, “Framework for Hypervisor in Grid,” *SSRG International Journal of Electrical and Electronics Engineering*, vol. 1, no. 1, pp. 9-15, 2014. [CrossRef] [Publisher link]

[8] Heba Nashaat, Nesma Ashry, and Rawya Rizk , “Smart Elastic Scheduling Algorithm for Virtual Machine Migration in Cloud Computing,” *Journal of Supercomputing*, vol. 75, pp.3842-3865, 2019. [CrossRef] [Google Scholar] [Publisher link]

[9] Jiangtao Zhang, Hejiao Huang, and Xuan Wang, “Resource Provision Algorithms in Cloud Computing: A Survey,” *Journal of Network and Computer Applications*, vol.64, pp. 23-42, 2016. [CrossRef] [Google Scholar] [Publisher link]

[10] Tiago C. Ferreto et al., “Server Consolidation with Migration Control for Virtualized Data Centers,” *Future Generation Computer Systems*, vol. 27, no. 8, pp. 1027-1034, 2011. [CrossRef] [Google Scholar] [Publisher link]

[11] Anton Beloglazov, and Rajkumar Buyya, “Managing Overloaded Hosts for Dynamic Consolidation of Virtual Machines in Cloud Data Centers under Quality of Service Constraints,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp.1366-1379, 2012. [CrossRef] [Google Scholar] [Publisher link]

[12] Xin Li et al., "Energy Efficient Virtual Machine Placement Algorithm with Balanced and Improved Resource Utilization in a Data Center," *Mathematical and Computer Modelling*, vol. 58, no. 5-6, pp. 1222-1235, 2013. [CrossRef] [Google Scholar] [Publisher link]

[13] Weijia Song et al., "Adaptive Resource Provisioning for the Cloud using Online Bin Packing," *IEEE Transactions on Computers*, vol. 63, no. 11, pp. 2647-2660, 2013. [CrossRef] [Google Scholar] [Publisher link]

[14] Inkwon Hwang, and Massoud Pedram, "Hierarchical Virtual Machine Consolidation in a Cloud Computing System," *2013 IEEE Sixth International Conference on Cloud Computing*, 2013. [CrossRef] [Google Scholar] [Publisher link]

[15] Jiangtao Zhang et al., "SLA Aware Cost Efficient Virtual Machines Placement in Cloud Computing," *2014 IEEE 33rd International Performance Computing and Communications Conference*, 2014. [CrossRef] [Google Scholar] [Publisher link]

[16] A. Richard William, and J. Senthilkumar, "Auto-Scalability in Cloud: A Survey of Energy and Sla Efficient Virtual Machine Consolidation," *SSRG International Journal of Computer Science and Engineering*, vol. 3, no. 11, pp. 7-11, 2016. [CrossRef] [Publisher link]

[17] Haibo Mi et al., "Online Self-reconfiguration with Performance Guarantee for Energy-efficient Large-scale Cloud Computing data Centers," *2010 IEEE International Conference on Services Computing*, 2010. [CrossRef] [Google Scholar] [Publisher link]

[18] Jing Xu, and Jose A.B. Fortes, "Multi-objective Virtual Machine Placement in Virtualized Data Center Environments," *2010 IEEE/ACM International Conference on Green Computing and Communications and International Conference on Cyber, Physical and Social Computing*, 2010. [CrossRef] [Google Scholar] [Publisher link]

[19] Sinong Wang, Huaxi Gu, and Gang Wu, "A New Approach to Multi-objective Virtual Machine Placement in Virtualized Data Center," *2013 IEEE 8th International Conference on Networking, Architecture and Storage*, 2013. [CrossRef] [Google Scholar] [Publisher link]

[20] Xiaoli Wang, Yuping Wang, and Yue Cui, "A New Multi-objective Bi-level Programming Model for Energy and Locality Aware Multi-job Scheduling in Cloud Computing," *Future Generation Computer Systems*, vol. 36, pp. 91-101, 2014. [CrossRef] [Google Scholar] [Publisher link]

[21] Chao Liu et al., "A New Evolutionary Multi-objective Algorithm to Virtual Machine Placement in Virtualized Data Center," *2014 IEEE 5th International Conference on Software Engineering and Service Science*, 2014. [CrossRef] [Google Scholar] [Publisher link]

[22] A. Sathya Sofia, and P. Ganesh Kumar, "Multi-objective Task Scheduling to Minimize Energy Consumption and Makespan of Cloud Computing Using NSGA-II," *Journal of Network and Systems Management*, vol. 26, pp. 463-485, 2018. [CrossRef] [Google Scholar] [Publisher link]

[23] Montassar Riahi, and Saoussen Krichen, "A Multi-objective Decision Support Framework for Virtual Machine Placement in Cloud Data Centers: A Real Case Study," *Journal of Supercomputing*, vol. 74, no. 7, pp. 2984-3015, 2018. [CrossRef] [Google Scholar] [Publisher link]

[24] Amin Yousefipour, Amir Masoud Rahmani, and Mohsen Jahanshahi, "Energy and Cost-aware Virtual Machine Consolidation in Cloud Computing," *Software: Practice and Experience*, vol. 48, no. 10, pp.1758-1774, 2018. [CrossRef] [Google Scholar] [Publisher link]

[25] Lizheng Guo et al., "Multi-objective Optimization for Data Placement Strategy in Cloud Computing," *International Conference on Information Computing and Applications*, pp. 119-126, 2012. [CrossRef] [Google Scholar] [Publisher link]

[26] Bo Xu et al., "Dynamic Deployment of Virtual Machines in Cloud Computing using Multi-objective Optimization," *Soft Computing*, vol. 19, pp. 2265-2273, 2015. [CrossRef] [Google Scholar] [Publisher link]

[27] Shangguang Wang et al., "Provision of Data-intensive Services through Energy-and QoS-Aware Virtual Machine Placement in National Cloud Data Centers," *IEEE Transactions on Emerging Topics in Computing*, vol. 4, no. 2, pp. 290-300, 2015. [CrossRef] [Google Scholar] [Publisher link]

[28] Seyed Ebrahim Dashti, and Amir Masoud Rahmani, "Dynamic VMs Placement for Energy Efficiency by PSO in Cloud Computing," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 28, no. 1-2, pp. 97-112, 2016. [CrossRef] [Google Scholar] [Publisher link]

[29] Hongjian Li et al., "Energy-efficient Migration and Consolidation Algorithm of Virtual Machines in Data Centers for Cloud Computing," *Computing*, vol. 98, pp. 303-317, 2016. [CrossRef] [Google Scholar] [Publisher link]

[30] Yongqiang Gao et al., "A Multi-Objective Ant Colony System Algorithm for Virtual Machine Placement in Cloud Computing," *Journal of computer and System Sciences*, vol. 79, no. 8, pp. 1230-1242, 2013. [CrossRef] [Google Scholar] [Publisher link]

[31] Md Hasanul Ferdaus et al., "Virtual Machine Consolidation in Cloud Data Centers using ACO Metaheuristic," *European Conference on Parallel Processing*, pp. 306-317, 2014. [CrossRef] [Google Scholar] [Publisher link]

[32] Wei-Tao Wen et al., "An ACO-based Scheduling Strategy on Load Balancing in Cloud Computing Environment," *2015 9th International Conference on Frontier of Computer Science and Technology*, 2015. [CrossRef] [Google Scholar] [Publisher link]

[33] Mingzhe Tan et al., "An Energy-aware Virtual Machine Placement Algorithm in Cloud Data Center," *Proceedings of the 2nd International Conference on Intelligent Information Processing*, pp. 1-9, 2017. [CrossRef] [Google Scholar] [Publisher link]

[34] Mohammad-Hossein Malekloo, Nadjia Kara, and May El Barachi, "An Energy Efficient and Sla Compliant Approach for Resource Allocation and Consolidation in Cloud Computing environments," *Sustainable Computing: Informatics and Systems*, vol. 17, pp. 9-24, 2018. [CrossRef] [Google Scholar] [Publisher link]

[35] Zhihua Li et al., "Energy-aware and Multi-resource Overload Probability Constraint-based Virtual Machine Dynamic Consolidation Method," *Future Generation Computer Systems*, vol. 80, pp. 139-156, 2018. [CrossRef] [Google Scholar] [Publisher link]

[36] Fagui Liu et al., "A Virtual Machine Consolidation Algorithm Based on Ant Colony System and Extreme Learning Machine for Cloud Data Center," *IEEE Access*, vol. 8, pp. 53-67, 2019. [CrossRef] [Google Scholar] [Publisher link]

[37] K. Karthikeyan et al., "Energy Consumption Analysis of VM Migration in Cloud Using Hybrid Swarm Optimization (ABC–BA)," *The Journal of Supercomputing*, vol. 76, pp. 3374–3390, 2020. [CrossRef] [Google Scholar] [Publisher link]

[38] Jianhua Jiang et al., "DataABC: A Fast ABC Based Energy-efficient Live VM Consolidation Policy with Data-intensive Energy Evaluation Model," *Future Generation Computer Systems*, vol. 74, pp. 132-141, 2017. [CrossRef] [Google Scholar] [Publisher link]

[39] Xiao-Ke Li et al., "Virtual Machine Placement Strategy based on Discrete Firefly Algorithm in Cloud Environments," *2015 12th International Computer Conference on Wavelet Active Media Technology and Information Processing*, 2015. [CrossRef] [Google Scholar] [Publisher link]

[40] Nidhi Jain Kansal, and Inderveer Chana, "Energy-aware Virtual Machine Migration for Cloud Computing-a Firefly Optimization Approach," *Journal of Grid Computing*, vol. 14, pp. 327-345, 2016. [CrossRef] [Google Scholar] [Publisher link]

[41] Boominathan Perumal, and Aramudhan Murugaiyan, "A Firefly Colony and Its Fuzzy Approach for Server Consolidation and Virtual Machine Placement in Cloud Datacenters," *Advance in Fuzzy Systems*, 2016. [CrossRef] [Google Scholar] [Publisher link]

[42] Keng-Mao Cho et al., "A Hybrid Meta-heuristic Algorithm for VM Scheduling with Load Balancing in Cloud computing," *Neural Computing and Application*, vol. 26, pp. 1297-1309, 2015. [CrossRef] [Google Scholar] [Publisher link]

[43] Sandeep G. Sutar, Pallavi J. Mali, and Amruta Y. More, "Resource Utilization Enhancement through Live VM Migration in Cloud using Ant Colony Optimization Algorithm," *International Journal of Speech Technology*, vol. 23, pp. 79–85, 2020. [CrossRef] [Publisher link]

[44] Anurag Satpathy et al., "Crow Search Based virtual Machine Placement Strategy in Cloud Data Centers with Live Migration," *Computers and Electrical Engineering*, vol. 69, pp. 334-350, 2017. [CrossRef] [Google Scholar] [Publisher link]

[45] Garima Verma, "Secure VM Migration in Cloud: Multi-Criteria Perspective with Improved Optimization Model," *Wireless Personal Communication*, vol. 124, pp. 75-102, 2022. [CrossRef] [Google Scholar] [Publisher link]

[46] S. Jenicka, and T. Grace Shalini, "Enhanced Hybrid Dual Channel Algorithm (EHDC) for Data Dissemination in VANETs," *SSRG International Journal of Computer Science and Engineering*, vol. 8, no. 7, pp. 1-9, 2021. [CrossRef] [Publisher link]

[47] K. Manojkumar, and S. Devi, "Jamming Attack in Wireless Sensor Networks using Ant Colony Algorithm," *SSRG International Journal of Computer Science and Engineering*, vol. 8, no. 2, pp. 6-9, 2021. [CrossRef] [Publisher link]

[48] Fei Zhang et al., "A Survey on Virtual Machine Migration: Challenges, Techniques, and Open Issues," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1206-1243, 2018. [CrossRef] [Google Scholar] [Publisher link]

[49] Lei Shi et al., "Provisioning of Requests for Virtual Machine Sets with Placement Constraints in IaaS Clouds," *2013 IFIP/IEEE International Symposium on Integrated Network Management*, 2013. [Google Scholar] [Publisher link]