Enhancing Harris Hawks Optimization Algorithm for Resource Allocation in Cloud Computing Environments

Ganghua Bai
School of Economics and Management, Hebi Polytechnic, Hebi 458030, China

***Abstract*—Cloud computing is revolutionizing the delivery of on-demand scalable and customizable resources. With its flexible resource access and diverse service models, cloud computing is essential to modern computing infrastructure. In cloud environments, assigning Virtual Machines (VMs) to Physical Machines (PMs) remains a complex and challenging task critical to optimizing resource utilization and minimizing energy consumption. Given the NP-hard nature of VM allocation, solving this optimization problem requires efficient strategies, usually addressed by metaheuristic algorithms. This study introduces a novel method for allocating VMs based on the Harris Hawks Optimization (HHO) algorithm. HHO has exhibited the capacity to provide optimal solutions to specific issues inspired by the hunting behavior of Harris's falcons in the natural world. However, there are often problems with convergence to local optima, which affects the quality of the solution. To mitigate this challenge, this study employs a tent chaotic map during the initialization phase, aiming for enhanced diversity in the initial population. The proposed method, Enhanced HHO (EHHO), has superior performance compared to previous algorithms. The results confirm the effectiveness of the introduced tent chaotic map improvement and suggest that EHHO can improve solution quality, higher convergence speed, and improved robustness in addressing VM allocation challenges in cloud computing deployments.**

Keywords*—*Cloud computing; virtual machine allocation; energy efficiency; resource utilization

I. Introduction

Cloud computing is a revolutionary paradigm that has fundamentally changed how we deal with modern computing [1]. It offers a diverse array of services and resources over the Internet that can be easily customized and accessed as needed [2]. This cutting-edge architecture enables consumers to leverage configurable computing resources, such as applications, storage, servers, and networks [3, 4]. Cloud computing is primarily characterized by its capacity to offer flexibility, agility, and cost-effectiveness by abstracting and virtualizing resources [5]. This enables users to allocate and release resources dynamically according to their specific needs [6]. Efficiently distributing Virtual Machines (VMs) onto Physical Machines (PMs) has become a crucial topic in cloud systems [7]. The allocation procedure substantially influences the usage of resources, consumption of energy, and overall performance of the system in cloud infrastructures [8].

VM allocation entails the optimal assignment of VMs to PMs to achieve optimal resource utilization, minimize energy consumption, and maintain satisfactory performance metrics [9, 10]. Due to the intrinsic complexity and NP-hard nature of this issue, conventional optimization approaches generally fail to deliver efficient solutions within acceptable time limits. Meta-heuristic algorithms have become prominent as effective and adaptable optimization methods to tackle these difficulties [11-13]. These algorithms provide a novel approach to address intricate optimization issues using principles derived from natural occurrences, social behavior, or biological systems [14-16]. Meta-heuristic algorithms are crucial in cloud computing to develop effective techniques to allocate VMs and find feasible solutions to this complex optimization issue [17]. Peer-to-peer (P2P) file sharing plays a crucial role in VM allocation by enabling decentralized distribution of resources, facilitating dynamic resource allocation and load balancing [18]. Furthermore, the integration of machine learning and deep learning techniques in VM allocation enhances decision-making processes by leveraging historical data and patterns to predict resource demands and optimize allocation strategies, ultimately improving overall system efficiency and performance in cloud computing environments [19, 20].

Heidari, et al. [21] introduced the Harris Hawks Optimization (HHO) algorithm, drawing inspiration from the hunting patterns of Harris hawks in the natural world. This algorithm encompasses three distinct stages: exploration, transition to exploitation, and exploitation. This algorithm distinguishes itself with its simplicity in principles, minimal parameterization, and robust local optimization capabilities. Its application has extended across various domains, including image segmentation, neural networks, control of electric machines, and other relevant fields. Despite its merits, the HHO algorithm presents limitations, such as restricted optimization accuracy, sluggish convergence rates, and susceptibility to falling into local optima, aligning with challenges prevalent in several meta-heuristic algorithms. Consequently, numerous researchers have attempted to improve the HHO algorithm.

For instance, Jia, et al. [22] proposed a mutation technique paired with parameter regulation to calculate escape energy during the exploration phase, yielding promising outcomes through parameter regulation. Houssein, et al. [23] suggested the integration of mutation and cross-cooperative gene operators, resulting in the development of an optimization method using oppositional learning. This innovative approach bolstered exploration capabilities and effectively generated the initial population. YiMing, et al. [24] integrated the Chan algorithm to compute a starting point and replaced the original positions to reduce pointless exploration and augment convergence speed. Despite these enhancement strategies, there is still much room for improving the HHO algorithm to address its inherent limitations.

A novel approach to VM allocation utilizing the HHO algorithm is presented in this study. A new variant of the HHO algorithm, Enhanced HHO (EHHO), addresses the limitations of the conventional HHO algorithm. By incorporating a tent chaotic map during the initialization phase, EHHO attempts to increase diversity within the initial population, effectively reducing the tendency of the algorithm to converge prematurely to local optima. This paper comprises five sections. Section II reviews existing research on VM allocation algorithms, highlighting their strengths and limitations. Section III introduces the HHO algorithm for the VM allocation problem. Section IV presents simulation outcomes, validating the efficacy of the proposed algorithm. Finally, Section V summarizes findings and discusses the implications of EHHO for enhancing VM allocation in cloud computing environments.

II. Related Work

This section provides a comprehensive overview of existing research on VM allocation in cloud computing environments. The strengths and weaknesses of various metaheuristic optimization techniques are discussed. Furthermore, the importance of VM allocation for optimizing resource utilization and minimizing energy consumption within cloud infrastructures is highlighted. Table I provides an overview of the methods discussed.

The increasing need for cloud computing services has led to the widespread deployment of worldwide cloud data centers, intensifying the difficulty of effectively controlling the energy usage of these facilities. Despite numerous software and hardware strategies proposed to address this issue, an optimal resolution remains elusive. Tarahomi and Izadi [25] proposed a novel strategy for managing cloud resources online, utilizing the live migration technique of VMs to decrease power usage. Their approach combines a power-aware and prediction-based VM allocation method and creates a three-tier structure to improve the energy efficiency of cloud data centers. Experimental findings underscore the effectiveness of their approach, demonstrating a noteworthy reduction in power consumption while concurrently enhancing service-level agreement violation (SLAV).

The Enhanced-Modified Best Fit Decreasing (E-MBFD) Algorithm was used by Shalu and Singh [26] to introduce a novel VM allocation methodology. This approach utilizes an Artificial Neural Network (ANN) to verify the VMs allocated to PMs. In addition, it provides the benefit of detecting incorrect assignments brought about by inefficient resource use, making the reassignment of these virtual computers easier. Empirical evidence demonstrates that the E-MBFD methodology surpasses traditional methods in terms of reduced SLA violations and decreased power consumption. A VM allocation method using the elephant herd optimization scheme was presented by Madhusudhan, et al. [27]. Upon conducting tests on real-time workloads, the methodology demonstrated substantial energy and resource utilization enhancements versus conventional approaches.

TABLE I. Comparison of VM Allocation Methodologies in Cloud Computing Environments

Method	Methodology	Strengths
[25]	Prediction-based and power-aware VM allocation	Integrates live migration of VMs to reduce power consumption. Three-tier framework for energy efficiency
[26]	Enhanced-modified best fit decreasing	Utilizes artificial neural network. Detects and corrects inefficient resource use.
[27]	Elephant herd optimization for VM allocation	Shows significant improvements in energy consumption and resource utilization.
[28]	Hybrid model with hierarchical task prioritization	Integrates BAT and Bar system model. Minimizes VM overload within the data center.
[29]	Energy-aware flower pollination algorithm	Employs dynamic switching probability. Considers memory, storage, and processor constraints for VM allocation
[30]	Energy-aware VM allocation using a two-step strategy	Uses SAG algorithm for VM power reduction. Addresses energy consumption through multiple VM power-down
[31]	Auction-based setup for online VM allocation	Mathematical model for efficient resource use. Aims to maximize social welfare through resource allocation

Sreenivasulu and Paramasivam [28] suggested an innovative hybrid approach that utilizes a hierarchical method to rank tasks prior to their submission to the scheduler. The Bandwidth-Aware Divisible Task (BAT) scheduling framework was upgraded by incorporating the Bar system approaches, resulting in an advanced hybrid optimization strategy. To mitigate VM overload within the data center, the hybrid model incorporates the Minimum Overload and Minimum Lease policy, facilitating pre-emption. The performance of this hybrid model was assessed through comprehensive evaluation using various parameters. The simulation outcomes convincingly demonstrated the efficacy and efficiency of this novel hybrid model.

Feng, et al. [30] propose an energy-aware VM allocation method. Using a two-step SAG algorithm, multiple VMs in cloud data centers can be powered down to reduce energy consumption. SAG was evaluated through extensive experiments, and its performance was measured and compared with other typical algorithms. In experiments, the global-energy-aware VM allocation method reduced cloud data center energy consumption compared to different algorithms. The problem of online VM allocation with multiple types of resources is addressed by Liu and Liu [31]. To achieve the most efficient overall use of resources, they proposed an accurate mathematical model based on an auction-based setup. Multiple VMs are provided and allocated efficiently to maximize social welfare and encourage users to provide truthful requests.

Usman, et al. [29] suggested the Energy-Aware Flower Pollination Algorithm (E-FPA) to distribute VMs within cloud data centers. Employing an optimization strategy named Dynamic Switching Probability (DSP), the allocation process efficiently discovers near-optimal solutions while exploiting local and global searches. This approach considers the limitations of PMs in terms of memory, storage, and processor while prioritizing energy-conscious allocations. As evidenced by MultiRecCloudSim, using the planet data, E-FPA outperformed First Fit Decreasing (FFD) by 24%, Order of Exchange Migration (OEM) by 21% and genetic algorithm by 22%. Consequently, implementing E-FPA significantly enhanced data center performance, thereby contributing to improved environmental sustainability.

III. PROPOSED METHOD

A. Cloud Model

Cloud computing architecture facilitates the effortless storage, retrieval, and concurrent handling of large volumes of data. Cloud resources, such as PMs and VMs, perform tasks in response to user requests. VM migration is a process specifically designed to address customer requirements promptly and flexibly, thereby ensuring the effective delivery of cloud-based offerings. Cloud computing uses resource allocation methods for efficiently assigning resources to VMs for task execution. Given that the effectiveness of the cloud model can be affected by performance degradation and overall cloud operation, it is crucial to design resource allocation algorithms carefully. Each task in the cloud is assigned a distinct deadline and duration. Following the principle of minimizing costs, the resource allocator assigns tasks to available VMs. The resource allocator consistently changes the state of VMs to guarantee appropriate task allocation and execution. Fig. 1 illustrates the process of allocating resources in the cloud model.

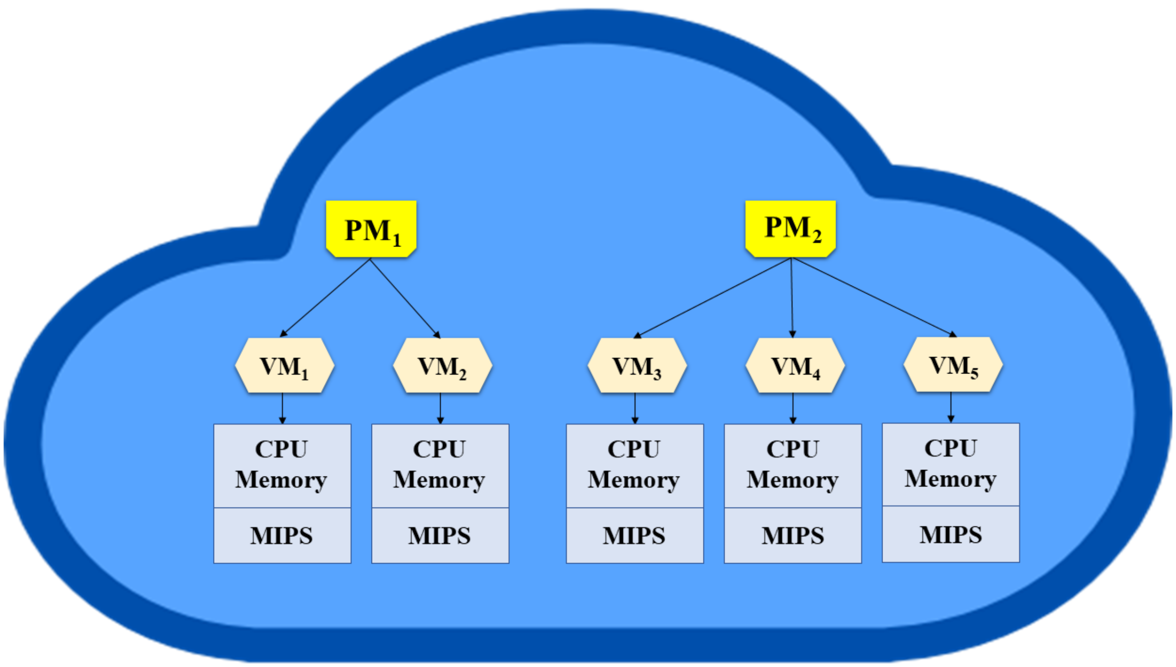

Fig. 1. Resource allocation process.

B. Problem Statement

The need to allocate resources efficiently in cloud service provisioning, taking into account service needs and reconfiguration costs, has resulted in the development of a new computing architecture in the cloud environment. This study presents an efficient strategy for allocating resources in the cloud computing architecture, utilizing the suggested EHHO algorithm. The EHHO algorithm is utilized to achieve optimal resource allocation, hence improving the overall efficiency of the cloud model. Due to cloud resources' extensive and dispersed characteristics, efficient resource allocation is essential for attaining maximum performance. PM oversees and regulates VMs, which differ in terms of MIPS and memory allocated to CPUs. The resource allocation paradigm includes many VM service suppliers associated with VMs, including private and external organizations. Given two PMs, labeled as P_1 and P_2, and five VMs, labeled as V_1, V_2, V_3, V_4, *and* V_5, respectively, user-assigned tasks are performed utilizing these VMs. The collection of VMs is represented as $V = (V_1, V_2, V_3, V_4, V_5)$. Users submit applications labeled as *A*, each consisting of distinct tasks labeled as *s*. Utilizing the EHHO algorithm in the resource allocation strategy greatly improves the efficiency of the cloud model.

C. Task Flow

Consider three tasks, denoted as s_1, s_2, and s_3, each with corresponding deadlines D_1, D_2, and D_3, start times S_1, S_2, and

S_3, and runtimes R_1, R_2, and R_3. These parameters are outlined in Table II, along with the task flow. These tasks are assigned to a VM for processing. The EHHO algorithm assigns tasks to VMs after receiving an application for cloud processing. VM allocation decisions consider variables such as runtime, deadline, and cost. The cloud architecture encompasses both public and private clouds, with a preference for allocating tasks to the private cloud due to its cost-free resources. Assigning the task with the lowest resource cost to the VM for efficient resource allocation is accomplished using the proposed optimization algorithm by evaluating the deadline and runtime of arriving tasks.

TABLE II. TASK FLOW

Tasks	Start time	Runtime	Deadline
s_1	S_1	R_1	D_1
s_2	S_2	R_2	D_2
s_3	S_3	R_3	D_3

Consider three different applications, denoted as T_1, T_2, and T_3, each comprising various tasks. Table III details the tasks' deadlines, runtimes, and start times for each application. The start time for all tasks is indicated as *1*. Tasks s_1 and s_2 belong to application T_1, tasks s_3, s_4, and s_5 belong to T_2, and tasks s_6 and s_7 belong to T_3. The resource allocation decisions are driven by the EHHO algorithm, prioritizing tasks based on their cost-effectiveness, runtime, and deadline considerations.

TABLE III. TASK DEADLINES, RUNTIMES, AND START TIMES FOR DIFFERENT APPLICATIONS

Parameter	Application (T_1)		Application (T_2)			Application (T_3)	
	S_1	S_2	S_3	S_4	S_5	S_6	S_7
Start time	1	1	1	1	1	1	1
Runtime	2	2	3	2	4	3	3
Deadline	4	3	6	5	4	7	8

Table IV illustrates resource allocation to VMs, considering their defined deadline and runtime. Tasks are allocated to VMs based on the lowest cost of each task. Assigning tasks to VMs is done based on the proposed EHHO, which ensures that tasks are completed within the constraints of the runtime and deadline.

TABLE IV. TASKS DETAILS

Time slots	V_1	V_2	V_3
P_1	s_2	s_5	s_4
P_2	s_2	s_5	s_4
P_3	s_2	s_5	s_6
P_4	s_1	s_5	s_6
P_5	s_1	s_5	s_6
P_6		s_3	s_7
P_7		s_3	s_7
P_8		s_3	s_7

Consider eight-time slots and three VMs for task allocation. For tasks s_1 and s_2 under application T_1 with values of 0.4 and 0.2, respectively, s_2 has the minimum value. Consequently, s_2 is allocated to V_1 due to its minimum cost. Given that the runtime of s_2 is 2 and the deadline is 3, it executes in V_1 during time slots P_1 and P_3. With a runtime and deadline of 2 and 4 for s_1, it is allocated to V_1 during time slots P_4 and P_5. For tasks s_3, s_4 and s_5 under application T_2 with values of 0.5, 0.6 and 0.4, respectively, s_5 has the minimum value and is allocated to V_2 during time slots P_1-P_5, given that the runtime of s_5 is 5. Then, s_3 is allocated to V_2 during time slots P_6-P_8. Similarly, the remaining tasks are allocated to VMs based on their minimal values.

D. Fitness Function

The EHHO algorithm, introduced in this study, assigns tasks to VMs at the lowest cost. The EHHO algorithm performs resource allocation by taking into account fitness values linked to characteristics, including skewness, resource utilization, MIPS, RAM, and CPU usage. Fig. 2 illustrates the suggested strategy for allocating cloud resources. Cloud resources are allocated based on the solution vector encoded for best performance. As shown in Fig. 3, each task is associated with a solution vector. The suggested optimization approach distributes tasks to VMs, prioritizing the task with the lowest value. When allocating resources in the cloud, tasks are compared based on their values. The solution vector has the form [1 × 7], representing the allocation of seven tasks.

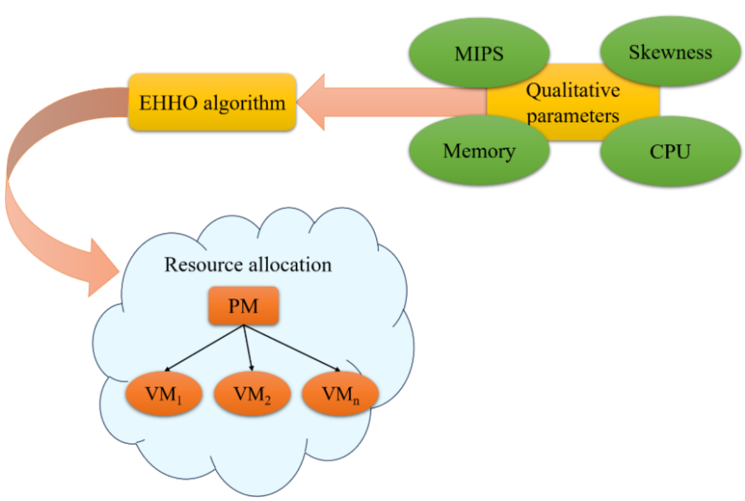

Fig. 2. Resource allocation model based on EHHO algorithm.

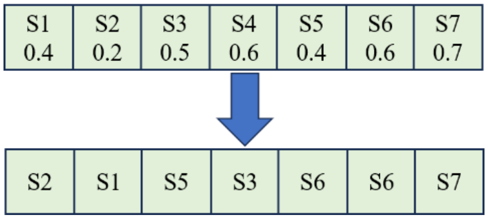

Fig. 3. Solution encoding.

The fitness evaluation aims to compute the fitness function and acquire satisfactory solutions. The fitness function with the lowest score is considered to be the optimal one. The fitness value is computed using Eq. (1).

$$f = \sum_{n=1}^{t} R_n + \sum_{m=1}^{p} (F_m + (1 - B_m) + G_m) \quad (1)$$

where, t refers to task count, G_m represents skewness, B_m reflects the resource utilization of m^{th} VM, F_m indicates nth VM cost, and R_n stands for the runtime of n^{th} task. The terms B_m and G_m are defined by Eq. (2) and Eq. (3), respectively.

$$B_m = \frac{U_m^v \times Q_m^v \times L_m^v}{U_m^t \times Q_m^t \times L_m^t} \times \frac{W_u}{W_x} \quad (2)$$

$$G_m = \left(\frac{B_m}{B} - 1\right)^2 \quad (3)$$

where, U_m^v signifies the MIPS usage of the m^{th} VM, Q_m^v describes the memory usage of the m^{th} VM, and L_m^v indicates the CPU usage of the m^{th} VM. U_m^t stands for the available MIPS, Q_m^t represents the available memory, and L_m^t represents the entire CPU capacity in the m^{th} VM. W_u refers to the use of a time slot, whereas W_x represents the maximum total number of slots.

E. Improved HHO Algorithm for Resource Allocation

The HHO algorithm employs mathematical equations to simulate Harris Hawks' hunting behavior to identify the most optimal solutions for problems. The Harris hawks in this algorithm serve as the candidate solutions, while the prey symbolizes the ideal solution [32]. The HHO algorithm consists of two phases: global exploration and local exploitation. The transition from global exploration to local exploitation is determined by the energy equation of the prey, calculated by Eq. (4) and Eq. (5), where E denotes the escape energy of the prey, E_0 represents the initial energy state of the prey, T is the maximum number of iterations, and *rand* is a random number between 0 and 1. When the absolute value of E is greater than or equal to 1, the HHO algorithm enters the global exploration phase. On the other hand, when the total value of E is less than 1, local exploitation begins. The different phases of the HHO are depicted in Fig. 4, illustrating how hawks trace, encircle, and ultimately attack their prey.

$$E = 2E_0\left(1 - \frac{t}{T}\right) \quad (4)$$

$$E_0 = 2 \times rand - 1 \quad (5)$$

During the period of global exploration, the Harris hawks thoroughly examine and oversee the search space, which is determined by the lower bound (*lb*) and upper bound (*ub*). They employ two distinct tactics to look for prey in a random manner. The Harris hawks' location is updated in each cycle with a certain probability (q) using Eq. (6). The equation relates the locations of the Harris Hawks in the $(t+1)^{th}$ and t^{th} iterations, denoted as X_{t+1} and X_t, respectively. The variable $X_{prey,t}$ represents the prey locations in the t^{th} iteration. r_1, r_2, r_3, r_4, and q are uniformly distributed random variables in the range [0, 1]. *lb* and *ub* represent the bottom and upper limits of the search space, respectively. The variable $X_{rand,t}$ represents the random position of the Harris hawks in the t^{th} iteration. The variable $X_{average,t}$ represents the mean location of the Harris hawks in the t^{th} iteration, given a population size of N.

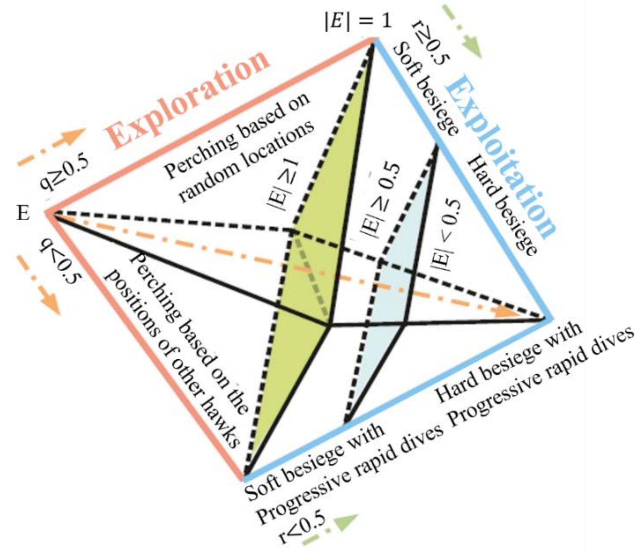

Fig. 4. HHO steps.

$$X_{t+1} = \begin{cases} X_{rand} - r_1|X_{rand} - 2r_2X_t| & q \geq 0.5 \\ (X_{prey,t} - X_{average,t}) - r_3(lb + r_4(ub - lb)) & q < 0.5 \end{cases} \quad (6)$$

These equations and strategies are utilized to guide the search for optimal solutions by the Harris hawks, simulating their hunting behavior to approach the optimal solution iteratively.

$$X_{average,t} = \frac{1}{N}\sum_{i=1}^{N} X_{i,t} \quad (7)$$

During the local exploitation stage, the value of E plays a vital role in determining the besiege technique used by the Harris hawks. A gentle besiege technique is undertaken when the magnitude of E is larger than or equal to 0.5. On the other hand, if the absolute value of E is less than 0.5, a rigorous besiege tactic is executed. The likelihood of the prey's successful escape is governed by the randomly generated variable u, created at initialization. If the value of u is larger than or equal to 0.5, then the prey can escape successfully. HHO employs four tactics to replicate the chase attack behaviors observed in Harris hawks, taking into account both the hawks' pursuit approach and the escape behavior of their prey.

When the escape energy E of the prey is sufficient and u is greater than or equal to 0.5, the Harris hawks gradually consume the prey's energy. They then execute a surprise dive in the best position to capture the prey. The position update strategy is given by Eq. (8)-(10), where ΔX_t represents the difference between the positions of the Harris hawks and the prey during each iteration, J denotes the random jump of the prey when escaping, and r_5 is a random number between 0 and 1.

$$X_{t+1} = \Delta X_t - E\left|JX_{prey,t} - X_t\right| \tag{8}$$

$$\Delta X_t = X_{prey,t} - X_t \tag{9}$$

$$J = 2(1 - r_5) \tag{10}$$

When the prey is exhausted and the escape energy E is very low $(|E| < 0.5)$, the Harris hawks swiftly raid the prey. The position update strategy is expressed by Eq. (11), which determines the rapid movement towards the prey.

$$X_{t+1} = X_{prey,t} - E|\Delta X_t| \tag{11}$$

When the escape energy E of the prey is sufficient $(|E| \geq 0.5)$ but u is less than 0.5, the Harris hawks establish a soft besiege strategy before launching an attack. The Levy function (LF) is integrated into HHO to simulate the prey's jumping action and escape mode. The position update strategy is given by Eq. (12)- (14), where D represents the problem dimension, S is a random vector of size $1 \times D$, u and v are random values between 0 and 1, and β is a constant set to 1.5.

$$X_{t+1} = \begin{cases} Y: X_{prey,t} - E\left|JX_{prey,t} - X_t\right| & F(Y) < F(X_t) \\ Z: Y + S \times LF(D) & F(Z) < F(X_t) \end{cases} \tag{12}$$

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}} \tag{13}$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}}\right)^{\frac{1}{\beta}} \tag{14}$$

When the prey's escape energy E is insufficient $(|E| < 0.5)$, the Harris hawks construct a hard besiege strategy before striking, reducing the average position distance between themselves and the escaping prey. Eq. (15) represents the expression of the position updating approach.

$$X_{t+1} = \begin{cases} Y: X_{prey,t} - E\left|JX_{prey,t} - X_{m,t}\right| & F(Y) < F(X_t) \\ Z: Y + S \times LF(D) & F(Z) < F(X_t) \end{cases} \tag{15}$$

HHO uses the energy parameter and the factor u to control the hunting strategies between the Harris hawks and prey. This allows the algorithm to move towards the best possible solution for the situation at hand. Recent studies have demonstrated that the integration of chaotic maps into population-based metaheuristic algorithms can enhance the efficiency of the search process. Chaotic maps are commonly incorporated at several stages of the algorithm, including the beginning population, exploration, or exploitation phase. The primary goal of this research is to augment the variety of the beginning population.

The initial location of the population provides a notable influence on both the variety of the population and the stability of the algorithm. Although the HHO algorithm gives the random distribution of population positions during initialization, it does not guarantee uniformity. Chaotic sequences have the properties of ergodicity and high unpredictability, which make them very suitable for improving performance. Chaotic mapping produces pseudo-random numbers that follow a uniform distribution from 0 to 1. By utilizing chaotic mapping, the starting placements of the hawks may be altered, thereby enhancing variation.

The mathematical description of the modification to the initial positions is shown in Eq. (16). In this equation, X_{i+1} represents the new position of the hawks after applying chaotic mapping, X_i denotes the current position of the hawks, and the parameter a is set to 0.7. By incorporating chaotic mapping into the initialization process, population diversity in the HHO algorithm is effectively enhanced, leading to potential improvements in performance.

$$X_{t+1} = \begin{cases} \frac{X_i}{a} & X_i < a \\ \frac{1-X_i}{1-a} & X_i \geq a \end{cases} \tag{16}$$

IV. Experimental Results

In this section, we aim to comprehensively assess and compare the performance of our proposed resource allocation algorithm (EHHO) with several existing approaches. To rigorously evaluate the effectiveness of our algorithm, we conducted a series of experiments utilizing the Matlab simulator version 2016b. Specifically, we selected three related algorithms for comparison: Glow Worm Swarm Optimization (GWO) [33], genetic [34], Particle Swarm Optimization (PSO) [35], and original HHO [21]. These algorithms were chosen based on their relevance and established usage in addressing similar optimization problems within cloud computing environments. The experiments were meticulously designed to cover a range of scenarios and configurations outlined in Table V. We employed key performance metrics, including skewness, CPU usage, memory utilization, and resource consumption, to objectively evaluate the effectiveness of our algorithm in comparison to the selected benchmarks. Furthermore, to ensure the robustness and reliability of our findings, each experiment was conducted multiple times, and the results were analyzed using statistical methods to account for variability and ensure consistency. This systematic approach allowed us to draw meaningful comparisons and insights regarding the performance of our proposed algorithm relative to existing state-of-the-art techniques.

TABLE V. SIMULATION PARAMETERS

Entity	Parameter	Value
Datacenter	Count	4
	Number of hosts	1
	Storage	1,000,000*2
	Bandwidth	10,000*2
	RAM	16,384*2
VM	Count	30
	Bandwidth	1000
	RAM	512 MB
	MIPS	1000*2
Cloudlets	Total number of tasks	100
	Task length	200

Resource utilization is a quantitative indicator that calculates the proportion of allocated resources to the overall number of available resources. It evaluates the efficiency of resource utilization.

$$R = \frac{C}{W} \tag{17}$$

Memory usage is the proportion of memory resources that are used over a period of time to process all tasks that have been submitted. The calculation is performed using Eq. (18), where v_i represents the total available memory and u_i represents the memory demanded for task execution.

$$M = \sum_{i=1}^{y} \frac{u_i}{v_i} \tag{18}$$

CPU utilization refers to the mean amount of CPU resources used by all servers when processing user requests. Eq. (19) is employed to determine the value, with H_i representing the total available CPU resources and E_i denoting the CPU resources demanded for task execution.

$$C = \sum_{i=1}^{y} \frac{E_i}{H_i} \tag{19}$$

Skewness quantifies the degree of asymmetry or lack of evenness in a probability distribution. It offers insight into the disparate consumption of various resources on a server. Skewness arises when a performance manager operates many memory-intensive VMs with a low workload, resulting in inadequate memory and a shortage of resources to support an extra VM. Eq. (20) is used to measure the unevenness in resource use throughout a server, which is known as skewness. The equation defines R as the resource consumption of the n^{th} VM, whereas A represents the average resource utilization.

$$W = \left(\frac{R_n}{A} - 1\right)^2 \tag{20}$$

The suggested technique demonstrates higher performance compared to current algorithms while considering 30 VMs. Fig. 5 to Fig. 8 depict the persistent superiority of the EHHO algorithm over the PSO, genetic, and GWO algorithms. The EHHO algorithm performs better in reducing skewness values, achieving faster convergence and maintaining this improvement even with increasing iterations. The excellence of this system is related to its rapid and precise adjustment to different datasets, which is made possible by enhanced learning rates and variable adjustments. Therefore, it allows for higher levels of effective optimization, resulting in improved efficiency.

Fig. 5 demonstrates that the suggested algorithm improves resource utilization in comparison to previous strategies while keeping the number of repetitions constant. This emphasizes its higher effectiveness and capacity to provide better outcomes with fewer repetitions. Moreover, Fig. 6 illustrates that the proposed technique has improved efficiency in memory utilization, requiring significantly less memory than existing methods for the same number of repeats. Fig. 7 demonstrates that the proposed method outperforms existing models in terms of task efficiency. It achieves greater efficiency by reducing the time required to accomplish tasks without altering the number of repeats.

The proposed approach clearly exhibits greater performance in comparison to existing algorithms across a range of measures. Although the statistics mostly show improvements in skewness values, it is crucial to underline that the algorithm's advantages also include enhanced memory utilization. The EHHO algorithm demonstrates improved efficiency in memory use, shown in Fig. 6. This figure clearly depicts a decrease in memory usage compared to other approaches that have the same number of repeats. This increase demonstrates the algorithm's capacity to efficiently allocate resources, leading to better usage of memory resources in the cloud computing environment. The EHHO method enhances resource optimization and operational efficiency in cloud computing systems by minimizing memory utilization.

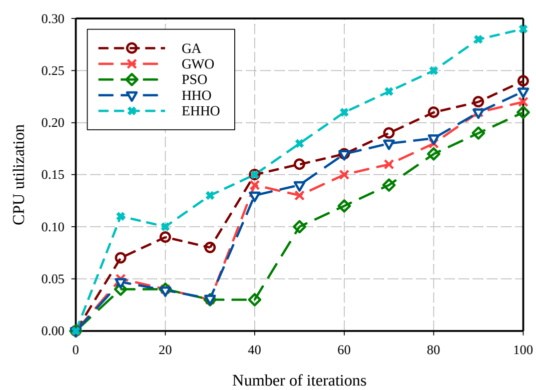

Fig. 5. CPU utilization comparison.

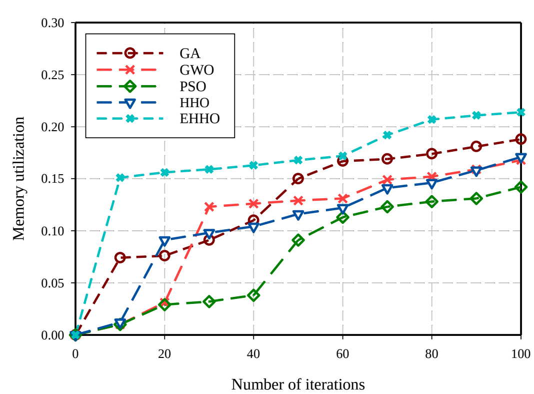

Fig. 6. Memory utilization comparison.

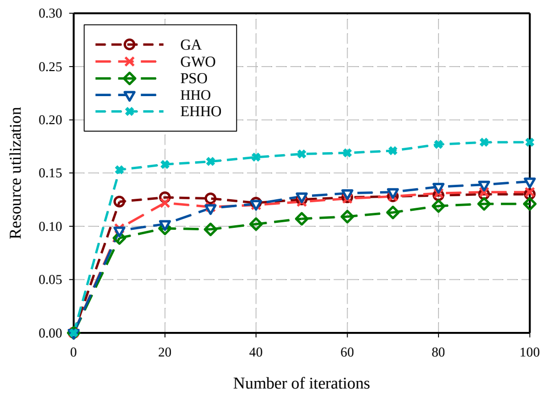

Fig. 7. Resource utilization comparison.

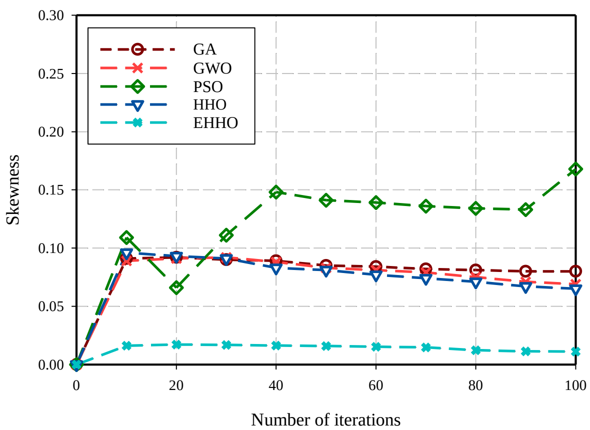

Fig. 8. Skewness comparison.

V. CONCLUSION

This research introduced an energy-efficient optimization method utilizing the HHO algorithm for allocating VMs in cloud computing environments. The suggested approach was evaluated against conventional techniques such as PSO, GWO, and genetic algorithms. Performance testing has shown that EHHO is better in several aspects. First and foremost, EHHO routinely surpasses other algorithms in terms of skewness. It rapidly produces reduced skewness values and maintains them consistently, even with additional repetitions. The improvement may be ascribed to the algorithm's heightened learning rate and parameter adjustment, which allows it to adapt to diverse datasets more efficiently. A decrease in skewness suggests a more equitable and effective allocation of resources among servers. Moreover, the suggested method demonstrates enhanced efficiency in utilizing resources. It efficiently employs a larger quantity of resources compared to current methods during an equivalent number of repetitions. This exemplifies its heightened efficacy and capacity to attain superior outcomes with fewer repetitions. Efficient allocation of resources is essential in cloud computing systems to maximize performance and fulfill user requirements. Furthermore, the method exhibits exceptional efficiency in terms of memory use. It consumes substantially less memory than current methods for the same number of iterations. This capability is especially beneficial in contexts with limited resources when optimizing memory is crucial for efficient task execution and overall system efficiency.

Given the results and consequences of this work, there are various possible areas for future research that may be explored and developed in the field of cloud computing resource allocation. A topic of potential exploration is examining the scalability and suitability of the EHHO method for larger and more intricate cloud computing settings. This may include expanding the algorithm to handle dynamic variations in workload, diverse kinds of resources, and many optimization targets. Incorporating machine learning approaches, such as reinforcement learning or deep learning, might improve the flexibility and intelligence of resource allocation choices in cloud settings. Moreover, examining the influence of several limitations, such as energy consumption, cost minimization, and security concerns, on resource allocation algorithms may result in the creation of more extensive and resilient optimization frameworks. Furthermore, investigating the implementation of EHHO in future concepts like edge computing and fog computing may provide valuable insights into its efficacy in decentralized and distributed computing settings.

REFERENCES

[1] S. Ahmadi, "Security And Privacy Challenges in Cloud-Based Data Warehousing: A Comprehensive Review," International Journal of Computer Science Trends and Technology (IJCST)–Volume, vol. 11, 2023.

[2] R. Nithiavathy, S. Janakiraman, and M. Deva Priya, "Adaptive Guided Differential Evolution - based Slime Mould Algorithm - based efficient Multi - objective Task Scheduling for Cloud Computing Environments," Transactions on Emerging Telecommunications Technologies, vol. 35, no. 1, p. e4902, 2024.

[3] J. Singh and M. S. Goraya, "An Autonomous Multi-Agent Framework using Quality of Service to prevent Service Level Agreement Violations in Cloud Environment," International Journal of Advanced Computer Science and Applications, vol. 14, no. 3, 2023.

[4] Z. Hai-yu, "Virtual Machine Allocation in Cloud Computing Environments using Giant Trevally Optimizer," International Journal of Advanced Computer Science and Applications, vol. 14, no. 9, 2023.

[5] W. Wang and Z. Liu, "Cloud Service Composition using Firefly Optimization Algorithm and Fuzzy Logic," International Journal of Advanced Computer Science and Applications, vol. 14, no. 3, 2023.

[6] S. Durairaj and R. Sridhar, "MOM-VMP: multi-objective mayfly optimization algorithm for VM placement supported by principal component analysis (PCA) in cloud data center," Cluster Computing, pp. 1-19, 2023.

[7] P. Devarasetty and S. Reddy, "Genetic algorithm for quality of service based resource allocation in cloud computing," Evolutionary Intelligence, vol. 14, pp. 381-387, 2021.

[8] M. Hanini, S. E. Kafhali, and K. Salah, "Dynamic VM allocation and traffic control to manage QoS and energy consumption in cloud computing environment," International Journal of Computer Applications in Technology, vol. 60, no. 4, pp. 307-316, 2019.

[9] V. Mongia and A. Sharma, "Energy Efficient and Performance Aware Multi-Objective Allocation Strategy in Cloud Environment," in 2020 International Conference on Advances in Computing, Communication & Materials (ICACCM), 2020: IEEE, pp. 368-373.

[10] K. Saidi and D. Bardou, "Task scheduling and VM placement to resource allocation in Cloud computing: challenges and opportunities," Cluster Computing, vol. 26, no. 5, pp. 3069-3087, 2023.

[11] B. Pourghebleh, A. A. Anvigh, A. R. Ramtin, and B. Mohammadi, "The importance of nature-inspired meta-heuristic algorithms for solving virtual machine consolidation problem in cloud environments," Cluster Computing, pp. 1-24, 2021.

[12] S. Mahmoudinazlou, A. Alizadeh, J. Noble, and S. Eslamdoust, "An improved hybrid ICA-SA metaheuristic for order acceptance and scheduling with time windows and sequence-dependent setup times," Neural Computing and Applications, pp. 1-19, 2023.

[13] A. Larijani and F. Dehghani, "An Efficient Optimization Approach for Designing Machine Models Based on Combined Algorithm," FinTech, vol. 3, no. 1, pp. 40-54, 2023.

[14] L. Jie, P. Sahraeian, K. I. Zykova, M. Mirahmadi, and M. L. Nehdi, "Predicting friction capacity of driven piles using new combinations of neural networks and metaheuristic optimization algorithms," Case Studies in Construction Materials, vol. 19, p. e02464, 2023.

[15] D.-C. Wu, M. Momeni, A. Razban, and J. Chen, "Optimizing demand-controlled ventilation with thermal comfort and CO2 concentrations using long short-term memory and genetic algorithm," Building and Environment, vol. 243, p. 110676, 2023.

[16] A. Larijani and F. Dehghani, "A Computationally Efficient Method for Increasing Confidentiality in Smart Electricity Networks," Electronics, vol. 13, no. 1, p. 170, 2023.

[17] B. G. Sheena and N. Snehalatha, "Multi - objective metaheuristic optimization - based clustering with network slicing technique for Internet of Things - enabled wireless sensor networks in 5G systems," Transactions on Emerging Telecommunications Technologies, vol. 34, no. 8, p. e4626, 2023.

[18] M. Momeni, D.-C. Wu, A. Razban, and J. Chen, "Data-driven Demand Control Ventilation Using Machine Learning CO2 Occupancy Detection Method," 2020.

[19] A. Omidi, A. Heydarian, A. Mohammadshahi, B. A. Beirami, and F. Haddadi, "An embedded deep learning-based package for traffic law enforcement," in Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 262-271.

[20] R. Choupanzadeh and A. Zadehgol, "A Deep Neural Network Modeling Methodology for Efficient EMC Assessment of Shielding Enclosures Using MECA-Generated RCS Training Data," IEEE Transactions on Electromagnetic Compatibility, 2023.

[21] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," Future generation computer systems, vol. 97, pp. 849-872, 2019.

[22] H. Jia, C. Lang, D. Oliva, W. Song, and X. Peng, "Dynamic harris hawks optimization with mutation mechanism for satellite image segmentation," Remote sensing, vol. 11, no. 12, p. 1421, 2019.

[23] E. H. Houssein, N. Neggaz, M. E. Hosney, W. M. Mohamed, and M. Hassaballah, "Enhanced Harris hawks optimization with genetic operators for selection chemical descriptors and compounds activities," Neural Computing and Applications, vol. 33, pp. 13601-13618, 2021.

[24] M. YiMing, S. Zhidong, and Z. Kang, "TDOA Localization Based on Improved Harris Hawk Optimization Algorithm [J]," Computer Engineering, vol. 46, no. 12, pp. 179-184, 2020.

[25] M. Tarahomi and M. Izadi, "A prediction - based and power - aware virtual machine allocation algorithm in three - tier cloud data centers," International Journal of Communication Systems, vol. 32, no. 3, p. e3870, 2019.

[26] Shalu and D. Singh, "Artificial neural network-based virtual machine allocation in cloud computing," Journal of Discrete Mathematical Sciences and Cryptography, vol. 24, no. 6, pp. 1739-1750, 2021.

[27] H. Madhusudhan, P. Gupta, D. K. Saini, and Z. Tan, "Dynamic Virtual Machine Allocation in Cloud Computing Using Elephant Herd Optimization Scheme," Journal of Circuits, Systems and Computers, p. 2350188, 2023.

[28] G. Sreenivasulu and I. Paramasivam, "Hybrid optimization algorithm for task scheduling and virtual machine allocation in cloud computing," Evolutionary Intelligence, vol. 14, no. 2, pp. 1015-1022, 2021.

[29] M. J. Usman et al., "Energy-efficient virtual machine allocation technique using flower pollination algorithm in cloud datacenter: a panacea to green computing," Journal of Bionic Engineering, vol. 16, pp. 354-366, 2019.

[30] H. Feng, Y. Deng, and J. Li, "A global-energy-aware virtual machine placement strategy for cloud data centers," Journal of Systems Architecture, vol. 116, p. 102048, 2021.

[31] X. Liu and J. Liu, "A truthful online mechanism for virtual machine provisioning and allocation in clouds," Cluster Computing, vol. 25, no. 2, pp. 1095-1109, 2022.

[32] R. Zhang, S. Li, Y. Ding, X. Qin, and Q. Xia, "UAV Path Planning Algorithm Based on Improved Harris Hawks Optimization," Sensors, vol. 22, no. 14, p. 5232, 2022.

[33] X. Liu, X. Zhang, W. Li, and X. Zhang, "Swarm optimization algorithms applied to multi-resource fair allocation in heterogeneous cloud computing systems," Computing, vol. 99, pp. 1231-1255, 2017.

[34] F.-H. Tseng, X. Wang, L.-D. Chou, H.-C. Chao, and V. C. Leung, "Dynamic resource prediction and allocation for cloud data center using the multiobjective genetic algorithm," IEEE Systems Journal, vol. 12, no. 2, pp. 1688-1699, 2017.

[35] L.-D. Chou, H.-F. Chen, F.-H. Tseng, H.-C. Chao, and Y.-J. Chang, "DPRA: Dynamic power-saving resource allocation for cloud data center using particle swarm optimization," IEEE Systems Journal, vol. 12, no. 2, pp. 1554-1565, 2016.