Evolutionary Intelligence (2020) 13:725–739
https://doi.org/10.1007/s12065-020-00386-9

RESEARCH PAPER

ChicWhale optimization algorithm for the VM migration in cloud computing platform

Srinivas Byatarayanapura Venkataswamy[1] · Indrajit Mandal[2] · Seetharam Keshavarao[3]

Received: 14 October 2019 / Revised: 30 January 2020 / Accepted: 8 March 2020 / Published online: 2 April 2020

Abstract
Nowadays, Virtual Machine (VM) migration becomes very popular in the cloud computing platform. Various VM migration based mechanisms are designed for optimal VM placement but remain a challenge due to improper energy consumption in the cloud model. This paper proposes an approach for VM migration in the cloud using an optimization algorithm, Chicken-Whale optimization algorithm (ChicWhale), which is developed by integrating the Whale optimization algorithm in Chicken swarm optimization. In the developed approach, a local migration agent is utilized for monitoring the memory and resources utilization in the cloud continuously, and the VM is migrated using the service provider based on the requirement of the VMs to complete a task assigned. At first, the cloud system is designed, and then the proposed ChicWhale is employed by moving the VMs optimally, and the fitness function for best VM migration is carried out by considering several parameters, like load, migration cost, resource availability, and energy. The performance of the VM migration strategy based on ChicWhale is evaluated in terms of energy consumption, resource availability, migration cost, and load. The proposed ChicWhale method achieves the maximal resource availability of 0.989, minimal migration cost of 0.0564, the minimal energy consumption of 0.481, and the minimal load of 0.0001.

Keywords Virtual machine migration · Chicken swarm optimization · Cloud computing · Whale optimization algorithm · Resource availability

1 Introduction

Cloud computing is nothing, but a computing theory for allowing the utilization of computing infrastructure at more than one level of abstraction. Due to the implications for higher availability and flexibility at limited cost, cloud computing has been paid a great deal of attention [1–4]. Meanwhile, due to the uneven task scale and various computing capacities of nodes, some computing nodes present in the cloud are underutilized whereas others are overloaded that results in unbalanced load distribution [1, 5, 6]. Hence, it is imperative for spreading loads over computing nodes to enhance user requirements [1, 7]. Several users and enterprises are permitted to maintain and construct data centers. The individuals of the cloud also enjoy several kinds of computing services provided by the public cloud [8]. Cloud computing with optimization algorithms is determined using NIST model for convenient and accessing various computing resources, such as services applications, networks, servers, and storage [9–12]. The data placement algorithms are used to improve the quality of various factors, which affect the workflows of the cloud computing environment [13]. The optimization algorithms are used in engineering research and industrial application to improve the performance of the system [14, 15].

VM migration is utilized in cloud data centres to save energy [16]. VM not only provides secure and efficient computing resources, and migration is also performed from the Physical Machine (PM) [16]. VM migration moves from one PM to others. This migration should be visible to the Operating System (OS) of guest, remote clients of VM, and the applications running on OS [17]. VM monitors, such as Xen, provide a mechanism to map VM into physical resources

✉ Srinivas Byatarayanapura Venkataswamy
srini13986@gmail.com

[1] Information Science and Engineering, Atria Institute of Technology, Bangalore, Karnataka, India

[2] Computer Science and Engineering, MINA Institute of Engineering and Technology, Nalgonda, Telangana, India

[3] Research and Development, Anveshana, Hyderabad, Telangana, India

[18]. Developing hardware is usually mapped with VM that is system independent. Virtualization managers, such as VM manager, and VM monitor are used to communicate with lower-level devices, and allocating and coordinating VMs [19]. The majority of the cloud operations are supported using VM migration and optimization algorithms [20], like zero-downtime hardware maintenance [21], traffic management [22–24], server consolidation [25], and energy management. The VM migration pattern is utilized for migrating VM from the source server to the target server. The migration patterns are classified into non-live migration and live VM migration. The live migration does not quit the service during the migration process; meanwhile, non-live pattern follows the pause, resume, and copy approaches for migrating the VM [26].

Live migration is defined by transforming VMs without distracting the services of end-users, which causes limited-service downtime as transferring the whole VM to destination host [27]. Live migration in VM reduces the malicious attacks, component as well as the infrastructural faults. Live migration involves the dynamic transfer of the VM from one PM to the other, which is apparent to guest OS, and remote users of VM [28], while the VM migrations are essential for resource management in the cloud [29]. Several algorithms are utilized for deciding which VMs need to be migrated from the source server to support decision making, including VM resource usage and resource requirements; QoS; usage of server resource; such as bandwidth or CPU that improves the tasks completion time, workload burst, and the prediction of VM resource [27, 30].

The primary intention of this research is to develop an approach for VM migration in the cloud computing platform by proposing an optimization algorithm. At first, the cloud system is designed, and the migration agent monitors the resource utilization and memory in the cloud repeatedly, and the loads are balanced by migrating the VMs to handle the tasks. Here, a VM migration strategy is established using the proposed ChicWhale. Then, the other parameters, such as resource availability, energy, migration cost, are computed. Here, the movement of VM from PM to another PM is an optimization issue, and it is performed through the developed ChicWhale, which is the combination of the Chicken Swarm Optimization (CSO) and Whale Optimization Algorithm (WOA).

The main contributions of the research paper towards VM migration in the cloud computing environment are illustrated below.

- Designing the multi-objective model for VM migration using the load factor, and the resource availability, for choosing the appropriate VM for the migration in the cloud.
- Proposing the ChicWhale Optimization algorithm by integrating CSO in WOA for selecting the optimal VM to handle the tasks effectively in the cloud.

The paper is structured in the following manner: Sect. 2 discusses existing methods of VM migration with challenges of the methods that remain the motivation for the research. The proposed method of ChicWhale is demonstrated in Sect. 3, and Sect. 4 provides the results of the developed methods. At last, Sect. 5 **concludes the research work.

2 Motivation

This section presents the literature survey of several methods utilized for VM migration, and the challenges of the existing works are discussed.

2.1 Literature survey

Several methods related to VM migration are described and analyzed as follows: Li et al. [31] presented the dynamic energy-efficient VM migration and consolidation approach using the energy-efficient model. In this framework, the double threshold with the multi-resource utilization was designed for VM migration. Then, the modified version of Particle Swarm Optimization (PSO) was established to resolve the issue in the traditional heuristic approaches. Here, the energy-efficient was found better, but required high service downtime. Satpathy et al. [32] developed a queueing structure for managing and scheduling a large set of VMs. After that, the Crow Search-based VM Placement (CSAVMP) approach was introduced for mitigating the power consumption, and resource wastage at data centers. The method failed to compute the migration of live VM placement for parallel, enhanced serial strategies, and serial. Rodrigues et al. [33] modeled an approach to reduce the service delay with the two cloudlet servers. This framework has a dual focus on controlling processing Delay, and communication elements through VM migration to improve the transmission delay. The method did not examine multiple cloudlets and Dynamicity issues of Edge Cloud Computing (ECC). Karthikeyan et al. [34] employed Naive Baye's classifier with hybrid Artificial Bee Colony–Bat Algorithm (ABC–BA) for reducing the energy consumption in VM placement. Innovative classifiers and other optimization algorithms with various performance measures were not considered for the migration of VM.

Paulraj et al. [35] developed resource-aware VM migration in the Internet of Service (IoT) cloud. In this framework, any sudden modify happened in sensing is observed by the clustering. Then, the best target server was chosen based on job arrival rate, and resource utilization. The major drawback of this method is, the VM was not optimally placed on the destination server. He et al. [36] developed Software-Defined Networking (SDN)-based cloud data centers for live VM migration. Here, the downtime, response time, and the evacuation time of various migrations are reduced, but the optimal downtime adjustment for various live migration tasks was not analyzed. Kansal et al. [37] presented an energy-aware VM migration for cloud computing using Firefly algorithm. This framework was utilized for maximally migrating loaded VM to the less loaded active node to maintain the energy efficiency of data centers. The method does not analyze the performance of the public, private or hybrid cloud. Jargalsaikhan Narantuya et al. [38] developed a service-aware approach for Cloud-to-cloud (C2C) migration of various VMs based on network traffic intensity for determining the dependent VMs of migration sequence for reducing the service downtime. The method failed to discuss the position of VMs in the destination before the migration of the source cloud. Patel [39] developed a multi-objective Dolphin Echolocation technique, which minimizes the power consumption, and resource wastage, but the cost of migration was high. Simarro et al. [40] developed a scheduling model for optimizing VM placements, which addresses the dynamic deployment cost of the virtual resources, but the required energy was high. Xu et al. [41] developed an improved multi-objective particle swarm optimization (IMOPSO) to improve resource usage, and reduce turnaround times of VMs. It obtained a Pareto optimal solution set, which had a better convergence and distribution. Anyhow, it did not focus on the dynamic environment. Yermolovich et al. [42] developed an optimizing trace-based virtual machine, which was used to store the execution path of the VM and also find the VM byte codes in the environment. It improves the overall performance, but the processing time was quite high when compared to other methods. Gao et al. [43] developed a multi-objective ant colony system algorithm for the VM placement problem, which obtains a set of non-dominated solutions that simultaneously minimizes total power consumption, and resource wastage. It was not scalable for the larger dataset (Table 1).

2.2 Challenges

This section deals with the challenges faced by the existing techniques of VM migration in the cloud platform.

- Due to the size of the image, it is not possible to Translocate the live VM across low bandwidth high latency WAN [44].
- In the post copy, at the destination while the VM tries for fetching the pages that are not changed yet, it redirects toward the source host, and therefore produces the network issues as VM at the source host is suspended [44].
- If the VM migration needs to carry on for running from suspends point after the migration, all the running states have to be transferred to the target site [22].
- After the migration of new location, few steps are necessary for making reachable to their users. For the live migration, the open connections must be kept alive during the migration [22].
- VM migration approaches aim to the total number of objectives, which includes mitigating the service downtime, optimizing the duration of migration, reducing the duration of QoS degradation, and the best bandwidth resource utilization during VM migration process [32]**.**
- In [35], a Resource-aware approach was developed for VM migration in the cloud computing environment. Here, the total amount of migrations, energy utilization and migration time was found better but failed to apply another optimization algorithm to place the VM in the destination server.

3 Cloud setup

This section depicts the VM migration scheme using the ChicWhale optimization algorithm in the cloud computing environment. The block diagram of the VM migration model in the cloud is depicted in Fig. 1. In the last few years, cloud computing plays a vital role in computer science. Cloud computing provides a flexible, and simple way of maintaining and retrieving data and files. The cloud model consists of several PMs to solve the requests from the users, and PM has to collect the VMs for dynamically processing the tasks. The VM available in the cloud is produced dynamically for mitigating the bottleneck issues present in cloud computing and also virtualization problem enhances the speed. The services required by the users arrive in the cloud as a task, and each task is given to the VM based on the round-robin manner. Here, the set of VMs is controlled by PM; the cloud has the load balancer, which checks the load of PM. When the load value of PM is greater than the threshold value, the VM migration is performed.

3.1 Proposed VM migration algorithm

The algorithmic steps involved of the developed VM migration model are illustrated below,

1. In the first step, the cloud with M number of VMs and N number of PMs are initialized.
2. In this step, the migration cost of PM is initially fixed to the high value and hence, the migration cost is equal to 1.
3. After that, the incoming tasks of VM are assigned using round-robin fashion at the time duration.

Table 1 Review on VM migration techniques

Authors	Methods	Advantages	Disadvantage
Li et al. [1]	Dynamic energy-efficient virtual machine (VM) migration and consolidation algorithm based on a multi-resource energy efficient model	Better energy efficiency in data center for cloud computing	High service downtime
Satpathy et al. [2]	Two-tier virtual machine placement algorithm	Better performance in terms of number of servers used, power consumed and resources wastage	High memory usage as the size of VM increases
Rodrigues et al. [3]	An analytic model for the Service Delay in ECC systems	Better performance and service delay	Multiple cloudlets and Dynamicity issues of ECC are not analyzed
Karthikeyan et al. [8]	Naive Bayes classifier with hybrid optimization using Artificial Bee Colony–Bat Algorithm	limited energy consumption	Innovative classifier and optimization with different performance measures was not considered for VM migration
Paulraj et al. [5]	Resource-aware virtual machine migration technique	It reduces the migration time, downtime and end to end delay	The virtual machines was not optimally placed on the destination server
He et al. [7]	SDN-enabled cloud data centers	Reduction in the response time, downtime, and the total evacuation time of multiple migrations	The optimal downtime adjustment configuration for different live migration tasks are not developed
Kansal et al. [17]	Firefly algorithm	It reduces the energy consumption of cloud data centers and the number of migrations	The performance is not analyzed with private, public or hybrid cloud
Narantuya et al. [31]	Service-aware strategy for C2C migration	Identifies the dependent VMs and significantly decreases service downtime	The location of the VMs at the destination before migrating from the source cloud is not discussed
Patel [39]	A multi-objective Dolphin Echolocation technique	It minimizes power consumption and resource wastage	High migration cost
Simarro et al. [40]	A scheduling model for optimizing VM placements	Addresses the dynamic deployment cost of the virtual resources	High energy consumption
Xu et al. [41]	An improved multi-objective particle swarm optimization (IMOPSO)	Improve resource usage, and reduce turnaround times of VMs, also better convergence and distribution	Did not focus on the dynamic environment
Yermolovich et al. [42]	An optimizing trace-based virtual machine	Store the execution path of the VM and find the VM byte codes	High processing time
Gao et al. [43]	A multi-objective ant colony system algorithm	Minimizes total power consumption and resource wastage	Not scalable for the larger dataset

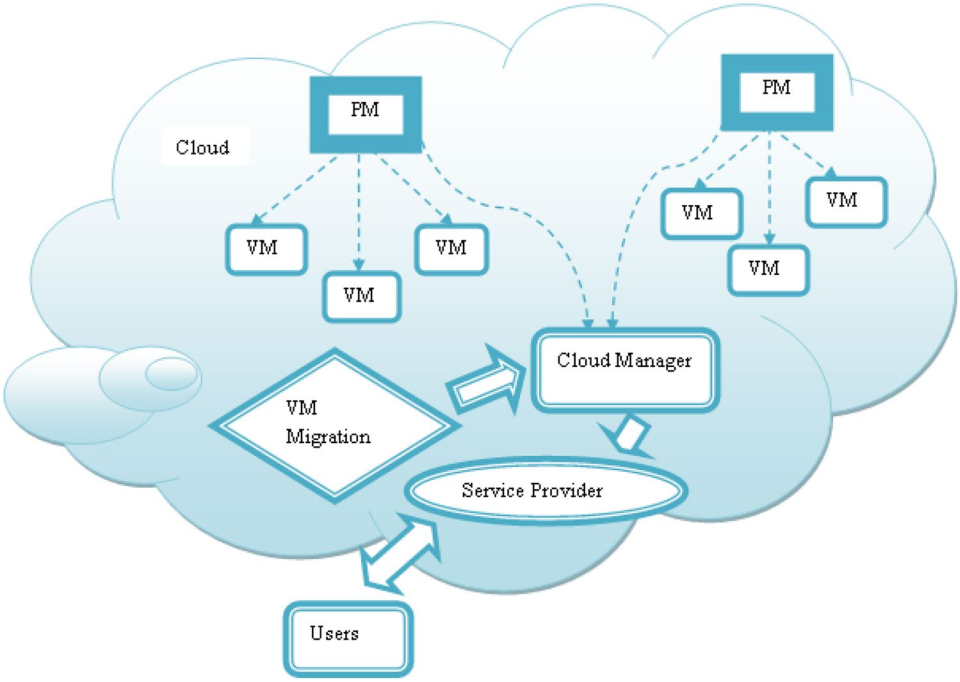

Fig. 1 VM migration model in the Cloud computing environment

4. Subsequently, compute the load value of the VM based on Eq. (2), and if the load value exceeds the value of the threshold, migrate the VM optimally using the proposed ChicWhale.
5. Find the other parameters, like resource availability, energy, and migration cost for the system.
6. Finally, the steps from (3) to (5) are repeated for every iteration, and the algorithm is terminated at the end of the iteration.

3.2 Virtual machine migration using chicken whale optimization algorithm

This section presents the developed VM migration model with the proposed ChicWhale optimization algorithm in the cloud computing environment, and the block diagram is shown in Fig. 2. The proposed ChicWhale optimization algorithm is designed by combining the CSO and WOA to select the optimal VM. The proposed ChicWhale optimization algorithm validates the load status based on each VM contained in the cloud and reallocated the task to another VM if the load of VM exceeds a specific threshold. Then, the proposed method defines the new fitness function based on various parameters like migration cost, load, resource availability, and energy. The proposed model assigns the appropriate tasks to VM in terms of fitness measure.

3.2.1 Initialization

Assume the cloud model consists of several numbers of PMs, and VMs. The cloud model contains P number of PMs, which are denoted as, $P = \{P_1, P_2, \ldots, P_n, \ldots P_N\}; 1 \le n \le N$, and under every PM, several VMs are present. Let us consider the VM available in n^{th} PM are represented as, $V = \{V_1^n, V_2^n, \ldots, V_m^n, \ldots V_M^n\}; 1 \le m \le M$, where, the symbol V_m^n denotes the m^{th} VM in n^{th} PM. In addition, the task requested from the individual user is assigned to each VM in a round-robin manner and is expressed as,$R = \left\{R_1, R_2, \ldots, R_u, \ldots, R_v\right\}$, where v specifies the total number of tasks assigned to each VM. The VM in the cloud model poses various parameters, which include the number of processing entities, CPU, bandwidth, memory, and Million Instructions per Second (MIPS). The mth VM present in nth PM in a cloud computing environment poses the following aspects which are expressed by

$$V_m^n = \left\{J_m^n, C_m^n, B_m^n, D_n^m, I_n^m\right\} \tag{1}$$

where J_m^n signifies the total number of processing entities of mth VM in nth PM, and the term C_m^n represents the number of CPUs used by mth VM in nth PM. The symbols B_m^n refer to the bandwidth and D_n^m signify the memory of mth VM in nth PM. The total number of MIPS utilized by mth VM in nth PM is denoted as I_n^m. All the above mentioned parameters are acquired, the value ranges from 1 *to* 10.

3.2.2 Load computation

The load is computed using the resources employed by the VM for processing tasks obtained from the user. Network bandwidth**,** MIPS, memory, CPU, and the total number of processing entities are utilized for evaluating the load in the

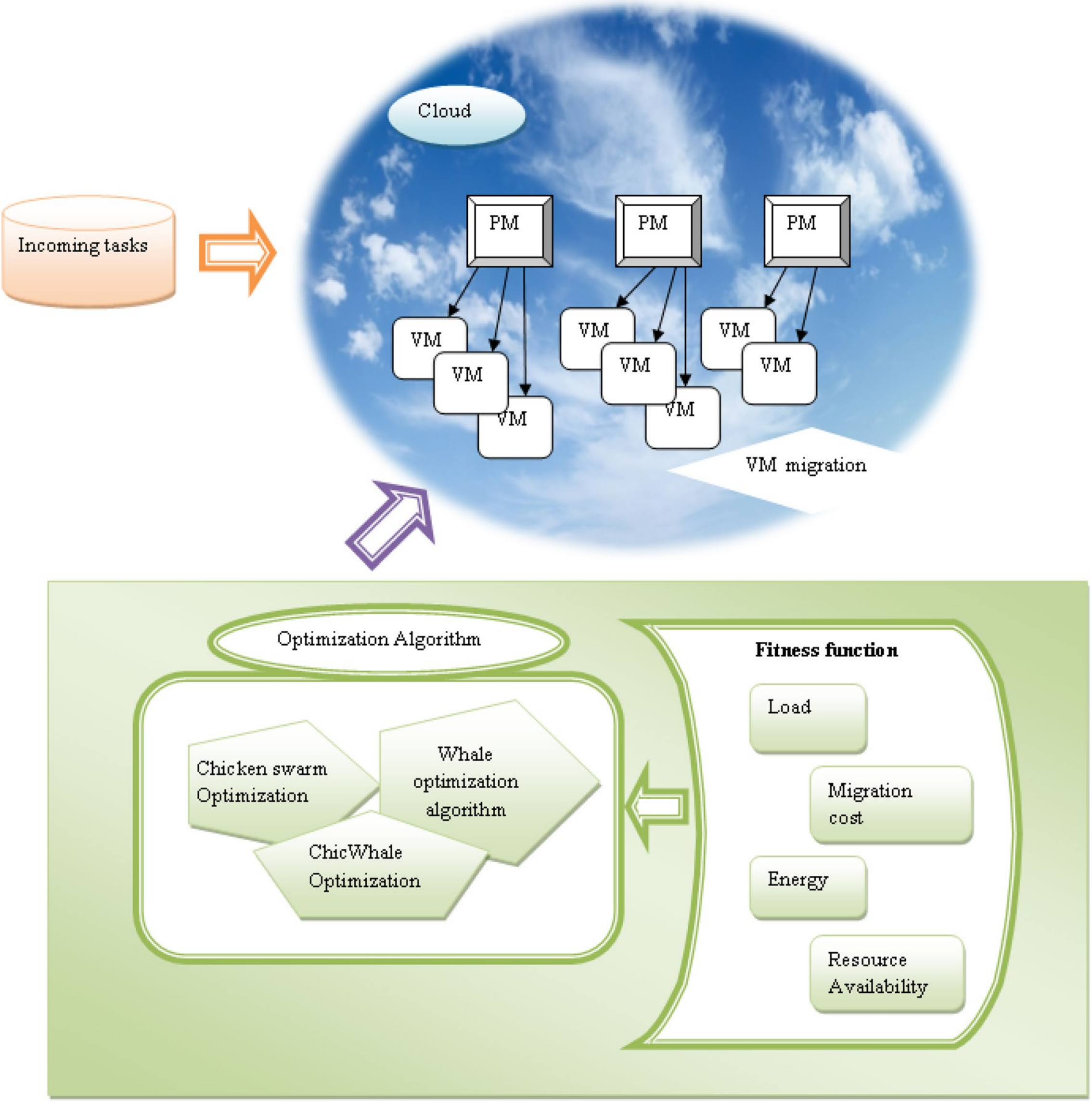

Fig. 2 Block diagram of the proposed ChicWhale-based VM migration scheme in the cloud computing platform

cloud. The load is computed with memory, CPU utilization, bandwidth, frequency, and MIPS, as expressed below,

$$Load\,(L) = \frac{R_U}{t} \tag{2}$$

where the resource utilization is denoted as R_U, and the term t refers to the time. The expression for resource utilized by mth VM present is given by,

where the total number of VMs in each PM is denoted as m, and the term J^F refers to the total amount of processing entities. The total amount of CPUs utilized in PM is represented as C^F, and the bandwidth employed in PM is represented as B^F. The terms D^F, and I^F signifies the memory and MIPS in PM. The normalizing factor is given as F.

$$R_U = \frac{1}{F}\sum_{i-1}^{M}\left(\frac{J^F}{\max\left(J^F\right)} + \frac{C^F}{\max\left(C^F\right)} + \frac{B^F}{\max\left(B^F\right)} + \frac{D^F}{\max\left(D^F\right)} + \frac{I^F}{\max\left(I^F\right)}\right) \tag{3}$$

3.2.3 Resource availability

It is employed to make sure the resource utilization. It must be optimized for well-organized load balancing. The resource availability is expressed as,

$$R_A = 1 - R_U \tag{4}$$

3.2.4 Migration cost

The migration cost of VM is based on the number of movements attained to the total movements performed in the cloud. The formula for migration cost in the whole cloud environment is given by,

$$\mathrm{M}_c = \frac{1}{N}\sum_{j=1}^{n}\left(\frac{G}{c * M}\right) \tag{5}$$

where, the total number of PMs, and the term G indicates the number of migration of VMs. The total number of VMs is denoted as M, and c indicates the constant.

3.2.5 Energy model

The energy is mainly based on power utilized for migration and data processing by each VM. Therefore, the energy of the cloud setup is directly based on the consumed power from various resources present in VM. The energy model of VM is given below,

$$E = \frac{1}{T}\sum_{t=1}^{T} K \tag{6}$$

where T signifies the total time period, and the power consumed in the VM of the cloud is denoted as K. The power consumption is expressed by,

$$K = p * K_{\max} + (1 - p) * K_{\max} * R_U^{cloud}; \quad 1 < p < t < T \tag{7}$$

where, the term $K_{\max}$ denotes the maximum power consumed, and R_U^{cloud} represents the resource utilization in the cloud, and is expressed by,

$$R_U^{cloud} = \frac{1}{N}\sum_{r=1}^{N} R_U \tag{8}$$

where, the total number of PMs is denoted as N.

3.3 Proposed ChicWhale optimization for the migration of VM

In this section, the VM migration approach is elaborated using the proposed ChicWhale optimization approach to migrate the VM in the cloud computing platform. The proposed ChicWhale is designed by combining WOA in CSO. Thus, the integration of WOA in CSO tunes the associated parameters for improving the algorithmic performance to obtain a globally optimum solution. WOA [45] is duly based on the hunting mechanism of humpback whales that searches for their prey based on the bubble-net attacking mechanism. It is worth noting that the search for the prey is both associated within or outer the search spaces through a series of steps, like encircling, exploitation, and exploration. The CSO [46] algorithm is inspired by the behaviour of chicken swarms. The chicken swarms can easily extract swarm intelligence for solving the optimization issues and can attain improved optimization accuracy and robustness. Moreover, the CSO algorithm provides an effective trade-off between the determinacy and randomness to find the optima. Thus, the CSO acts intelligently for optimizing the issues and poses the self-adaptive ability to address the optimization issues. The solution encoding, fitness function, and the algorithm of the proposed ChicWhale optimization approach are explained below.

1. **Solution encoding**

The proposed ChicWhale optimization approach selects a suitable VM for migration. Let us assume that the PM 1 consists of 3 VMs and PM 2 having 2 VMs. The incoming tasks assigned to each VM using round-robin fashion. In this method, time slices are assigned to every process, and the process has to be carried out in a circular order, also all the processes are handled without priority. If the VM exceeds the threshold value, then the VM is migrated to the underloaded PMs, and the optimal VM for the migration is assigned using the optimization algorithm. Figure 3 depicts the solution encoding of the developed method to identify the optimal VM for migration. The first task in VM 1 is completed within the threshold value, the second and third tasks are not completed within the threshold value. Hence, the execution

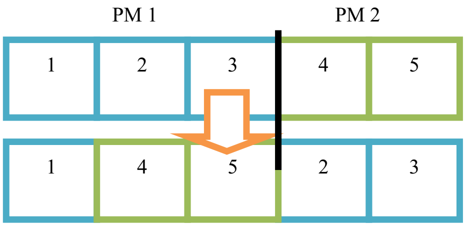

Fig. 3 Solution encoding

of tasks two and three has been postponded and the tasks four and five have been executed. The execution of tasks four and five are completed within the threshold value. Afterthat, the execution of tasks two and three has been resumed. Here, the solution represents the VM, denoted as $S = \{S_1, S_2, \ldots, S_K\}$, where K denotes the total number of underloaded VMs.

2. **Fitness evaluation**

The fitness function is computed for determining the best solution from the solution set. The fitness function of the ChicWhale optimization approach is formulated in terms of four parameters, like load, energy, and so on. The fitness is a maximization function, and thus the optimal VM is chosen for execution. The fitness function of the proposed ChicWhale optimization approach is expressed by,

$$Fitness(f) = \alpha R_A + \beta(1 - L) + \gamma\left(1 - \mathrm{M}_c\right) + \delta(1 - E) \tag{9}$$

where, the terms α,β,γ, and δ are the weights ranging from $0\ to\ 1$. The resource utilization is denoted as R_A, and the load is represented as L. The symbol M_C signifies the migration cost, and E refers to the energy.

3. **Algorithmic procedure of the proposed ChicWhale optimization approach**

The algorithmic steps of the proposed model to select the best VM for migration is illustrated below,

(a) *Initialization* In the initial step, the parameters of the optimization including the population are initialized, which includes:$\{\mathrm{X}_{gh},\ 1 \le g \le a;\ 1 \le h \le k\}$, where, a refer to the population size, and the dimension is indicated as k. $\mathrm{X} \in \{\ \alpha,\beta,\gamma,\delta\ \}$.

(b) *Evaluation of fitness function* The fitness is computed for each solution on the basis of the fitness function depicted in Eq. (9). The fitness function is taken as the maximization function, and a solution producing the maximum fitness is considered as the best solution.

(c) *Movement of rooster* Rooster with the best fitness value may search their food in a wider range than that of worst fitness values. Hence, the behavior of the rooster movement is expressed as,

$$\mathrm{X}_{gh}(\tau + 1) = \mathrm{X}_{gh}(\tau) * \left(1 + rand(0, \sigma^2)\right) \tag{10}$$

$$\sigma^2 = \begin{cases} 1; & if\ q_g \le q_s \\ \exp\left(\frac{q_s - q_g}{|q_g| + \in}\right); & otherwise, \end{cases} s \in [1, a],\ s \ne g \tag{11}$$

where $\mathrm{X}_{gh}(\tau + 1)$ be the new location of the rooster $\mathrm{X}(\tau)$ at iteration $\tau + 1$, and $\mathrm{X}_{gh}(\tau)$ represents the current location of the rooster at iteration τ. The term $rand(0, \sigma^2)$ represents the random distribution with standard deviation σ^2, and mean 0. The randomly selected rooster is denoted as s, and q_g be the fitness value of related rooster X_g.

(d) *Movement of hen* Hens followed their group-mate for foraging. Additionally, they steal their food in a random manner, which is found by other chicken, though they are repressed by another chicken. The other leading hens may have benefit in competing for the food than more submissive ones. The standard equation of the foraging behavior of hens is given by,

$$\begin{aligned}\mathrm{X}_{gh}(\tau + 1) =& \mathrm{X}_{gh}(\tau) + W_1 * rand\big(\mathrm{X}_{b1,h}(\tau) \\ & -\mathrm{X}_{gh}(\tau) + W_2 * rand(\mathrm{X}_{b2,h}(\tau) - \mathrm{X}_{gh}(\tau)\big)\end{aligned} \tag{12}$$

where,

$$W_1 = \frac{\exp\left(q_g - q_{b1}\right)}{abs\left(q_g\right) + \in}$$

$$U_2 = \exp(q_{b2} - q_g)$$

where $rand$ denotes the random number ranges from $0\ and\ 1$. $b1$ be the index of the rooster, and $b2$ denotes the index of chicken (hen or rooster) that are randomly selected from the swarm. Here, $b1 \ne b2$. j be the fibers.

The update equation of bypass WOA is expressed as,

$$\mathrm{X}_{gh}(\tau + 1) = \mathrm{X}^*(\tau) - H.S \tag{13}$$

Assuming $\left(\mathrm{X}^*(\tau) > \mathrm{X}(\tau)\right)$,

$$\mathrm{X}_{gh}(\tau + 1) = \mathrm{X}^*(\tau) - H\left(Y\mathrm{X}^*(\tau) - \mathrm{X}_{gh}(\tau)\right) \tag{14}$$

$$\mathrm{X}_{gh}(\tau + 1) = \mathrm{X}^*(\tau) - HY\mathrm{X}^*(\tau) + H\mathrm{X}_{gh}(\tau) \tag{15}$$

$$\mathrm{X}_{gh}(\tau) = \frac{\mathrm{X}_{gh}(\tau + 1) - \mathrm{X}^*(\tau) + HY\mathrm{X}^*(\tau)}{H} \tag{16}$$

where, the coefficient vectors are denoted as H and S, $\mathrm{X}(\tau)$ refer to the location vector, $\mathrm{X}^*(\tau)$ is the current better position, The coefficient vectors are expressed as,

$$H = 2d\,.\,b - d \tag{17}$$

$$S = 2\,.\,b \tag{18}$$

where d value decreased linearly between $2\ to\ 0$, and the b is the random vector value ranges from $0\ and\ 1$.

Rearranging Eq. (12),

$$\begin{aligned} \mathrm{X}_{gh}(\tau+1) =& \mathrm{X}_{gh}(\tau) + W_1 * rand\mathrm{X}_{b1,h}(\tau) \\ & - W_1 * rand\mathrm{X}_{gh}(\tau) + W_2 * rand\mathrm{X}_{b2,h}(\tau) \\ & - W_2 * rand\,\mathrm{X}_{gh}(\tau) \end{aligned} \tag{19}$$

$$\begin{aligned} \mathrm{X}_{gh}(\tau+1) =& \mathrm{X}_{gh}(\tau)\big(1 - W_1 * rand - W_2 * rand\big) \\ & + W_1 * rand\mathrm{X}_{b1,h}(\tau) + W_2 * rand\,\mathrm{X}_{b2,h}(\tau) \end{aligned} \tag{20}$$

Substituting Eq. (16) in Eq. (20),

$$\begin{aligned} \mathrm{X}_{gh}(\tau+1) =& \left[\frac{\mathrm{X}_{gh}(\tau+1)}{H} - \frac{-\mathrm{X}^*(\tau) - HY\mathrm{X}^*(\tau)}{H}\right] \\ & \big(1 - W_1 * rand - W_2 * rand\big) + W_1 * rand\mathrm{X}_{b1,h}(\tau) \\ & + W_2 * rand\,\mathrm{X}_{b2,h}(\tau) \end{aligned} \tag{21}$$

$$\begin{aligned} \mathrm{X}_{gh}(\tau+1) =& \frac{\mathrm{X}_{gh}(\tau+1)}{H}\big(1 - W_1 * rand - W_2 * rand\big) - \frac{\mathrm{X}^*(\tau) - HY\mathrm{X}^*(\tau)}{H} \\ & \big(1 - W_1 * rand - W_2 * rand\big) + W_1 * rand\mathrm{X}_{b1,h}(\tau) + W_2 * rand\,\mathrm{X}_{b2,h}(\tau) \end{aligned} \tag{22}$$

$$\begin{aligned} & \mathrm{X}_{gh}(\tau+1) - \frac{\mathrm{X}_{gh}(\tau+1)}{H}\big(1 - W_1 * rand - W_2 * rand\big) = W_1 * rand\mathrm{X}_{b1,h}(\tau) \\ & + W_2 * rand\,\mathrm{X}_{b2,h}(\tau) - \frac{\mathrm{X}^*(\tau) - HY\mathrm{X}^*(\tau)}{H}\big(1 - W_1 * rand - W_2 * rand\big) \end{aligned} \tag{23}$$

$$\begin{aligned} & \mathrm{X}_{gh}(\tau+1)\left(\frac{H - 1 + W_1 * rand + W_2 * rand}{H}\right) = W_1 * rand\mathrm{X}_{b1,h}(\tau) \\ & + W_2 * rand\,\mathrm{X}_{b2,h}(\tau) - \frac{\mathrm{X}^*(\tau) - HY\mathrm{X}^*(\tau)}{H}\big(1 - W_1 * rand - W_2 * rand\big) \end{aligned} \tag{24}$$

$$\mathrm{X}_{gh}(\tau+1) = \frac{H}{H - 1 + W_1 * rand + W_2 * rand}\left[\begin{array}{l} W_1 * rand\mathrm{X}_{b1,h}(\tau) + W_2 * rand\,\mathrm{X}_{b2,h}(\tau) \\ -\dfrac{\mathrm{X}^*(\tau)(1 - HY)}{H}\big(1 - W_1 * rand - W_2 * rand\big) \end{array}\right] \tag{25}$$

(e) *Movement of chick* The chicks move around their mother for searching the food, and is expressed by,

$$\mathrm{X}_{gh}(\tau+1) = \mathrm{X}_{gh}(\tau) + OZ * \big(\mathrm{X}_{eh}(\tau) - \mathrm{X}_{gh}(\tau)\big) \tag{26}$$

where $\mathrm{X}_{eh}(\tau)$ denotes the location of h^{th} chick's mother. OZ represents the speed of the chick that follows the mother, and is selected randomly in range $[0, 2]$.

(f) *Checking the feasibility of solution* The feasibility of the solution is computed based on the fitness function. If the newly generated solution is best than the previous one, then it is changed by the new solution.

(g) *Termination* Repeat the steps for the maximal iterations until the global optimal best solutions are determined. Thus, the optimization algorithm discussed in this section aims at determining the optimal weights for VM migration.

The pseudo-code of the developed ChicWhale optimization is depicted in Algorithm 1, which demonstrates the step-wise description of the algorithm.

Algorithm1. Pseudocode for the proposed ChicWhale algorithm

```
Input: Chicken swarm population $\{X_{gh}, 1 \le g \le a; 1 \le h \le k\}$
Output: Best solution
Procedure:
Begin
    Population initiation: $\{X_{gh}, 1 \le g \le a; 1 \le h \le k\}$ , and read all the parameters
    Calculate the fitness values for all chickens, at $\tau = 0$, and arranged the swarm and mother child relationship in hierarchical order.
    While $\tau < \max gen$
        $\tau = \tau + 1$
        If $(\tau \% \max gen = 0)$
            Establish the hierarchical order in swarm and mother child relationship
        Else
            for $g = 1 : a$
                If $g = rooster$
                    Update the rooster based on equation (10)
                End if
                If $g = hen$
                    Update the hen based on equation (25)
                End if
                If $g = chick$
                    Update the chick based on equation (26)
                End if
            End for
        Check the feasibility of solutions
        Return the best solution
        $\tau = \tau + 1$
    End while
    Optimal solution is obtained
End
```

The flowchart for the proposed ChicWhale Optimization algorithm is given Fig. 4.

4 Results and discussion

The section demonstrates the results achieved by the developed ChicWhale model, and the performance is evaluated with respect to the previous methods based on varying the number of iteration.

4.1 Experimental arrangement

The experimentation of the developed model is done in JAVA, operating in the PC with Windows 10 OS and 2 GB RAM.

4.2 Simulation setup

The cloud environment employed for simulation contains 10 PMs and 50VMs, and the total incoming tasks vary as 25, and 75. Table 2 depicts the simulation setup of the proposed system.

4.3 Performance metrics

The metrics utilized for the analysis include load, migration cost, energy consumption, and resource availability. These are the commonly used performance metrics in the existing VM migration methods in cloud computing platform. The derivation of the energy consumption, migration cost, resource availability, and the load is defined in Eqs. (6), (5), (4), and (2). Table 3 shows the definition of the performance metrics used in this paper.

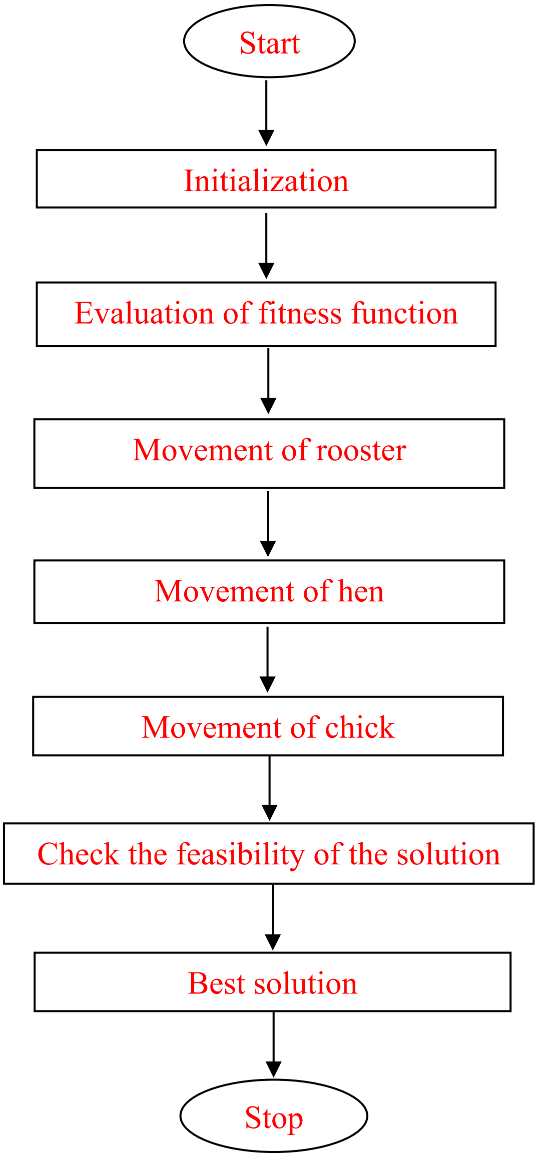

Fig. 4 Flowchart of the proposed ChicWhale optimization algorithm

4.4 Comparative techniques

The methods, such as Artificial Bee Colony-Bat Algorithm (ABC-BA) [34], Firefly [37], WOA [45] are used for the comparison with the proposed ChicWhale for the analysis.

Table 2 Simulation setup

Parameter	Value
Number of PMs	10
Number of VMs	50
Total incoming tasks	25—75
VM memory	1 GB
VM bandwidth	1 GB

4.4.1 Comparative analysis with incoming task = 25

The comparative analysis of the developed method is analyzed based on load, migration cost, energy consumption, and resource availability with incoming task = 25 are depicted in tables. Table 4 illustrates the analysis based on load by varying the number of iterations. When the number of iteration is 50, then the corresponding load values computed by existing ABC-BA, Firefly, WOA, and proposed ChicWhale are 0.0098, 0.0038, 0.0029, and 0.0019, respectively. For 80th iteration, the existing techniques, like ABC-BA, Firefly, and WOA, possess the load of 0.0046, 0.0019 and 0.0016, respectively, which is comparatively higher than the ChicWhale. For the same iteration, the developed ChicWhale acquired the load value of 0.0015. Similarly, when the number of iteration increased to 90, the methods, ABC-BA, Firefly, and WOA, attained the load of 0.0016, 0.0010, and 0.0003, whereas the load of the developed method is 0.0002.

The comparative analysis based on migration cost is shown in Table 5. When the number of iteration is 60, the migration cost values achieved by ABC-BA, Firefly, WOA, and proposed model are 0.195, 0.132, 0.0605, and 0.059, respectively. For 70th iteration, the existing techniques, like ABC-BA, Firefly, and WOA, possesses the migration cost of 0.130, 0.117, and 0.103, respectively, which is comparatively higher than the proposed ChicWhale. For the same number of iteration, the developed ChicWhale acquired the migration cost of 0.085. Similarly, when the number of iteration increased to 80, the methods, ABC-BA, Firefly, and WOA, attained the migration cost of 0.159, 0.127, and 0.108, whereas the migration cost of the developed method is 0.091. Similarly, for 90th iteration, the migration cost values measured by ABC-BA, Firefly, and WOA, and proposed model are 0.077, 0.0693, 0.068, and 0.059.

Table 3 Performance metrics and definition

Performance metrics	Definition
Load	The load is computed using the resources employed by the VM for processing tasks obtained from the user
Migration cost	The migration cost of VM is based on the number of movements attained to the total movements performed in the cloud
Energy consumption	The energy is mainly based on power utilized for migration and data processing by each VM
Resource availability	It is employed to make sure the resource utilization

Table 4 Comparative analysis of Load when task = 25

Methods/ iterations	ABC-BA	Firefly	WOA	Proposed ChicWhale
50	0.0098	0.0038	0.0029	0.0019
60	0.0105	0.0042	0.0022	0.001
70	0.0057	0.0035	0.0017	0.0001
80	0.0046	0.0019	0.0016	0.0015
90	0.0016	0.001	0.0003	0.0002

Table 5 Comparative analysis of Migration cost when task = 25

Methods/ iterations	ABC-BA	Firefly	WOA	Proposed ChicWhale
50	0.1676	0.1034	0.0863	0.0565
60	0.1958	0.1326	0.0605	0.0594
70	0.1307	0.117	0.1036	0.0852
80	0.1597	0.1277	0.108	0.0917
90	0.0777	0.0693	0.0681	0.0592

Table 6 Comparative analysis of Energy consumption when task = 25

Methods/ iterations	ABC-BA	Firefly	WOA	Proposed ChicWhale
50	0.4977	0.496	0.4947	0.4872
60	0.4981	0.4944	0.4942	0.4874
70	0.498	0.4951	0.4947	0.4945
80	0.4978	0.4969	0.4967	0.4937
90	0.4986	0.4984	0.4971	0.4966

Table 7 Comparative analysis of Resource availability when task = 25

Methods/ iterations	ABC-BA	Firefly	WOA	Proposed ChicWhale
50	0.9377	0.951	0.9599	0.9739
60	0.9289	0.9441	0.957	0.9898
70	0.9537	0.958	0.9715	0.9819
80	0.9618	0.9676	0.9716	0.9812
90	0.9426	0.9659	0.9728	0.9819

Table 8 Comparative analysis of Load when task = 75

Methods/ iterations	ABC-BA	Firefly	WOA	Proposed ChicWhale
50	0.0228	0.0199	0.0133	0.0098
60	0.0187	0.0165	0.0109	0.0083
70	0.0131	0.0116	0.0081	0.0052
80	0.0108	0.0089	0.0064	0.0047
90	0.0129	0.0085	0.0062	0.0057

Table 9 Comparative analysis of Migration cost when task = 75

Methods/ iterations	ABC-BA	Firefly	WOA	Proposed ChicWhale
50	0.1035	0.0987	0.0708	0.0629
60	0.1555	0.1436	0.1064	0.0843
70	0.1016	0.0729	0.0657	0.0564
80	0.1129	0.1108	0.1059	0.0723
90	0.0998	0.0817	0.0678	0.0591

The analysis in terms of energy cost is depicted in Table 6. For 60th iteration, the existing techniques, like ABC-BA, Firefly, and WOA, possesses the energy cost of 0.498, 0.494, and 0.494, respectively, which is comparatively higher than the ChicWhale. For the same iteration, the developed ChicWhale acquired the energy cost of 0.487. Similarly, when the number of iteration increased to 70, the methods, ABC-BA, Firefly, and WOA, attained the energy cost of 0.498, 0.495, and 0.494, whereas the energy cost of the developed method is 0.494. When the number of iteration is 80, the energy cost values measured by ABC-BA, Firefly, WOA, and proposed model are 0.497, 0.496, 0.496, and 0.493, respectively. When the number of iteration is kept to 90, the energy cost values measured by ABC-BA, Firefly, and WOA, and proposed model are 0.498, 0.498, 0.497, and 0.496, respectively.

Table 7 depicts the analysis in terms of resource availability by varying the number of iterations. For 50th iteration, the existing techniques, like ABC-BA, Firefly, and WOA, possesses the resource availability of 0.937, 0.951, and 0.959, respectively, which is comparatively lower than the ChicWhale. For the same iteration, the developed ChicWhale acquired the resource availability of 0.973. Similarly, when the number of iteration increased to 70, the methods, ABC-BA, Firefly, and WOA, attained the resource availability of 0.953, 0.958, and 0.971, whereas the resource availability of the developed method is 0.981. When 80th iteration is considered, the methods, like ABC-BA, Firefly, and WOA acquire the resource availability value of 0.961, 0.967, and 0.971. Meanwhile, the developed ChicWhale obtained the resource availability value of 0.981. For 90th iteration, the resource availability values measured by ABC-BA, Firefly, and WOA are 0.942, 0.965, and 0.972, whereas the proposed ChicWhale achieved the resource availability of 0.981.

Table 10 Comparative analysis of Energy consumption when task = 75

Methods/ iterations	ABC-BA	Firefly	WOA	Proposed ChicWhale
50	0.4875	0.4867	0.4846	0.4805
60	0.4906	0.4863	0.4862	0.4826
70	0.4915	0.4902	0.4879	0.4839
80	0.4939	0.4921	0.4912	0.4866
90	0.4936	0.4922	0.4907	0.4832

4.4.2 Comparative analysis with incoming task = 75

The comparative analysis of the developed method is analyzed in terms of load, migration cost, energy consumption, and resource availability with incoming task = 75 are depicted in tables. Table 8 illustrates the analysis based on load by varying the number of iterations. When the number of iteration is 50, then the corresponding load values computed by existing ABC-BA, Firefly, WOA, and proposed ChicWhale are 0.022, 0.019, 0.013, and 0.009, respectively. For 80th iteration, the existing techniques, like ABC-BA, Firefly, and WOA, possess the load of 0.010, 0.008, and 0.006, respectively, which is comparatively higher than the ChicWhale. For the same iteration, the developed ChicWhale acquired the load value of 0.004. Similarly, when the number of iteration increased to 90, the methods, ABC-BA, Firefly, and WOA, attained the load of 0.012, 0.008, and 0.006, whereas the load of the developed method is 0.005.

The comparative analysis based on migration cost is shown in Table 9. When the number of iteration is 60, the migration cost values achieved by ABC-BA, Firefly, WOA, and proposed model are 0.155, 0.143, 0.106, and 0.084, respectively. For 70th iteration, the existing techniques, like ABC-BA, Firefly, and WOA, possesses the migration cost of 0.101, 0.072, and 0.065, which is comparatively higher than the proposed ChicWhale. For the same number of iteration, the developed ChicWhale acquired the migration cost of 0.056. Similarly, when the number of iteration increased to 80, the methods, ABC-BA, Firefly, and WOA, attained the migration cost of 0.112, 0.110, and 0.105, whereas the migration cost of the developed method is 0.073. Similarly, for 90th iteration, the migration cost values measured by ABC-BA, Firefly, and WOA, and proposed model are 0.099, 0.081, 0.067, and 0.059.

The analysis in terms of energy cost is depicted in Table 10. For 60th iteration, the existing techniques, like ABC-BA, Firefly, and WOA, possesses the energy cost of 0.490, 0.486, and 0.486, respectively, which is comparatively higher than the ChicWhale. For the same iteration, the developed ChicWhale acquired the energy cost of 0.482. Similarly, when the number of iteration increased to 70, the methods, ABC-BA, Firefly, and WOA, attained the energy cost of 0.491, 0.490, and 0.487, whereas the energy cost of the developed method is 0.484. When the number of iteration is 80, the energy cost values measured by ABC-BA, Firefly, WOA, and proposed model are 0.493, 0.492, 0.491, and 0.486, respectively. When the number of iteration is kept to 90, the energy cost values measured by ABC-BA, Firefly, and WOA, and proposed model are 0.493, 0.492, 0.490, and 0.483, respectively.

Table 11 depicts the analysis in terms of resource availability by varying the number of iterations. For 50th iteration, the existing techniques, like ABC-BA, Firefly, and WOA, possesses the resource availability of 0.758, 0.760, and 0.777, respectively, which is comparatively lower than the ChicWhale. For the same iteration, the developed ChicWhale acquired the resource availability of 0.792. Similarly, when the number of iteration increased to 70, the methods, ABC-BA, Firefly, and WOA, attained the resource availability of 0.796, 0.852, and 0.855, whereas the resource availability of the developed method is 0.882. When 80th iteration is considered, the methods, like ABC-BA, Firefly, and WOA acquire the resource availability value of 0.841, 0.843, and 0.866. Meanwhile, the proposed ChicWhale obtained the resource availability value of 0.893. For 90th iteration, the resource availability values measured by ABC-BA, Firefly, and WOA are 0.789, 0.856, and 0.872, whereas the proposed ChicWhale achieved the resource availability of 0.892.

Table 11 Comparative analysis of Resource availability when task = 75

Methods/ iterations	ABC-BA	Firefly	WOA	Proposed ChicWhale
50	0.7588	0.7606	0.7776	0.7927
60	0.7709	0.7937	0.8047	0.8343
70	0.7969	0.8522	0.8552	0.882
80	0.8414	0.8438	0.8661	0.8931
90	0.7899	0.8564	0.8723	0.8929

Table 12 Comparative analysis

Methods	Load	Migration cost	Energy consumption	Resource availability
ABC-BA	0.0016	0.0777	0.487	0.961
Firefly	0.0010	0.069	0.486	0.967
WOA	0.0003	0.0605	0.484	0.972
Proposed ChicWhale	0.0001	0.0564	0.481	0.989

4.5 Comparative discussion

Table 12 depicts the comparative discussion of previous ABC-BA, Firefly, and WOA and the developed ChicWhale in terms of load, energy consumption, migration cost, and resource availability parameters with incoming tasks 25, and 75 by altering the number of iterations The minimum performance measured by proposed ChicWhale in terms of load parameter is 0.0001, whereas the load values of existing ABC-BA, Firefly, and WOA are 0.0016, 0.0010, and 0.0003, respectively. The minimal migration cost achieved by the proposed ChicWhale with a value of 0.0564, whereas the existing ABC-BA, Firefly, and WOA acquired the migration cost of 0.0777, 0.069, and 0.0605. The minimal energy consumption value computed by the proposed ChicWhale is 0.481, whereas the existing ABC-BA, Firefly, and WOA methods acquired the energy consumption of 0.487, 0.486, and 0.484. Meanwhile, the proposed ChicWhale has achieved the maximal resource availability of 0.989, whereas the existing ABC-BA, Firefly, and WOA have attained the resource availability of 0.961, 0.967, and 0.972. Among the existing methods, the WOA gives better performance results for the metrics, such as load, migration cost, energy consumption, and resource availability. The performance of the proposed ChicWhale algorithm is 66.66%, 6.77%, 0.62%, and 1.72% better than the performance of the existing WOA algorithm in terms of metrics, such as load, migration cost, energy consumption, and resource availability.

5 Conclusion

This research paper presents the VM migration strategy using the novel optimization algorithm in the cloud computing model. The developed model employs the multi-objective framework designed from various factors, like load, resource availability, energy consumed, migration cost, and power consumption. Accordingly, the searching criteria are computed to identify the best service based on constraints. The VM migration model employs the developed ChicWhale optimization approach to assign VM. The developed ChicWhale optimization is the nature-inspired algorithm that depends on both the properties of CSO and WOA. The simulation of developed VM migration strategy with ChicWhale optimization algorithm utilized the cloud model with 10 PM as well as 50 VM. The performance of the ChicWhale is measured in terms of load, migration cost, energy consumption, and resource availability. The developed ChicWhale produces the minimal load of 0.0001, minimal migration cost of 0.0564, the minimal energy consumption of 0.481, and the maximal resource availability of 0.989, which indicates the superiority of the proposed method. The proposed algorithm can be used in Medical Image Quality Enhancement System, Optimal Allocation of Water Resources, and Mechanical and Production Engineering fields. In the future, we will integrate the technology like redundancy elimination to reduce the overhead.

References

1. Huang W, Ma Z, Dai X, Mingdi Xu, Gao Yi (2018) Fuzzy clustering with feature weight preferences for load balancing in cloud. Int J Softw Eng Knowl Eng 28(5):593–617
2. Jansen W, Grance T (2011) Guidelines on security and privacy in public cloud computing, pp 800–144
3. Mateusz G, Alicja G, Pascal B (2015) Cloud brokering: current practices and upcoming challenges. IEEE Cloud Comput 2(2):40–47
4. Nayyar A (2019) Handbook of cloud computing: basic to advance research on the concepts and design of cloud computing. BPB Publication, Noida
5. Milani AS, Navimipour NJ (2016) Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends. J Netw Comput Appl 71(1):86–98
6. Nayyar A (2011) Private virtual infrastructure (PVI) model for cloud computing. Int J Softw Eng Res Pract 1(1):10–14
7. Jiang YC (2016) A survey of task allocation and load balancing in distributed systems. IEEE Trans Parallel Distrib Syst 27(2):585–599
8. Ning J, Cao Z, Dong X, Liang K, Ma H, Wei L (2017) Auditable-time outsourced attribute-based encryption for access control in cloud computing. IEEE Trans Inf Forensics Secur 13:94–105
9. Nayyar A, Puri V, Suseendran G (2018) Artificial Bee Colony optimization—population-based meta-heuristic swarm intelligence technique. In: Proceedings of ICDMAI, vol 2, pp 513–525
10. Nayyar A, Le D-N, Nguyen NG (2018) Advances in swarm intelligence for optimizing problems in computer science. CRC Press, Boca Raton
11. Nayyar A, Le D-N, Nguyen NG (2018) Advances in swarm intelligence for optimizing problems in computer science. In: Evolutionary computation
12. Nayyar A, Le D-N, Nguyen NG (2018) Advances in swarm intelligence for optimizing problems in computer science. In: Introduction to swarm intelligence
13. Kaur A, Gupta P, Singh M, Nayyar A (2019) Data placement in era of cloud computing: a survey, taxonomy and open research issues. Scalable Comput Pract Exp 20(2):377–398
14. Durbhaka GK, Selvaraj B, Nayyar A (2018) Firefly swarm: metaheuristic swarm intelligence technique for mathematical optimization. In: Data management, analytics and innovation, pp 457–466
15. Nayyar A, Singh R (2016) Ant colony optimization—computational swarm intelligence technique. In: 3rd International conference on computing for sustainable global development (INDIACom), New Delhi, India
16. Xue W, Li W, Qi H, Li K, Tao X, Ji X (2017) Communication-aware virtual machine migration in cloud data centres. Int J High Perform Comput Netw 10(4/5):372–380
17. Nelson M, Lim B-H, Hutchins G (2005) Fast transparent migration for virtual machines. In: USENIX annual technical conference, general track, pp 391–394
18. Xiao Z, Song W, Chen Q (2013) Dynamic resource allocation using virtual machines for cloud computing environment. IEEE Trans Parallel Distrib Syst 24(6):1107–1117
19. Xu X (2012) From cloud computing to cloud manufacturing. Robot Comput-Integr Manuf 28(1):75–86

20. Nipanikar SI, Hima Deepthi V (2019) Enhanced whale optimization algorithm and wavelet transform for image steganography 2(3):23–32
21. Cheng L, Li T (2016) Efficient data redistribution to speed up big data analytics in large systems. In: Proceedings of IEEE 23rd international conference on high performance computing (HiPC), pp 91–100
22. Zhang F, Liu G, Xiaoming F, Yahyapour R (2018) A survey on virtual machine migration: challenges, techniques and open issues. IEEE Commun Surv Tutor 20(2):1206–1243
23. Petruccelli U, Antonello R (2019) Assessment of the drivers number as a tool for improving efficiency of public transport services. Ing Ferrov 74(4):295–315
24. Petruccelli U (2011) La qualita' percepita nel trasporto pubblico locale: un modello multicriteri per la selezione di scenari migliorativi. The perceived quality of the local public transit: a multi-criteria model to select improvement scenarios. Ingegneria Ferroviaria 9:717–743
25. Padala P, Zhu X, Wang Z, Singhal S, Shin KG et al (2007) Performance evaluation of irtualization technologies for server consolidation. HP Labs Tec. Report
26. Ahmad RW, Gani A, Hamid SHA, Shiraz M, Xia F, Madani SA (2015) Virtual machine migration in cloud data centers: a review, taxonomy, and open research issues. J Supercomput 71(7):2473–2515
27. Silva Filhoa MC, Monteiroa CC, Inaciob PRM, Freireb MM (2018) Approaches for optimizing virtual machine placement and migration in cloud environments: a survey. J Parallel Distrib Comput 111:222–250
28. Medina V (2014) García JM",A survey of migration mechanisms of virtual machines". ACM Comput Surv (CSUR) 46(3):1–33
29. Osanaiye O, Chen S, Yan Z, Lu R, Choo K-KR, Dlodlo M (2017) From cloud to fog computing: a review and a conceptual live VM migration framework. IEEE Access 5:8284–8300
30. Singh SP, Nayyar A, Kaur H, Singla A (2019) Dynamic task scheduling using balanced VM allocation policy for fog computing platforms. Scalable Comput Pract Exp 20(2):433–456
31. 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
32. 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 Electr Eng 69:334–350
33. Rodrigues TG, Suto K, Nishiyama H, Kato N (2017) Hybrid method for minimizing service delay in edge cloud computing through VM migration and transmission power control. IEEE Trans Comput 66(5):810–819
34. Karthikeyan K, Sunder R, Shankar K, Lakshmanaprabu SK, Vijayakumar V, Elhoseny M, Manogaran G (2018) Energy consumption analysis of Virtual Machine migration in cloud using hybrid swarm optimization (ABC–BA). J Supercomput. https://doi.org/10.1007/s11227-018-2583-3
35. Paulraj GJL, Francis SAJ, Peter JD, Jebadurai IJ (2018) Resource-aware virtual machine migration in IoT cloud. Future Gener Comput Syst 85:173–183
36. He TZ, Toosi AN, Buyya R (2019) Performance evaluation of live virtual machine migration in SDN-enabled cloud data centers. J Parallel Distrib Comput 131:55–68
37. Kansal NJ, Chana I (2016) Energy-aware virtual machine migration for cloud computing—a firefly optimization approach. J Grid Comput 14(2):327–345
38. Narantuya J, Zang H, Lim H (2018) Service-aware cloud-to-cloud migration of multiple virtual machines. IEEE Access 6:76663–76672
39. Patel RP (2019) Energy efficient VM migration in cloud datacenter using Dolphin echolocation optimization with Tchebycheff algorithm. Int J Recent Technol Eng (IJRTE) 8(2):86–94
40. Simarro JLL, Moreno-Vozmediano R, Montero RS, Llorente IM (2011) Dynamic placement of virtual machines for cost optimization in multi-cloud environments. In: 2011 international conference on high performance computing & simulation, Istanbul, Turkey
41. Xu B, Peng Z, Xiao F, Gates AM, Yu J-P (2015) Dynamic deployment of virtual machines in cloud computing using multi-objective optimization. Soft Comput 19(8):2265–2273
42. Yermolovich A, Wimmer C, Franz M (2009) Optimization of dynamic languages using hierarchical layering of virtual machines. In: Proceedings of the 5th symposium on Dynamic languages, pp 79–88
43. 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
44. Singh G, Gupta P (2016) A review on migration techniques and challenges in live virtual machine migration. In: Proceedings of 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)
45. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
46. Meng X, Liu Y, Gaol X, Zhang H (2014) A new bio-inspired algorithm: chicken swarm optimization. In: International conference in swarm intelligence ICSI 2014: advances in swarm intelligence, pp 86–94