A Two-Tier Energy-Aware Resource Allocation Framework in Cloud Computing Using the Spider Wasp Optimizer

Chang Lei, Liu Chenguang*
Department of Information Engineering, Hebei Chemical & Pharmaceutical College, Hebei, Shijiazhuang 050026, China
E-mail: centos168@163.com, liuchenguang0522@163.com
*Corresponding author

Keywords: green cloud computing, resource allocation, quality of service, energy efficiency, spider wasp optimizer, metaheuristic algorithms, service response time, power consumption

Received: March 2, 2025

Cloud infrastructures are increasingly subject to performance and environmental requirements, particularly in large data centers with substantial energy expenditures. This paper addresses the issue of energy-aware quality of service (QoS) guaranteed resource allocation and presents an original two-tiered resource allocation scheme based on a Green Management Module (GMM) and a Spider Wasp Optimizer (SWO)-driven Cloud Management Module (CMM). The CMM pre-filters candidate resources based on the latest measurements beforehand, and the GMM selects the optimal resource, aided by the SWO, to inform the allocation decision. Experimental results show that the scheme reduces service response time and energy consumption by 25% and 26% compared to baseline schemes such as dynamic VM provisioning and VM allocation policies. The scheme presents an energy-aware and scalable design with tremendous prospects for optimizing resource use in existing clouds.

Povzetek: Članek predstavi dvostopenjski, na energijo občutljiv okvir razporejanja virov (CMM+GMM) z metaheuristiko Spider Wasp Optimizer, ki v simulacijah skrajša odzivni čas in porabo energije v oblačnih podatkovnih centrih.

1 Introduction

1.1 Context

Computers and cell phones have proliferated worldwide over the last two decades. Due to this, competition among companies has also increased, and more emphasis has been placed on expanding into new geographical territories to ensure sustainable profitability [1]. To effectively adapt to these changing demands and achieve optimal operational efficiency, improvements in resource utilization are necessary [2]. Cloud computing offers consumers a diverse set of unlimited computing resources tailored to their individual needs. These are accessed instantly at any time and from any location. This is achieved by a pay-per-use system where customers are only charged for the services they actively use [3]. Recent advancements in related domains, such as the use of machine learning for analyzing economic factors [4], optimal allocation strategies for energy and infrastructure in microgrids [5, 6], and intelligent security systems for dynamic network environments [7], highlight the potential of computational intelligence in addressing complex operational challenges. These developments collectively demonstrate how optimization, predictive analytics, and adaptive control can be integrated into cloud-based solutions to improve efficiency, security, and sustainability.

The cloud provides resources with consolidated horizontal and vertical scaling, as well as flexible or elastic functions, which prevent businesses from paying for resources that are not being used. Additionally, the flexibility and scalability of cloud services enable companies to quickly adapt to changes in market demand [8]. Such agility will foster improved innovation and competitive advantages. Central management of cloud infrastructure makes cloud infrastructure provision easier [9]. Both cloud service providers and customers can manage resources more effectively by adopting virtualization technology and implementing dynamic resource scheduling strategies [10].

1.2 Problem statement

Efficient resource allocation would minimize job execution time, maximize resource utilization, and reduce resource consumption. The task scheduling aspect has been considered paramount in cloud data centers to mitigate resource constraints resulting from increased workload [11]. The allocation of tasks to cloud resources to meet scheduling objectives and optimize QoS metrics remains a complex and evolving challenge in cloud computing. Recent research [12, 13] further investigates this aspect, highlighting the challenges of balancing cost, energy efficiency, Service Level Agreement (SLA) satisfaction, and dynamic workload variability. The primary scheduling issue is determining the right resource to accomplish tasks. Scheduling algorithms can enhance most of the QoS factors, namely, execution cost, energy consumption, reliability, task rejection rate, and resource efficiency. At the same time, it guarantees compliance with SLAs while considering constraints such as deadlines, priorities, and the need to resolve load imbalance issues affecting overused and underutilized resources [14].

The growth of cloud computing infrastructure and the expansion of data centers over the past few years underscore the importance of energy conservation in cloud computing environments [15]. As is well known, data centers consume a substantial amount of energy, resulting in increased operational costs and environmental damage [16].

Energy consumption in cloud data centers has become a critical concern due to its economic and environmental impact. As cloud services scale globally, data centers are consuming increasingly large amounts of electricity, estimated to account for nearly 1% of global electricity demand [17]. Not only does it result in exceedingly high cloud operation expenses, but it is also a significant contributor to carbon emissions and climate change. According to [18], unsustainable energy consumption in data centers negates the goals of green IT and necessitates effective allocative efficiency mechanisms. Therefore, it is needed to develop QoS-aware and energy-saving resource allocation systems to achieve scalable, affordable, and environmentally friendly cloud infrastructure.

1.3 Contribution

This paper proposes an innovative resource allocation approach to maximize energy savings in cloud computing settings. It combines two modules, namely green management and cloud management. The first module finds appropriate resources from the existing pool. In contrast, the subsequent module further narrows down this selection to establish the most efficient allocation of resources in terms of power consumption. The proposed method significantly reduces average service response time using this two-tiered approach. This research makes several contributions to the field.

- Enhancing server status analysis efficiency by grouping cloud servers based on their geographical proximity;
- Optimizing service provisioning time by assigning the most suitable servers, taking into account multiple parameters;
- Promoting energy efficiency by allocating the most appropriate cloud server for each task;
- Leveraging metaheuristic algorithms to select appropriate cloud resources effectively.

To guide this study, we formulated the following research questions:

- Can a two-tier resource allocation framework improve both energy efficiency and QoS in cloud computing environments?
- Does the Spider Wasp Optimizer (SWO) outperform traditional metaheuristic algorithms in optimizing energy-aware resource allocation?

Correspondingly, we test the following hypotheses:

- The proposed two-tier framework significantly reduces average service response time compared to baseline methods.
- The SWO algorithm yields superior energy-saving performance by achieving more efficient resource utilization than existing.

While this research focuses on improving energy efficiency and service responsiveness, two of the most pressing challenges in cloud computing, it is essential to acknowledge that resource allocation can also pursue additional objectives. These include cost reduction, fault tolerance, security, scalability, and load adaptability, all of which are critical depending on the application context. Although these goals fall outside the scope of this study, the proposed method could be extended in future work to incorporate multi-objective optimization frameworks that address such concerns in parallel with energy and QoS goals.

2 Related work

As highlighted in Table 1, energy consumption and resource allocation in cloud computing have made significant advancements. Various approaches have been proposed to minimize energy consumption and optimize resource usage.

Nguyen and Cheriet [19] introduced an environmentally conscious virtual slice framework that increases energy efficiency while accounting for the intermittent nature of renewable energy sources. Virtual Machine (VM) deployment and traffic management within a virtual data center are optimized based on traffic volume, VM placement, bandwidth, and available renewable energy resources. A solution to the virtual slice allocation problem was developed by studying various cloud consolidation strategies. Based on simulations of the GSN network, it was found that the model is better at reducing the network footprint than existing methods.

Yang, et al. [20] have developed an efficient and environmentally friendly virtual data center framework and introduced two energy-efficient embedding algorithms. To evaluate the proposed algorithms, extensive simulations were performed across network scales and topologies. Compared with current embedding techniques, the adoption rates of virtual data centers, the long-term revenue of cloud service providers, energy consumption in low-utilization environments, and revenue generation in high-utilization cloud data centers were analyzed.

Khosravi, et al. [21] modeled the total energy cost by combining the server's direct energy consumption with the PUE-based overhead, considering the IT load and outside temperature. The model also finds integrating renewable energy sources into data center locations, reducing dependence on traditional, carbon-intensive grid power. By comparing various VM placement strategies, researchers determined important parameters influencing total energy consumption (renewable and non-renewable), emissions, and costs. The study verifies the supremacy of a method that considers dynamic overall energy consumption, available renewable energy, and PUE while ensuring service level agreements.

Wu, et al. [22] developed a novel two-stage adaptive VM workload prediction model that exploits the inherent regularity of command-line applications. This model enables elastic allocation of CPU and memory resources for VMs. Departing from the ordinary time series forecasting, natural language processing extracts and handles feature attributes from command-line applications, aided by feature reduction through gray relational analysis. The prediction model uses a Bayesian classifier to identify applications that incur significant CPU increases and an adaptive neuro-fuzzy inference system to predict CPU and memory intensity. The elastic resource provisioning strategy was proved and tested through detailed experiments, which confirmed the successful completion of VM performance and resource efficiency with the scheme.

Al-Wesabi, et al. [23] presented a hybrid meta-heuristic approach for optimizing resource allocation and energy efficiency in the cloud computing context. Their methodology begins with extracting relevant features from the customer's task requirements, followed by dimensionality reduction using principal component analysis. The range of function outputs shows the optimal resource allocation. The core of their solution is the hybrid algorithm, developed based on Group Teaching Optimization (GTO) and Rat Swarm Optimization (RSO) algorithms. The hybrid scheme optimizes the resource allocation between the VMs in the cloud data center. The improved performance of the proposed model was demonstrated through rigorous simulation, conducted with the aid of the CloudSim simulator, across various performance indicators.

Tai, et al. [24] developed an original approach to increase resource allocation and energy efficiency in heterogeneous data centers within the vision of Green IT. Their procedure involves modeling energy consumption over time, inter-task dependency, and data backup necessity. Framing the issue as an energy conservation problem that is nonlinear and complex, the researchers employed mathematical programming and Lagrangian relaxation to formulate an energy conservation solution. The procedure optimally allocates computational resources, schedules tasks, and reduces energy consumption, with the advantage of achieving highly achievable performance levels in cloud data centers.

Sharma and Rawat [25] developed a novel hybrid approach by coupling the Spotted Hyena Optimizer (SHO) and an Artificial Neural Network (ANN) to address the problem of VM allocation. The new method was validated with the most significant performance indicators, which included migration number, SLA violation, execution times, resource consumption, and energy consumption. The SHO algorithm was employed to generate an overall dataset, thereby enhancing the accuracy of the ANN model. The hybrid scheme demonstrated improved convergence and global optimality compared to IqMc, SHO, and Genetic algorithms under real and simulated workload behavior.

Gurusamy and Selvaraj [26] introduced a novel framework: resource allocation in cloud computing with a multi-level auto-associative polynomial convolutional neural network. The approach combines Starling Murmuration Optimizer (SMO)-inspired short-term scheduling with the objective of reduced makespan and optimized throughput. Resource management is carried out using a HAPCNN (Hierarchical Auto-Associative Polynomial Convolutional Neural Network) module, taking into account bandwidth and resource loading limitations. Data privacy is ensured with Fractional Discrete Meixner Moments Encryption (FDMME).

Table 1: Previous research on energy-efficient resource allocation in cloud computing

Study	Algorithm type	Optimization goals	Computational overhead	Key outcomes	Comparison with the proposed method
[19]	Heuristic	Energy efficiency (using renewable energy)	Medium	Reduces network footprint	No hierarchical optimization; less responsive to real-time load
[20]	Heuristic	Energy usage, revenue, and adoption rate	High	Better embedding in VDCs	Not metaheuristic; lacks fine-grained optimization
[21]	Analytical model	Energy + carbon cost	Medium	Accounts for PUE and renewables	Requires real-time environmental data integration
[22]	Hybrid (NLP + ANFIS)	VM performance and resource use	High	Elastic VM allocation	High complexity, no focus on metaheuristics
[23]	Metaheuristic (GTO + RSO)	Resource allocation and energy efficiency	Medium	Validated via CloudSim	Does not integrate real-time queue and location metrics
[24]	Mathematical programming	Scheduling + energy use	High	Green-aware task allocation	More static and centralized vs. dynamic SWO
[25]	Metaheuristic + ANN	VM allocation, SLA, and energy usage	Medium	Improved convergence	Relies on extensive ANN training data
[26]	CNN + SMO	Resource utilization and task scheduling	Very High	Strong security and good throughput	High computational cost and complexity

Proposed	Metaheuristic	Response time, energy usage, and resource utilization	Low	25% ↓ in response time and 26% ↓ in energy use	Two-tiered, scalable, low-overhead solution with real-time adaptation

While the available work has made immense contributions to energy-efficient VM placement and green cloud policies, there are still some gaps. Most approaches employ non-scalable hierarchical decision hierarchies or static/algebraically tuned algorithms that may converge too abruptly or exhibit inefficient load balancing. Additionally, very few integrate geo-awareness, online queue behavior, and adaptive metaheuristics under the same umbrella. Through the implementation of the two-tier hierarchical structure (CMM-GMM) along with the introduction of the SWO, the proposed approach eliminates the loopholes in achieving optimal QoS and optimal energy consumption.

3 Proposed resource allocation strategy

This section describes a novel resource provisioning method designed to achieve maximum energy efficiency and QoS levels in cloud environments. The intensified environmental crisis, primarily caused by human-made processes and technological advancements, demands sustainable and feasible solutions. Even as cloud computing provides unique efficiency and scalability, it has inadvertently contributed to the carbon footprint by increasing the number of data centers. Given this reality, it is imperative to devise carbon-cutting methods to offset environmental destruction caused by cloud computing.

Energy efficiency in cloud computing requires the optimal allocation of resources. When resources are allocated effectively, service requests can be fulfilled promptly while minimizing energy consumption. Load balancing between cloud servers reduces underutilization and overload, thereby reducing energy consumption. Our research utilizes SWO to determine the optimal resource configurations based on the utility level.

Here, the utility level is an application-independent indicator of the volume of resource utilization, calculated from the readings of the CPU, memory, and the number of active tasks for each interval. Utility level is an abstract representation of the amount of resource utilization, with no dependency on the type of application or the application's semantics being executed on the VM. This enables the SWO algorithm of the GMM to optimize resource provisioning based on the observable level of the system's performance and non-application-centric information, while maintaining a general and flexible architecture design.

SWO was selected due to its hunting- and escape-inspiration adaptive mechanism, offering the best compromise between exploration and exploitation in complex search spaces. Relative to Particle Swarm Optimization (PSO), which is subject to the chance of premature convergence in multimodal areas, and Ant Colony Optimization (ACO), which is underpinned by stochastic path dependency and pheromone updating with the potential to become prohibitively costly, the biologically rooted nest and mating phases of SWO enable more stable diversification of solution and management of convergence. Both global and fine-grained resource configuration refinement is offered, which is necessary in dynamic clouds.

As shown in Figure 1, to facilitate effective resource allocation and sustainable environments, a two-level architecture is proposed that comprises a Green Management Module (GMM) and a Cloud Management

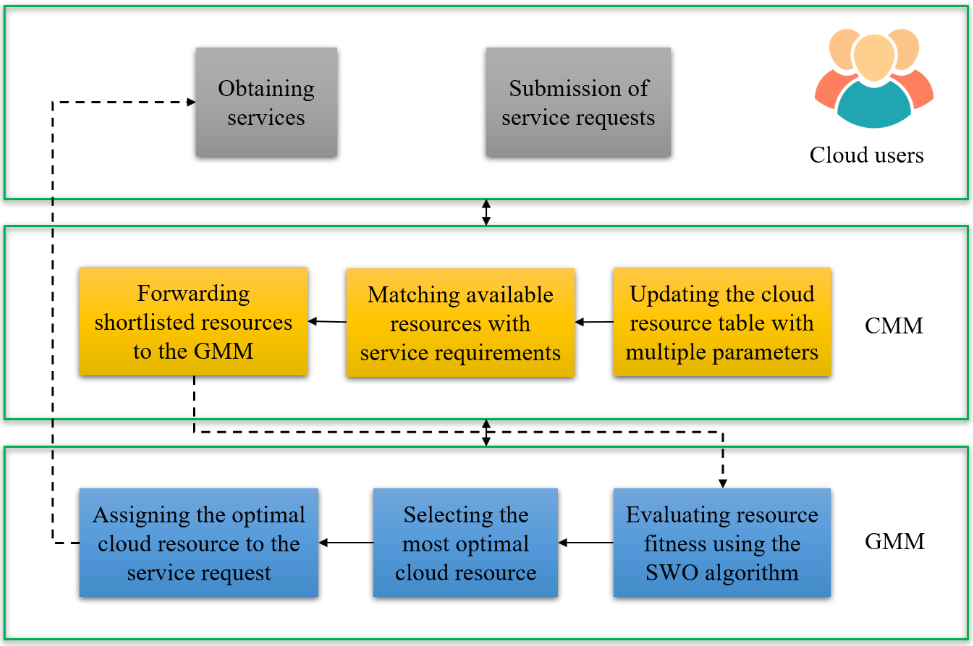
Obtaining services
Submission of service requests
Cloud users
Forwarding shortlisted resources to the GMM
Matching available resources with service requirements
Updating the cloud resource table with multiple parameters
CMM
Assigning the optimal cloud resource to the service request
Selecting the most optimal cloud resource
Evaluating resource fitness using the SWO algorithm
GMM

Figure 1: Two-tier architecture for energy-efficient cloud resource allocation

Module (CMM). The GMM maintains a global snapshot of locations and execution states for cloud data centers, while the CMM facilitates the balanced distribution of service requests across available resources.

The CMM serves as a management layer responsible for monitoring available system resources. The CMM optimizes VM placement and resource selection based on measurable system indicators (e.g., queue length, service response time, distance), without requiring visibility into the internal behavior of VM workloads. A cloud data center, a fundamental part of the infrastructure, consists of multiple cloud servers, each of which can host multiple VMs. Equations 1–3 define distinct identifiers for VMs, servers, and data centers.

$$VM = \{vm_1, vm_2, \dots, vm_n\} \tag{1}$$

$$CS = \{cs_1, cs_2, \dots, cs_n\} \tag{2}$$

$$CDC = \{cdc_1, cdc_2, \dots, cdc_n\} \tag{3}$$

A VM is explicitly named $vm_i^{cs_i}$, indicating its association with the cloud server instance cs_i. This nomenclature enables efficient VM identification and management. The CMM maintains a comprehensive database containing details of all cloud data centers, servers, and VMs. This repository includes attributes such as geographical coordinates, queue length, average service response time, service type, and capacities. Dynamic metrics, such as queue length and response time, are updated in real-time within the CMM to reflect system changes.

The system's workload mainly consists of VM requests initiated by cloud users. The CMM optimizes VM placement and resource allocation without requiring knowledge of the specific applications running in the VMs. Nevertheless, each VM is assumed to operate at maximum capacity and fully utilize the allocated computing resources.

To enhance the QoS of the system, the queue length and service response time are integrated. The overall response time is calculated according to the estimated average duration between consecutive service completions, as defined in Eq. 4, where SR_{time} represents the service response time, *n* refers to the overall number of tasks processed by a particular VM, and SQ_{time} denotes the time taken to complete the service request. Ideally, response time should be minimized to ensure optimal performance. Eq. 5 shows the queue length dynamically adjusted by increasing upon new requests and decreasing upon service completion.

$$ASR_{time} = \frac{\sum_{i=1}^{n}(SR_{time} - SQ_{time})}{Total\ handled\ services} \tag{4}$$

$$QL = \begin{cases} New\ service\ request: QL + 1 \\ Responded\ service: QL - 1 \end{cases} \tag{5}$$

The CMM prioritizes the selection of machines for incoming service requests based on queue length, response time, and geographical distance. The module selects the potential hosts and sends the most appropriate choices to the GMM for further processing.

The GMM's task is to identify the optimum VM to handle service requests. The CMM submits a list of probable candidates to the GMM, and the SWO algorithm determines the best machine from the list. SWO is a metaheuristic algorithm inspired by the hunting, nesting, and parasitic behavior of particular wasp species. The algorithm's basic principles are based on observing female spider wasps that search for, paralyze, and transport spiders to their nests, ultimately laying an egg in the paralyzed prey.

The SWO algorithm initiates the simulation by placing a population of randomly distributed virtual wasps in the search region, thereby modeling the foraging behavior of wasps in their surroundings. The virtual wasps search in a manner analogous to wasp foraging behavior for optimal solutions to problems. When an effective solution is encountered, the algorithm replicates the wasp's capturing and carrying away of their catches through adjusting the solution according to predetermined rules. The essential behavioral components for the SWO algorithm are as follows:

- Searching: Simulates the wasp's initial exploration of the environment to locate potential solutions.
- Following and escaping: Models the wasp's pursuit of prey, incorporating evasion tactics.
- Nesting: Represents the process of constructing a solution by combining and refining existing components to create a new, more comprehensive one.
- Mating: Incorporates a mating mechanism inspired by the wasp's reproductive strategy, utilizing a uniform crossover operator to generate new solutions.

In the SWO framework, each female spider wasp is mathematically represented as a D-dimensional solution vector (Eq. 6). An initial population of *N* such vectors is randomly generated within predefined upper (*H*) and lower (*L*) parameter limits, as shown in Eq. 7. The whole initial population is called SW_{Pop}, with individual solutions stochastically generated according to Eq. 8.

$$\overrightarrow{SW} = [x_1, x_2, x_3, \dots, x_D] \tag{6}$$

$$SW_{Pop} = \begin{bmatrix} sw_{1,1} & sw_{1,2} & \dots & sw_{1,D} \\ sw_{2,1} & sw_{2,2} & \dots & sw_{2,D} \\ \vdots & \vdots & \vdots & \vdots \\ sw_{N,1} & sw_{N,2} & \dots & sw_{N,D} \end{bmatrix} \tag{7}$$

$$\overrightarrow{SW_i^t} = \vec{L} + \vec{r} \times (\vec{H} - \vec{L}) \tag{8}$$

Where *t* represents the generation index and *i* represents the population index (in the range from 1 to *N*), *r* is a D-dimensional random vector with elements evenly distributed between 0 and 1. The algorithm then simulates the hunting, nesting, and mating behavior of spider wasps to identify the optimal solution from a set of candidates determined by CMM.

The optimization process begins with a search phase, analogous to a wasp looking for prey. This exploration occurs randomly within the solution space. Once a possible solution is identified, the algorithm moves into an encirclement and pursuit phase, mimicking the wasp's predatory behavior. Finally, the wasp metaphorically drags the captured prey, representing the identified solution, to a nest for further processing.

The exploration phase of the SWO algorithm mimics the foraging strategies of female spider wasps. In this phase, individuals traverse the search space with a constant step size, as defined in Eq. 9, in search of suitable prey. This equation updates the position of each wasp at each generation (*t*) based on the positions of two randomly selected population members (indexed by a and b) and a random motion component (μ_1) calculated using Eq. 10. Here, r_1 is a uniformly distributed random number, and *rn* follows a normal distribution.

$$\overrightarrow{SW}_i^{t+1} = \overrightarrow{SW}_i^{t} + \mu_1 \times \left(\overrightarrow{SW}_a^{t} - \overrightarrow{SW}_b^{t}\right) \quad (9)$$

$$\mu_1 = |rn| \times r_1 \quad (10)$$

μ_1 represents the displacement of a virtual wasp's position in the solution space during the global exploration phase. It is a D-dimensional vector that influences how far the algorithm explores new candidate solutions at each iteration. Larger values encourage exploration, while smaller values promote exploitation of local optima.

In Eq. 9, two indices a and b are randomly selected from the current population of wasp agents. These correspond to different solution vectors in the search space. The movement of a given wasp is guided by the difference between these two agents' positions, allowing the algorithm to explore based on population diversity. The update rule thus simulates a directional movement influenced by randomly selected peers and a stochastic component, which together ensure broad and adaptive coverage of the solution space.

An additional exploration strategy was incorporated because wasps can lose sight of their prey. Equations 11-13 introduce a second exploration method with a smaller step size, focusing on a randomly selected population member (c). This approach allows for more targeted searches around potential prey locations. B is the binary vector indicating the direction of exploration for each dimension. It determines whether to apply a random motion or move toward another wasp's position. Parameter l (uniformly distributed between -2 and 1) and random numbers r_3 and r_4 (uniformly distributed between 0 and 1) determine the choice between the two exploration methods (Equation 14).

$$\overrightarrow{SW}_i^{t+1} = \overrightarrow{SW}_c^{t} + \mu_2 \times \left(\vec{L} + \vec{r_2} \times \left(\vec{H} - \vec{L}\right)\right) \quad (11)$$

$$\mu_2 = B \times \cos(2\pi l) \quad (12)$$

$$B = \frac{1}{1 + e^{l}} \quad (13)$$

$$\overrightarrow{SW}_i^{t+1} = \begin{cases} Eq.\ 9 & r_3 < r_4 \\ Eq.11 & otherwise \end{cases} \quad (14)$$

As soon as the spider wasp discovers prey caught in its web, it launches an attack. However, the prey often escapes capture by falling to the ground. The wasp pursues the fallen prey, paralyzes it, and transports it to a prepared nest. Alternatively, the wasp may lose track of the prey while descending, requiring a simultaneous pursuit and evasion strategy.

This complex behavior can be modeled as two main trends. During the first chase, the wasp pursues the prey and captures it. This phase is represented mathematically by Eq. 15, which updates the wasp's position based on the location of the prey. Conversely, the second trend simulates escape from prey, characterized by an increasing distance between predator and prey with each iteration. This behavior can cause the prey to seek refuge in areas away from the wasp.

$$\overrightarrow{SW}_i^{t+1} = \overrightarrow{SW}_i^{t} + C \times \left|2 \times \vec{r_5} \times \overrightarrow{SW}_a^{t} - \overrightarrow{SW}_i^{t}\right| \quad (15)$$

Eq. 16 mathematically models predatory pursuit, where the initial proximity between wasps and prey can vary depending on their speeds. Two scenarios are considered: (1) a wasp exceeding the speed of the prey (*C > 0.5*) and (2) a prey exceeding the speed of the wasp (*C < 0.5*). The parameter *C*, a distance control factor initialized at two and decreasing linearly to zero, influences the wasp's speed. If *C* is greater than 0.5, the wasp is faster than its prey, representing escape behavior. On the other hand, if it is less than 0.5, the wasp's movement is significantly restricted, making prey capture more difficult.

$$C = \left(2 - 2 \times \left(\frac{t}{t_{max}}\right)\right) \times r_6 \quad (16)$$

Eq. 16 models the dynamic interaction between the predator (wasp) and its prey (candidate solution) by controlling their relative movement based on a distance control coefficient. The coefficient starts at a high value (typically 2) and linearly decreases toward zero over time (t), thereby gradually shifting the algorithm from exploration to exploitation.

The variables a, t, t_{max}, r_5, and r_6 represent a randomly selected population index, a current and maximum score, a randomly generated vector within the interval [0, 1], and a random number within the same interval, respectively.

The prey's escape from the wasp leads to a gradual increase in their distance. Initially characterized by exploitation, this phase transitions into exploration as the distance increases. Eq. 17 simulates this behavior change. The vector $\overrightarrow{vc}$, normally distributed between k and $-k$, and calculated from Eq. 18, the distance increases. Eq. 19 randomly determines the balance between these two behavioral trends.

$$\overrightarrow{SW}_i^{t+1} = \overrightarrow{SW}_i^{t} \times \overrightarrow{vc} \quad (17)$$

$$k = 1 - \left(\frac{t}{t_{max}}\right) \quad (18)$$

$$\overrightarrow{SW}_i^{t+1} = \begin{cases} Eq.15 & r_3 < r_4 \\ Eq.17 & otherwise \end{cases} \quad (19)$$

At the beginning of the optimization process, all wasps engage in global exploration of the solution space to identify promising regions with near-optimal solutions. Subsequently, an escape mechanism enables the wasp to explore and exploit its local environment, thereby avoiding local optima. Eq. 20 governs the dynamic interplay between exploration and exploitation, with the random variable p ($0 \le p \le 1$) influencing this balance. Figure 2 visually represents this compromise.

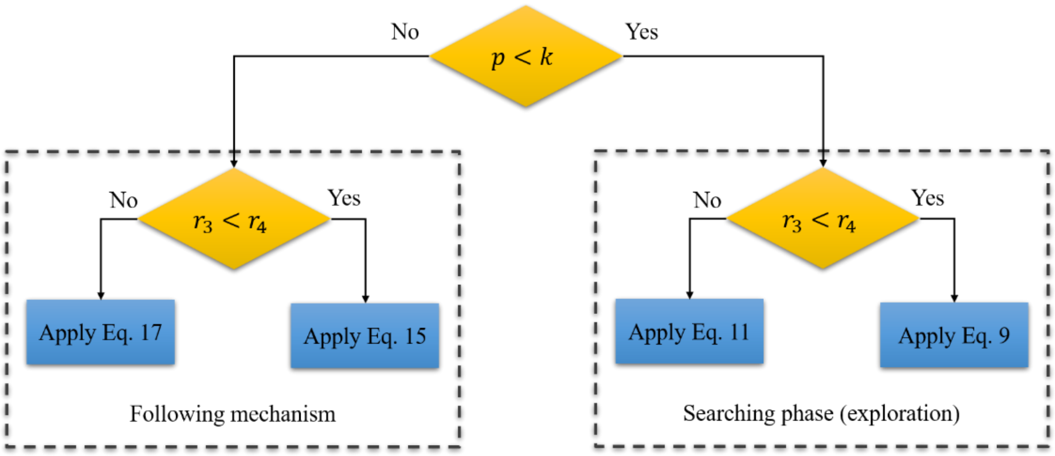

Figure 2: Dynamic balance between global exploration and local exploitation in the SWO algorithm

$$\overrightarrow{SW}_i^{t+1} = \begin{cases} Eq.14 & p < k \\ Eq.19 & otherwise \end{cases} \quad (20)$$

After immobilizing the prey, the female wasp transports it to a pre-made nest. Nest-building strategies vary and include underground burrows, aerial mud structures, or the appropriation of existing cavities. The proposed algorithm consists of two equations to simulate this diversity.

The first equation (Eq. 21) models nest site selection based on proximity to a suitable spider, referred to as the optimal egg-laying site. The term *SW** represents the best-identified solution to date. Conversely, the second equation (Eq. 22) randomizes nest placement around a selected female wasp, considering a step size to prevent nest overlap. In this equation, r_3 is a random value within the interval [0, 1], y is a value generated by Levy flight, *a*, *b*, and *c* are indices of randomly selected solutions, and *U* is a binary vector that controls the step size application. The binary vector *U* is determined by Eq. 23, where r_4 and r_5 represent random values in the interval [0, 1]. These two nest-building strategies are selected randomly according to Eq. 24.

$$\overrightarrow{SW}_i^{t+1} = \overrightarrow{SW}^* + \cos(2\pi l) \times \left(\overrightarrow{SW}^* - \overrightarrow{SW}_i^t\right) \quad (21)$$

$$\overrightarrow{SW}_i^{t+1} = \overrightarrow{SW}_a^t + r_3 \times |\gamma| \times \left(\overrightarrow{SW}_a^t - \overrightarrow{SW}_i^t\right) + (1 - r_3) \times \vec{U} \times \left(\overrightarrow{SW}_b^t - \overrightarrow{SW}_c^t\right) \quad (22)$$

$$\vec{U} = \begin{cases} 1 & \vec{r_4} > \vec{r_5} \\ 0 & otherwise \end{cases} \quad (23)$$

$$\overrightarrow{SW}_i^{t+1} = \begin{cases} Eq.21 & r_3 < r_4 \\ Eq.22 & otherwise \end{cases} \quad (24)$$

Eq. 25 regulates the interaction between hunting and nesting behavior, as shown in Figure 3. All wasps initially look for prey, then build nests and provide food.

$$\overrightarrow{SW}_i^{t+1} = \begin{cases} Eq.20 & i < N \times k \\ Eq.24 & otherwise \end{cases} \quad (25)$$

The SWO algorithm considers spider wasp mating behavior. A key feature of these insects is their sex determination based on host size: smaller hosts produce male offspring, while larger hosts produce female offspring. In this model, each spider wasp represents a possible solution within a generation, with the spider wasp's eggs symbolizing newly generated solutions. Eq. 26 controls the production of these offspring.

$$\overrightarrow{SW}_i^{t+1} = Crossover(SW_i^t, SW_m^t, CR) \quad (26)$$

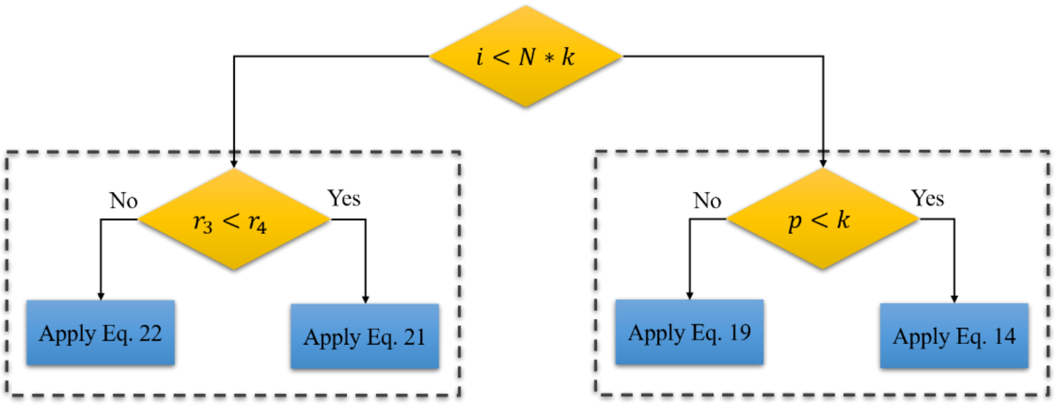

Figure 3: Interaction between hunting and nesting behaviors in the SWO algorithm

In the biological behavior of spider wasps, the size of the host determines the sex of the offspring. Larger hosts usually yield female wasps, which are most likely to survive and reproduce. The same principle is adopted in SWO to skew offspring generations in favor of high-fitness solutions. In the optimization context, a “host” represents a candidate cloud resource configuration, and its size (fitness) determines whether it should propagate high-quality traits.

During the mating phase, new candidate solutions (offspring) are created by combining “female” and “male” wasps using a uniform crossover operator. The crossover rate specifies the amount of genetic material (i.e., configuration parameters for resources) transferred from each parent. The operation injects diversity while retaining good features from high-performance solutions. Therefore, the nest-building and mate selection step in SWO approximates fine local search and solution diversification, which is pivotal for fine-tuning VM allocation strategies that conserve energy while maintaining QoS integrity.

A uniform crossover operator is applied to a parent wasp pair SW_i^t (female) and SW_m^t (male), with the probability defined by the crossover rate (*CR*). To distinguish male wasps from females, Eq. 27 is used, producing male wasps distinct from the female population. This equation uses normally distributed random numbers (β and β_1), the exponential constant (e), and values derived from equations 28 and 29. Crossover combines the genetic material of the parents and produces offspring that inherit their parents' traits. A predefined trade-off rate (*TR*) balances hunting and mating behaviors, as detailed in the experimental section.

$$\overrightarrow{SW}_m^{t+1} = \overrightarrow{SW}_i^t + e^l \times |\beta| \times \vec{v}_1 + (1 - e^l) \times \vec{v}_2 \tag{27}$$

$$\vec{v}_1 = \begin{cases} \vec{x}_a - \vec{x}_i & f(\vec{x}_a) < f(\vec{x}_i) \\ \vec{x}_i - \vec{x}_a & otherwise \end{cases} \tag{28}$$

$$\vec{v}_2 = \begin{cases} \vec{x}_b - \vec{x}_c & f(\vec{x}_b) < f(\vec{x}_c) \\ \vec{x}_c - \vec{x}_b & otherwise \end{cases} \tag{29}$$

The SWO algorithm assesses the suitability of available resources and determines the most suitable option to fulfill the service request. An initial resource selection list is created based on factors such as distance, queue length, average response time, and capacity. The GMM then examines this list to identify the optimal resource using a fitness function, as formalized in Eq. 30.

$$F = \frac{Total\ service\ requests\ \times Total\ energy\ used}{Utility\ degree} \tag{30}$$

4 Performance evaluation

The proposed approach considers three key metrics: average service response time, throughput, and energy consumption. These critical metrics have been chosen for their paramount importance in computing efficiency, overall scalability, and power consumption in computational cloud solutions, using their vital analysis to prove the given problem. The proposed method is compared with existing methods, namely dynamic VM allocation, VM provisioning, and data center provisioning, to validate the model. These approaches represent conventional techniques used in real-world systems. The results showed that the latter proved the model's superiority, particularly in optimizing QoS while significantly reducing energy consumption.

All experiments were conducted using CloudSim 3.0.3, a simulation toolkit that supports modeling and simulation of cloud environments and virtualized resource provisioning. The infrastructure consisted of simulated data centers with 5 to 25 cloud nodes, each node equipped with 16 GB of RAM, 8-core processors, and 5 TB of storage. Workloads ranging from 100 to 1000 service requests were generated synthetically to simulate real-world user traffic.

The average service response time is the time elapsed between the request and the completion of the service. It

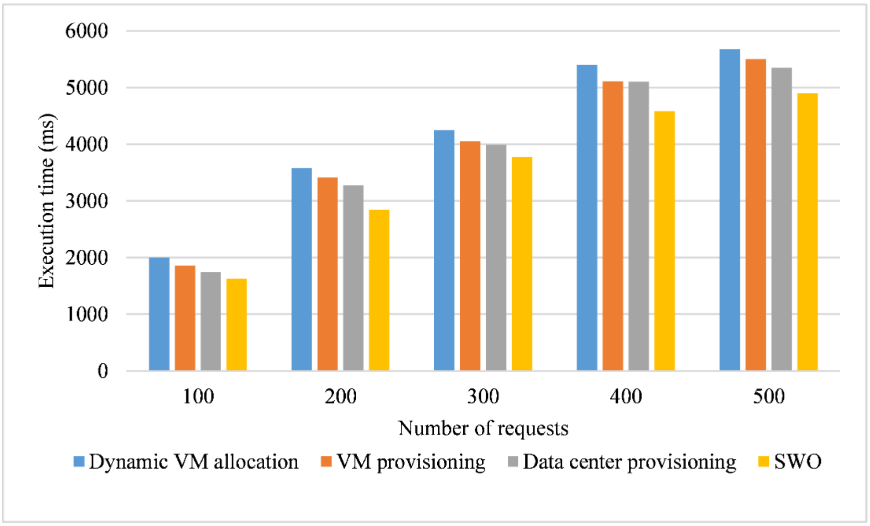

Figure 4: Service execution time comparison

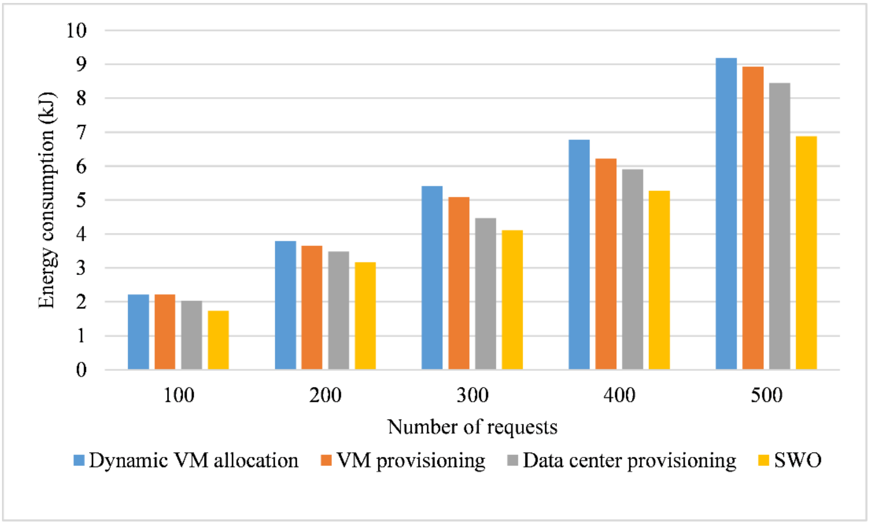

Figure 5: Energy consumption comparison

is evident from Figure 4 that the proposed method outperformed the baseline techniques at all levels of service requests. At the highest workload of 500 requests, the proposed method achieved a 24.7% ± 1.8% reduction in mean execution time, which was statistically significant ($p < 0.05$) compared to the baseline methods. It argues that this performance increase was achieved with the aid of the hierarchical resource selection strategy: the pre-selection of resources was conducted at the CMM level and refined by the GMM to select the most suitable resource based on priority levels.

Figure 4 illustrates the linear relationship between execution time and the number of service requests, demonstrating that the proposed approach is scalable. The proposed approach offers an efficient allocation process, avoiding substantial delays compared to baseline techniques, particularly for high workloads. The improved response behavior and faster execution time are likely the result of the metaheuristics of SWO, which are theoretically designed to explore the solution space systematically and prefer resource configurations with higher efficiency. Although the performance is improved, the specific influence of minimizing the search space and avoiding contention is assumed based on the algorithm design and not directly observed. Future work will involve developing special-purpose metrics to explicitly verify these factors.

Energy efficiency is the key to sustainable cloud data centers. Figure 5 illustrates the energy consumption of the various methods for variable workloads. The maximum energy consumption of the proposed approach is 6.8 kJ at 500 service requests, which is significantly lower compared to Dynamic VM Allocation (9.2 kJ), VM Provisioning (8.9 kJ), and Data Center Provisioning (8.4 kJ). The energy-saving capability of the proposed method comes from emphasizing resource allocation efficiency, thereby reducing redundant activation of underutilized servers.

The most significant energy savings achieved by the proposed approach are typically derived from the hierarchical allocation framework. For each task, the method selects the most suitable server based on

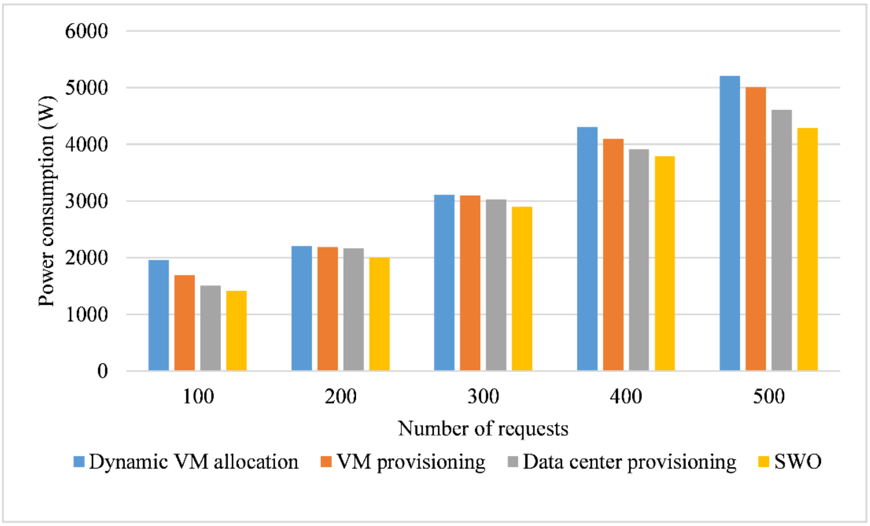

Figure 6: Execution power consumption comparison

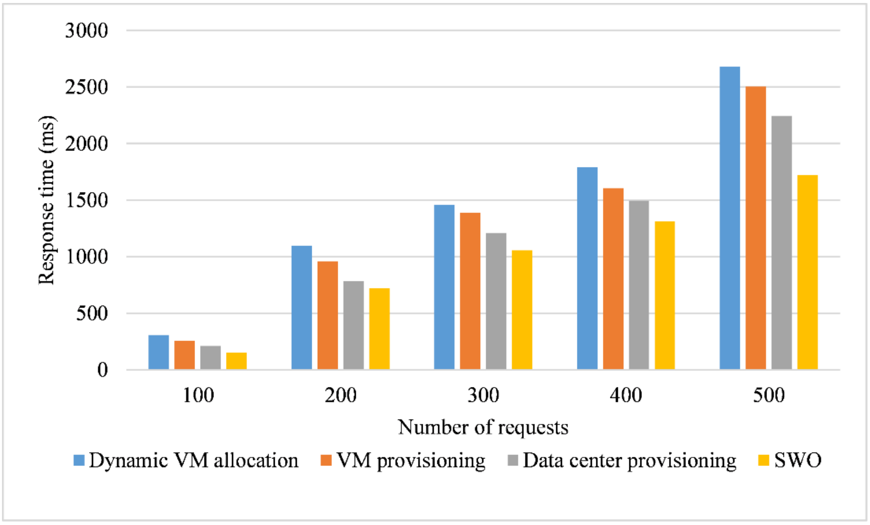

Figure 7: Response time comparison

geographical proximity and energy profiles to minimize power-hungry operations. SWO further optimizes resource usage by distributing tasks to servers operating within their optimal energy range. These results demonstrate that the proposed approach can satisfy the growing computational demands while minimizing operational costs and environmental impacts.

The power consumption is shown in Figure 6, which further verifies the efficiency of the proposed method. The power consumption for the proposed method at 500 requests was approximately 4,800 W. In contrast, for Dynamic VM Allocation, the power consumption was 5,500 W. This trend of reduction for every workload indicates the effectiveness of SWO in maintaining a perfect balance between energy efficiency and active/idle resources.

Figure 7 illustrates the response time of the proposed method compared to Dynamic VM Allocation, VM Provisioning, and Data Center Provisioning approaches for varying numbers of requests. The proposed SWO method demonstrated the lowest response time across all workload levels, indicating its efficiency in handling service requests. For instance, it can be seen that for the highest workload of 500 requests, the response time using the SWO method is significantly lower than that of the other approaches, and the latency is reduced by approximately 20% compared to Dynamic VM Allocation. Such performance could be attributed to how the hierarchical resource allocation strategy adopted in SWO optimizes resource selection and allocation, aiming to avoid bottlenecks.

Table 2: Statistical comparison of the proposed SWO-based method with baseline approaches

Metric	Compared methods	Mean difference	Std. Dev.	P-value
Service response time	SWO vs. Dynamic VM	-1.87 s	0.41 s	0.0012
Energy consumption	SWO vs. VM Provisioning	-1.63 kJ	0.38 kJ	0.0036

To validate the performance differences between our proposed method and existing strategies, we conducted paired two-tailed t-tests at a 95% confidence level for each performance metric. Table 2 reports the average improvements, standard deviations, and p-values for service response time and energy consumption across five runs.

The tests demonstrate that the predictive hierarchical resource allocation scheme based on SWO is scalable in terms of problem size. With the increase in the number of service requests, the runtime demonstrates almost linear behavior, indicating stable and manageable performance even with large workloads. Additionally, the increase in the number of cloud nodes has a moderate impact on runtime. This is because the CMM conducts the pre-selection of suitable resources, which significantly reduces the search space before the GMM applies the optimality algorithm based on the SWO. Overall, the synergistic operation of the systems ensures the responsiveness and energy efficiency of the computer systems with large-scale, dynamic clouds.

5 Discussion

The experimental results indicate that the designed two-tier SWO-based approach achieves significant improvements in service responsiveness and energy efficiency. Compared to existing methods, the designed approach achieves an average response time improvement of 25% and an energy consumption reduction of 26% under peak loads. This efficiency is attributed to the hierarchical design, which minimizes the resource candidates before the invocation of the SWO, thus minimizing the solution search space and avoiding resource contention.

Unlike others that heavily rely on mathematical programming or deep learning models, our method mitigates the high computational intensity by restricting the execution of the SWO to a pre-filtered candidate pool.

SWO outperforms its peer metaheuristics due to its adaptive global exploration and exploitation trade-off, based on the natural dynamics of hunting and mating behaviors.

Despite these strengths, the proposed method has several limitations. First, the SWO algorithm's performance can be sensitive to workload variability; sudden spikes in user demand may affect convergence quality unless it is dynamically calibrated. Second, although the SWO exhibits faster convergence than several traditional metaheuristics, it may still require tuning for real-time or low-latency applications. Third, the current implementation is tailored to homogeneous cloud architectures; its portability to federated or hybrid cloud platforms remains untested.

One notable trade-off of our solution is the incremental computational cost attributable to the SWO's iterative optimization process. Although lightweight compared to deep learning methods, it still incurs moderate runtime overhead, which may become suboptimal in ultra-time-critical applications. Nonetheless, the cost is fairly justified due to the consistent improvements in energy and QoS performance.

6 Conclusion

The presented study introduces an intelligent resource provisioning paradigm capable of enhancing energy efficiency while ensuring robust QoS in cloud infrastructure. Due to the larger environmental footprint of data centers and the growth of cloud infrastructure, the issue of unsustainable and dumb resource management is more acute than ever. To that end, we developed a two-tiered hierarchical structure that mixes the CMM and the GMM. The CMM actively filters and filters down suitable cloud resources based on the real-time parameters of the systems, namely queue length, response time, and location, and submits them to the GMM. Based on the predator-prey behavior metaheuristic algorithm, the GMM fine-tunes the selection and optimizes it according to energy efficiency. This collaborative process facilitates effective VM placement and adaptive resource provisioning. By utilizing both real-time perception of the system and intelligent optimizations, the introduced paradigm adaptively responds to variations in workload and infrastructure state. This results in reduced power consumption and improved response times across the entire cloud infrastructure.

Our results indicate a promising method for integrating nature-inspired algorithms into hierarchical management structures that can address the dual problems of QoS and energy efficiency in cloud computing. With the ongoing growth in size, cloud infrastructures will need to adopt new approaches to reduce their environmental footprint while maintaining reliable and efficient service provisioning. Future work could include determining the scalability of this approach when transitioning to an even more extensive and diverse cloud environment, as well as its adaptability to evolving technologies such as edge computing and IoT.

While the devised method is mainly applied to centralized cloud infrastructures, its architectural principles and optimization model are, at the core, extensible to the fog and edge computing paradigm. The hierarchical structure (CMM-GMM) is applicable by introducing intermediary layers that represent the fog nodes or edge devices. Additionally, the SWO can potentially optimize resource allocation among heterogeneous devices by adjusting its fitness function to consider edge-specific factors, such as latency, mobility, and battery limitations. Considering the increasing convergence of fog, cloud, and edge technologies, the proposed method provides a practical foundation for future distributed resource management systems.

Conflict of interest

The authors declare that they have no conflicts of interest.

References

[1] M. Gams and T. Kolenik, "Relations between electronics, artificial intelligence and information society through information society rules," *Electronics,* vol. 10, no. 4, p. 514, 2021, doi: https://doi.org/10.3390/electronics10040514.

[2] J. Ma, C. Zhu, Y. Fu, H. Zhang, and W. Xiong, "Cloud Computing Resource Scheduling Method Based on Optimization Theory," *Informatica,* vol. 48, no. 23, 2024, doi: https://doi.org/10.31449/inf.v48i23.6901.

[3] B. Pourghebleh, A. Aghaei 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,* vol. 24, no. 3, pp. 2673-2696, 2021, doi: https://doi.org/10.1007/s10586-021-03294-4.

[4] M. B. Bagherabad, E. Rivandi, and M. J. Mehr, "Machine Learning for Analyzing Effects of Various Factors on Business Economic," *Authorea Preprints,* 2025, doi: https://doi.org/10.36227/techrxiv.174429010.09842200/v1.

[5] M. Ahmadi *et al.*, "Optimal allocation of EVs parking lots and DG in micro grid using two-stage GA-PSO," *The Journal of Engineering,* vol. 2023, no. 2, p. e12237, 2023, doi: https://doi.org/10.1049/tje2.12237.

[6] A. Kermani *et al.*, "Energy management system for smart grid in the presence of energy storage and photovoltaic systems," *International Journal of Photoenergy,* vol. 2023, no. 1, p. 5749756, 2023, doi: https://doi.org/10.1155/2023/5749756.

[7] E. Rivandi and R. Jamili Oskouie, "A Novel Approach for Developing Intrusion Detection Systems in Mobile Social Networks," *Available at SSRN 5174811,* 2024, doi: https://dx.doi.org/10.2139/ssrn.5174811.

[8] B. Pourghebleh and N. J. Navimipour, "Data aggregation mechanisms in the Internet of things: A systematic review of the literature and recommendations for future research," *Journal of Network and Computer Applications,* vol. 97, pp. 23-34, 2017, doi: https://doi.org/10.1016/j.jnca.2017.08.006.

[9] J. Guan, "Enhanced Network Security Hybrid Cloud Workflow Scheduling Using Levy-Optimized Slime Mould Algorithm," *Informatica,* vol. 49, no. 18, 2025, doi: https://doi.org/10.31449/inf.v49i18.7327.

[10] V. Hayyolalam, B. Pourghebleh, M. R. Chehrehzad, and A. A. Pourhaji Kazem, "Single-objective service composition methods in cloud manufacturing systems: Recent techniques, classification, and future trends," *Concurrency and Computation: Practice and Experience,* vol. 34, no. 5, p. e6698, 2022, doi: https://doi.org/10.1002/cpe.6698.

[11] O. K. J. Mohammad and B. M. Salih, "Improving Task Scheduling In Cloud Datacenters By Implementation Of An Intelligent Scheduling Algorithm," *Informatica,* vol. 48, no. 10, 2024, doi: https://doi.org/10.31449/inf.v48i10.5843.

[12] J. Ma, C. Zhu, Y. Fu, H. Zhang, and W. Xiong, "Reliable Task Scheduling in Cloud Computing Using Optimization Techniques for Fault Tolerance," *Informatica,* vol. 48, no. 23, pp. 159-170, 2024, doi: https://doi.org/10.31449/inf.v48i23.6901.

[13] Y. Mehor, M. Rebbah, and O. Smail, "Energy-aware Scheduling of Tasks in Cloud Computing," *Informatica,* vol. 48, no. 16, 2024, doi: https://doi.org/10.31449/inf.v48i16.5741.

[14] V. Hayyolalam, B. Pourghebleh, A. A. Pourhaji Kazem, and A. Ghaffari, "Exploring the state-of-the-art service composition approaches in cloud manufacturing systems to enhance upcoming techniques," *The international journal of advanced manufacturing technology,* vol. 105, no. 1, pp. 471-498, 2019, doi: https://doi.org/10.1007/s00170-019-04213-z.

[15] S. R. Swain, A. Parashar, A. K. Singh, and C. N. Lee, "An intelligent virtual machine allocation optimization model for energy-efficient and reliable cloud environment," *The Journal of Supercomputing,* vol. 81, no. 1, pp. 1-26, 2025, doi: https://doi.org/10.1007/s11227-024-06734-1.

[16] F. J. Maldonado-Carrascosa, S. García-Galán, M. Valverde-Ibáñez, T. Marciniak, M. Szczerska, and N. Ruiz-Reyes, "Game theory-based virtual machine migration for energy sustainability in cloud data centers," *Applied Energy,* vol. 372, p. 123798, 2024, doi: https://doi.org/10.1016/j.apenergy.2024.123798.

[17] R. Buyya, S. Ilager, and P. Arroba, "Energy-efficiency and sustainability in new generation cloud computing: a vision and directions for integrated management of data centre resources and workloads," *Software: Practice and Experience,* vol. 54, no. 1, pp. 24-38, 2024, doi: https://doi.org/10.1002/spe.3248.

[18] F. J. Maldonado Carrascosa, D. Seddiki, A. Jiménez Sánchez, S. García Galán, M. Valverde Ibáñez, and A. Marchewka, "Multi-objective optimization of virtual machine migration among cloud data centers," *Soft Computing,* vol. 28, no. 20, pp. 12043-12060, 2024, doi: https://doi.org/10.1007/s00500-024-09950-2.

[19] K. K. Nguyen and M. Cheriet, "Environment-aware virtual slice provisioning in green cloud environment," *IEEE Transactions on services computing,* vol. 8, no. 3, pp. 507-519, 2014, doi: https://doi.org/10.1109/TSC.2014.2362544.

[20] Y. Yang, X. Chang, J. Liu, and L. Li, "Towards robust green virtual cloud data center provisioning," *IEEE Transactions on Cloud Computing,* vol. 5, no. 2, pp. 168-181, 2015, doi: https://doi.org/10.1109/TCC.2015.2459704.

[21] A. Khosravi, L. L. Andrew, and R. Buyya, "Dynamic vm placement method for minimizing energy and carbon cost in geographically distributed cloud data centers," *IEEE Transactions on Sustainable Computing,* vol. 2, no. 2, pp. 183-196, 2017, doi: https://doi.org/10.1109/TSUSC.2017.2709980.

[22] X. Wu, H. Wang, D. Wei, and M. Shi, "ANFIS with natural language processing and gray relational analysis based cloud computing framework for real time energy efficient resource allocation," *Computer communications,* vol. 150, pp. 122-130, 2020, doi: https://doi.org/10.1016/j.comcom.2019.11.015.

[23] F. N. Al-Wesabi, M. Obayya, M. A. Hamza, J. S. Alzahrani, D. Gupta, and S. Kumar, "Energy aware resource optimization using unified metaheuristic optimization algorithm allocation for cloud computing environment," *Sustainable Computing: Informatics and Systems,* vol. 35, p. 100686, 2022, doi: https://doi.org/10.1016/j.suscom.2022.100686.

[24] K.-Y. Tai, F. Y.-S. Lin, and C.-H. Hsiao, "An integrated optimization-based algorithm for energy efficiency and resource allocation in heterogeneous cloud computing centers," *Ieee Access,* vol. 11, pp. 53418-53428, 2023, doi: https://doi.org/10.1109/ACCESS.2023.3280930.

[25] S. Sharma and P. S. Rawat, "Efficient resource allocation in cloud environment using SHO-ANN-based hybrid approach," *Sustainable Operations and Computers,* vol. 5, pp. 141-155, 2024, doi: https://doi.org/10.1016/j.susoc.2024.07.001.

[26] S. Gurusamy and R. Selvaraj, "Resource allocation with efficient task scheduling in cloud computing using hierarchical auto-associative polynomial convolutional neural network," *Expert Systems with Applications,* vol. 249, p. 123554, 2024, doi: https://doi.org/10.1016/j.eswa.2024.123554.