Received: 9 March 2021 | Revised: 30 June 2021 | Accepted: 1 August 2021

IET Communications

DOI: 10.1049/cmu2.12274

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

A multi-objective method for virtual machines allocation in cloud data centres using an improved grey wolf optimization algorithm

Masoud Hashemi[1] | **Danial Javaheri**[2] | **Parisa Sabbagh**[3] | **Behdad Arandian**[4] | **Karlo Abnoosian**[5]

[1] Zabol University, Zabol, Iran

[2] Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

[3] DISA-MIS Department, University of Salerno, Fisciano, Italy

[4] Department of Electrical Engineering, Dolatabad Branch, Islamic Azad University, Isfahan, Iran

[5] Department of Statistics, Science and Research Branch, Islamic Azad University, Tehran, Iran

Correspondence
Masoud Hashemi, Zabol University, Zabol, Iran.
Email: Masoudhashemi60@yahoo.com
Karlo Abnoosian, Department of Statistics, Science and Research Branch, Islamic Azad University, Tehran, Iran.
Email: karlo.abnoosian@srbiau.ac.ir

Abstract

Cloud computing is a rapidly evolving computational technology. It is a distributed computational system that offers dynamically scaled computational resources, such as processing power, storage, and applications, delivered as a service through the Internet. Virtual machines (VMs) allocation is known as one of the most significant problems in cloud computing. It aims to find a suitable location for VMs on physical machines (PMs) to attain predefined aims. So, the main purpose is to reduce energy consumption and improve resource utilization. Because the VM allocation issue is NP-hard, meta-heuristic and heuristic methods are frequently utilized to address it. This paper presents an energy-aware VM allocation method using the improved grey wolf optimization (IGWO) algorithm. Our key goals are to decrease both energy consumption and allocation time. The simulation outcomes from the MATLAB simulator approve the excellence of the algorithm compared to previous works.

1 | INTRODUCTION

Cloud computing has become a popular service because of the rapid advancement of information and communication technology [1, 2]. Cloud computing and data-storage systems use several technologies to link and manage required resources across several devices [3, 4]. It is a concept for supplying and distributing on-demand network accessibility to a public pool of computing resources with little administrative effort and/or service provider interaction [5]. Cloud computing has developed as a modern paradigm to offer clients a service through the Internet to fulfil many objectives, such as controlling the robots and classification [6–9]. It is a location-independent, virtualized, and on-demand pay-per-use pricing model to increase productivity and maximize resource usage [10].

Cloud computing has significantly altered the conventional ownership structure (where computational powers were owned in the past) to the present subscription model [11]. It offers pay-as-you-go accessibility to elastic resources in the shape of services, with a pay-as-you-go model based on actual resource use [12]. So far, the deployment of clouds has been fuelled by three primary service models: platform, software, and infrastructure as a service (PaaS, SaaS, and IaaS) [13]. The greatest degree of abstraction is SaaS, which allows customers to reach applications hosted in cloud data centres (CDC) through the Internet. For example, it has enabled enterprises to have more flexible access to the software by allowing unrestricted and on-demand access to various ready-to-use apps. Corporations may also reduce direct or internal expenditures like license fees and IT infrastructure preservation by using SaaS. PaaS is designed for consumers that want more control over their IT resources, and it provides a basis for building and deploying cloud apps, including auto-scaling and programming models. For instance, it has made it simple for designers to construct apps that utilize the cloud's elastic resource paradigm. Eventually, IaaS provides access to computational resources, such as virtual machines (VMs) and storage space, by leasing them [14]. This layer has served as the cornerstone of cloud computing and the base for PaaS and SaaS. It has performed this work by allowing subscribers to utilize the information technology (IT) infrastructure only when they require it to alter the number of resources utilized on a flexible basis and spend only for what is utilized, all while totally controlling the resources [15].

Virtualization is known as a fundamental cloud computing technology and offers an efficient way to manage dynamic resources in cloud data centres using some well-known methods, such as clustering [16, 17]. The VMs are separated from each other, and they can act as a whole system to execute users' applications [18]. Also, the physical machine (PM) as a server or host of VMs provides all required VM resources, including network bandwidth, storage, memory, and CPU [19]. Therefore, the power utilization of cloud data centres is increasing as cloud computing becomes more prevalent. Surprisingly, data centre power is squandered for various reasons, including server use, network equipment, and inefficient data centre cooling systems [20, 21]. As a result, energy efficiency has gained significance in wireless and cloud data centres [22]. Lately, different works have been proposed on energy efficiency, which is categorized into three main groups, including task scheduling, migration, and resource allocation [23, 24]. Consolidation of VMs is a method of making intelligent and effective utilization of cloud resources. The VM allocation issue is one of the most difficult aspects of VM consolidation. It is defined as finding the appropriate PM for a VM that reduces the running PMs or hosts in the cloud data centres [25].

Academic investigators have focused on VM allocation as an NP-hard issue that has been addressed using various metaheuristic and heuristic approaches [26]. Therefore, different objectives have been considered to improve load balance, reduce cost and network consumption, mitigate SLA violations, increase energy efficiency, and maximize resource utilization [27]. There are many nature-inspired algorithms which have been commonly used recently in many domains, such as Harris hawks [28, 29], moth flame [30–32], fruit fly [33, 34], whale [35, 36], bacterial foraging [37], grasshopper [38] and ant colony [39, 40] optimization algorithms. In the present investigation, the improved grey wolf optimization algorithm (IGWO) was used to tackle the issue of VM allocation. This improved algorithm is based on changes in wolf populations and the effect of each wolf's fitness level. It has solved optimization problems in many different fields effectively. The main goals are to reduce energy consumption and VMs allocation time in the cloud using the IGWO algorithm. Also, the key contributions to the present article are as follows:

- Offering an improved algorithm to address the virtual machine allocation issue at cloud data centres;
- Improving the VM's energy consumption in an environment of cloud;
- Improving the VM's allocation time in an environment of the cloud.

The remainder of the present article is arranged as follows: The second section examines the relevant literature. The suggested method is explained in Section 3. In Section4, the simulation outcomes are reported. Eventually, Section 5 brings the study to a conclusion.

2 | RELATED WORK

Cloud computing has emerged as the latest technology in the computation domain [41]. Cloud service providers (CSPs) have access to high-performance computing resources known as data centres (DCs). Users are served by CSPs using VMs [42]. VMs with a high memory has been popular lately [43]. A host has deployed several heterogeneous VMs on a parallel computing platform termed a data centre in the cloud computing platform [44]. Virtualization technology grants VM subscribers administrative access within the guest operating system, allowing them to adjust their runtime resources to meet their individual requirements [45]. The latest literature on VM migration has concentrated chiefly on the choice of destination data centres or destination servers based on migration time optimization and application quality of service (QoS) [46, 47]. The following are some studies on clouds and VMs that have been published in the literature.

Yadav et al. [48] introduced GradCent for managing overloaded hosts in cloud data centres. Utilizing an actual CPU workload, this approach is employed to generate an upper CPU usage threshold to discover overloaded hosts. Furthermore, they presented the minimum size utilization (MSU) dynamic VM selection technique for picking VMs from an overloaded host for VM consolidation. Finally, they combined real-world workload measurements from numerous PlanetLab VMs with CloudSim simulations. Compared to baseline methods, the suggested algorithms reduced energy usage and SLA violations by 23 percent and 27.5 percent on average, respectively.

Yadav et al. [49] introduced adaptive heuristic techniques for overloaded host identification and VM choosing, including least medial square regression and minimal usage prognostication. The suggested VM selection technique takes into account the sorts of applications being executed and their CPU use at various time intervals across the VMs. The suggested methods are validated by utilizing the CloudSim simulator and a PlanetLab workload trace. Compared to earlier techniques for overloaded host detection and VM selection, the findings indicated that these techniques significantly reduce the CDC's electric energy usage.

Satpathy et al. [50] introduced the VRMap model, which discovered an optimum remapping strategy that balances remapping costs and migration overheads without sacrificing service quality. To find the best solution, VRMap used a genetic-metaheuristic. The findings indicated that while VRMap performs similarly in remapping costs, it outperforms other competitions in downtime and migration time.

In cloud data centres, Naik et al. [51] suggested effective VM scheduling using the multi-objective krill herd optimization selection method. For the VMs, targets like power usage, load volume, and resource waste are assessed, and the entropy is computed for the determined objectives. Then, prioritized tasks are assigned to the VMs based on the entropy value. According to the findings, the suggested entropy-based scheduling outperformed some algorithms.

Another energy-efficient VM allocation method was proposed in [52] using an improved ant colony optimization (ACO) algorithm. Reducing the energy consumption and communication cost over traffic-aware data centre networks are the main aims of this work. Nevertheless, it just checks out CPU power usage, which is the most prevalent utilization. Moreover, it does not evaluate VM algorithms with accurate trace data of the cloud platform.

Also, considering optimizing the CPU utilization while decreasing energy consumption, the researchers in [53] addressed the VM placement problem. They examined the performance of four common multi-objective evolutionary algorithms utilizing the Planet Lab dataset and Cloudsim in this regard. According to simulation outcomes, the non-dominated sorting genetic algorithm II (NSGA-II) exhibits improved performance regarding SLA violation, SLA, and energy usage. Furthermore, in terms of an arithmetic mean of CPU usage, NSGAII performs better. Furthermore, eMOEA outperforms the competition in terms of CPU utilization standard deviation.

Satpathy et al. [54] suggested a resource-aware solution based on a metaheuristic crow search algorithm (CSA) to integrate a large number of VMs onto minimum DCs while maximizing data centre usage to match the service level agreement (SLA) and desired QoS. To achieve the required results, they presented two separate techniques: (i) greedy crow search (GCS) and (ii) traveling salesman problem-based hybrid crow search (TSPCS). They compared their suggested technique to the conventional first fit (FF) methodology to assess its performance, and their new approaches greatly outperformed the classical technique.

A multi-objective VM placement technique is offered in [55], considering the different necessities of cloud suppliers and clients. It aims to minimize load variance and energy consumption, improve the robustness of PMs, and maximize resource utilization. An energy-efficient knee point-driven evolutionary algorithm (EEKPDEA) is proposed to handle the multi-objective optimization issue presented in this paper. In addition, to enhance population initialization in EEKPDEA, an energy-efficient-oriented population initialization strategy (EEPIS) was suggested. The simulation outcomes indicated that the suggested model offers good performance in load balancing, energy consumption, and robustness. However, it ignores the abnormal loads on PMs, such as overhead.

To increase a CSP's profit, Addya et al. [56] suggested a strategy termed maximum VM placement (MVMP) with low energy consumption. It is a bi-objective optimization issue that is tackled with the simulated annealing (SA) method. In the field of strategic VM placement, their MVMP method is compared against five modern algorithms: Hybrid GA (HGA), Marotta and Avallone (MA) method, first-fit decreasing (FFD), modified best-fit decreasing (MBFD), and random deployment. MVMP outperforms the Marotta and Avallone (MA) technique, MBFD, HGA, FFD, and random placement regarding the number of servers utilized, profit, energy consumption, and execution time. Furthermore, the MVMP's scalability is tested by utilizing two scenarios: (i) a fixed number of VMs and (ii) a fixed number of servers.

An optimized resource allocation algorithm at data centres is proposed by Sharma and Reddy [57], in which GA and dynamic voltage frequency scaling (DVFS) techniques are combined. The frequency of PMs is modified in DVFS depending on the workload, and GA is utilized to allocate VMs in an energy-efficient way. The suggested work aimed to decrease the power utilization of data centres, increase the convergence of the solutions, and utilize resources. The simulation outcomes have shown that the proposed model consumes 22.4% less energy than the compared method. However, notwithstanding the advancements in the proposed algorithm, the power consumption of data centres has not been reduced since the DVFS method is restricted to just the CPU.

Moreover, an energy-efficient VM placement using the biogeography-based optimization (BBO) algorithm is suggested in [58]. The suggested model is simulated using the MATLAB simulator and compared to GA. In terms of energy usage, the findings demonstrate that the suggested model outperforms the GA. Nevertheless, it does not study the QoS, particularly the SLA violation. In addition, it only concentrates on the new VM allocation.

The grouping GA independently does not obtain an efficient solution for the VM allocation issue. So, in the suggested model in [59], the researchers improved this algorithm by presenting an effective method to encode and produce novel solving methods. The vector-packing problem has been used to model the VM placement issue to improve resource usage, decrease the number of used servers, and reduce energy usage. In addition, the suggested crossover process has improved operational efficiency, reduced resource wastage and power consumption, and managed the various collections of the VMs.

Table 1 compares the existing techniques for allocating the virtual machines at data centres and shows the advantages and disadvantages of each method. For solving these problems, virtual machine allocation is suggested at data centres using the improved grey wolf optimization algorithm, which will be followed by the minimum energy consumption and the minimum virtual machine allocation time.

3 | PROPOSED METHOD

The present section initially explains the issue and then describes the energy and time models. Next, Section 3.3 discusses the proposed algorithm. Finally, in Sections 3–4, the suggested algorithm is presented.

TABLE 1 Comparison of available methods

Article	Method	Main features
Yadav et al. [48]	Using a GradCent algorithm to identify overloaded hosts utilizing a real CPU workload Proposing MSU algorithm	• Minimizing energy consumption • Minimizing SLA violation
Yadav et al. [49]	Proposing adaptive heuristic algorithms, including least medial square regression	• Improving overloaded host detection • Enhancing usage prognostication for VM selection from overloaded hosts
Satpathy et al. [50]	Proposing a VRMap model that found an optimal remapping plan	• Balancing the remapping expenses and migration overheads with no compromising service quality
Naik et al. [51]	Proposing effective scheduling (entropy-based krill herd optimization) with multi-objective VM selection in cloud data centers	• Minimizing completion cloud • Improving cloud task partitioning scheduling • Improving round-robin techniques
Wei et al. [52]	VM allocation using an improved ACO algorithm	• Reducing communication cost over traffic-aware data center networks • Not evaluating VM algorithms with real trace data of the cloud platform
López-Pires and Barán [53]	Using NSGA-II for VM placement problem	• Optimizing the CPU utilization • Decreasing energy consumption
Satpathy et al. [54]	Proposing a resource-aware method based on a met heuristic GCS	• Improving QoS with maximum data centre utilization • Consolidating a large number of VMs on minimal DCs to meet the SLA
Ye et al. [55]	Using EEKPDEA for multi-objective VM placement technique	• Minimizing load variance • Minimizing energy consumption • Improving the robustness of PMs • Maximize the resource utilization
Addya et al. [56]	proposing the MVMP technique to enhance the profit attained by a CSP	• Improving energy consumption • Improving profit • Improving execution time
Sharma and Reddy [57]	Combining GA and DVFS technique for resource allocation at data centres	• Decreasing the power usage of data centres • Increasing the convergence of the solutions • Improving resource utilization
Ali and Lee [58]	Using the BBO algorithm for VM placement	• Reducing energy consumption • Only focusing on the fresh VM allocation
Jamali and Malektaji [59]	Using the vector packing issue to model the problem of VM placement	• Improving the operational efficiency • Reducing the resource wastage • Reducing power consumption

3.1 | Problem description

The placement and diagnosis problems are very popular in many intelligent systems [60–62]. In this paper, we considered that several PMs are linked through a network in cloud data centres. They are assigned to several VMs running a diversity of apps. Figure 1 shows the diagram of the VM allocation problem. In this problem, n VMs with the diverse requested resources are allocated to suitable m PMs to obtain the objectives, including robustness, maximizing resource utilization, and minimizing power consumption. Each PM offers several resource types, namely, CPU, network bandwidth, and memory. The resource capabilities of PMs and the VMs resource requests are generally dissimilar. As shown in Figure 2, a VM allocation outline can be denoted as a vector in which the elements indicate the PMs and the element's value indicates the VMs.

3.2 | Energy consumption model

A linear connection between CPU use and energy utilization defines server energy consumption [63]. Therefore, the CPU utilization of the VMs is calculated by Equation (1).

$$\mathrm{CPU}_{\mathrm{Vm},\ i} = \frac{\mathrm{VM}_{\mathrm{mips},i}}{\mathrm{Host}_{\mathrm{mips},j}} \tag{1}$$

In Equation (1), $\mathrm{VM}_{\mathrm{mips},i}$ indicates the required mips for VM i, and $\mathrm{Host}_{\mathrm{mips},j}$ is the whole capacity of PM j. Also, $\mathrm{CPU}_{\mathrm{Vm},i}$

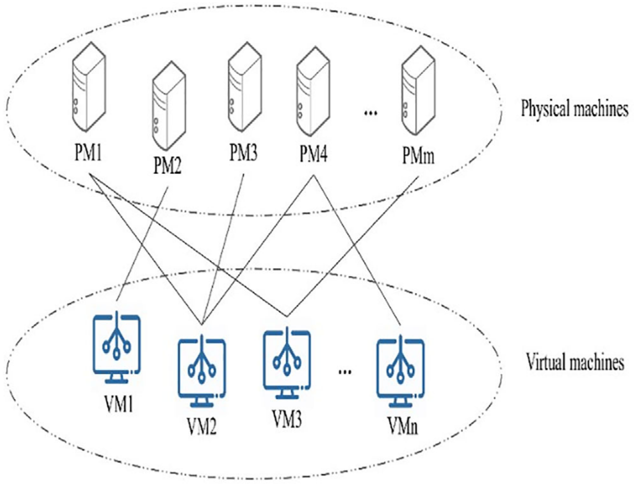

FIGURE 1 Diagram of VM allocation problem

1	2	3	4	5	6	7	8	9	…	n
15	2	14	23	6	36	4	1	17	…	m

FIGURE 2 An array of VM allocation solution

indicates the CPU usage of VM *i*. The CPU utilization of PM *j* is calculated by Equation (2). Moreover, the energy consumption of PM *j* as a function of the CPU utilization is calculated using Equation (3). P_j is the current energy consumption of PM *j*, P_j^{idle} indicates the energy consumption (Watt) of PM *j* in idle, and P_j^{busy} is the maximum energy consumption (Watt) of the PM *j* at 100% CPU utilization.

$$\mathrm{CPU}_{\mathrm{Host},j} = \sum_{i \in r_j} \mathrm{CPU}_{Vm,j} \tag{2}$$

$$P_j = \begin{cases} \left(P_j^{\mathrm{busy}} - P_j^{\mathrm{idle}}\right) * \mathrm{CPU}_{\mathrm{host},j} + P_j^{\mathrm{idle}} \ ; \ \mathrm{CPU}_{\mathrm{host},j} > 0 \\ P_j^{\mathrm{idle}} \ ; \ \mathit{Otherwise} \end{cases} \tag{3}$$

To calculate the system's overall energy consumption during VMs allocation, the energy consumption of all involved hosts in the problem should be considered. Therefore, the main aim of the suggested method in this paper is to decrease the energy consumption of PM that is calculated by Equation (4).

$$\mathrm{Energy} = \sum_{j=1}^{\mathrm{M}} P_j = \sum_{j=1}^{\mathrm{M}} \left[\left(P_j^{\mathrm{busy}} - P_j^{\mathrm{idle}}\right) * \mathrm{CPU}_{\mathrm{host},j} + P_j^{\mathrm{idle}}\right] \tag{4}$$

In Equation (4), the condition of Equation (5) should be considered. The binary variable x_{ij} indicates if VM *i* can be allocated to Pm *j*. If the VM *i* is allocated to the PM *j* then the variable x_{ij} will be 1; otherwise, it will be 0. Generally, VM *i* is allocated to PM *j* when the available resources of PM *j* are greater than the minimum resource requirements of VM *i*.

$$\sum_{j=1}^{\mathrm{M}} x_{ij} = 1 \qquad \forall \mathrm{i} \in N \tag{5}$$

3.3 | Allocation time model

Another goal is to optimize the required time to allocate VMs to the appropriate hosts. The time required to perform the allocation operation depends on the capacity of the hosts. Therefore, the data set required for the time parameter is randomly generated numerically between 0.1 and 10 ms for each source. Then, to calculate the total allocation time, the time associated with the involved hosts in the process must be summed together according to Equation (6).

$$\mathrm{Time} = \sum_{i=1}^{n} T_i \tag{6}$$

In this study, there are two goals for optimization, both of which must be minimized. However, these two parameters have different units and cannot be combined. Therefore, to calculate the value of fitness, these values must be normalized and added together using the user's priority coefficients. The coefficients must be defined in such a way that their sum is equal to 1 ($\sum_{i=1}^{2} w_i = 1$). The final value of Equation (7) is obtained. In this equation, the two parameters of time and energy are obtained from the equations described earlier and then normalized and summed together by the mentioned coefficients (w_1 & w_2), which results in final fitness at the end:

$$\mathrm{Fitness} = w_1 \times \mathrm{Energy} + w_2 \times \mathrm{Time} \tag{7}$$

$$= \frac{\mathrm{Energy_{max}} - \mathrm{Energy}}{\mathrm{Energy_{max}} - \mathrm{Energy_{min}}} \tag{8}$$

$$\mathrm{Time} = \frac{\mathrm{Time_{max}} - \mathrm{Time}}{\mathrm{Time_{max}} - \mathrm{Time_{min}}} \tag{9}$$

4 | GREY WOLF ALGORITHM

Inspired by the social life and hunting of grey wolves, the GWO algorithm uses four types of wolves to simulate leadership hierarchies [64, 65]. Grey wolves have a well-defined social hierarchy [66]. The group's leaders are Alpha, a male and female duo. Alpha is primarily in charge of deciding about hunting, sleeping locations, and waking times, among other things. The group is dictated to by Alpha's choices [67]. On the other hand, Alpha has been shown to follow the other wolves in the group in a democratic manner. The whole group acknowledges the Alpha by holding its tail down in group aggregation. The group must follow Alpha's commands. Only the group's Alpha wolves are able to pick their partners. Surprisingly, Alpha is not always the group's most vital member, but it is the greatest at managing the group. It demonstrates that a group's structure and order are more essential than its power [68].

The second level of the grey wolf hierarchy is beta. Beta wolf is Alpha's consultant and organizer of the group. Beta implements Alpha's orders across the group and reports back to Alpha. Omega is the grey wolf with the lowest rank. Omega takes on the character of the helpless victim. They were the final wolf group to be killed. It is termed subordinate or Delta if the wolf is not Omega, Beta, or Alpha. The Delta wolf must provide reports to Alpha and Beta but dominates Omega.

Besides the social hierarchy, hunting of grey wolves has three phases: tracking, chasing, and approaching the bait [69]. To simulate the wolves' social hierarchy, we assume Alpha to be the best option and Beta and Delta to be the second and third-best alternatives. The remaining candidate solutions are categorized as Omega. Thus, Alpha, Beta, and Delta are the three groups that drive the optimization, and the fourth group follows them. Wolf siege behaviour modelling uses Equations (10) and (11):

$$\vec{D} = \left| \vec{C}.\vec{X}_p(t) - \vec{X}(t) \right| \tag{10}$$

$$\vec{X}_{(t+1)} = \vec{X}_p(t) - \vec{A}.\vec{D} \tag{11}$$

where t is the number of current iterations, A and C are the coefficient vectors, X_p is the hunting position vector, and X is the position vector of a wolf. Equations (12) and (13) are used to calculate the vectors A and C [70]:

$$\vec{A} = 2\vec{a}.\vec{r}_1 - \vec{a} \tag{12}$$

$$\vec{c} = 2\vec{r}_2 \tag{13}$$

The vector a is reduced linearly from 2 to 0 within the exploitation and exploration phases iteration. r is a random vector between 0 and 1. Due to the randomness of vectors r_1 and r_2, wolves can randomly change their position inside the bait-enclosed space using Equations (14) and (15). An n-dimensional search space may be used to apply the same approach. In this situation, the grey wolves circle the best solution found in dimensions greater than the cube's dimensions. Alpha often leads grey wolf hunting. Sometimes Delta and Beta wolves also hunt. To replicate this behaviour, the three best answers are kept, and the other search agents are compelled to update their positions in accordance with the best search agents' positions using Equation (16):

$$\vec{D}_\delta = \left| \vec{C}_3.\vec{X}_\delta - \vec{X} \right| \quad \vec{D}_\beta = \left| \vec{C}_2.\vec{X}_\beta - \vec{X} \right| \quad \vec{D}_\alpha = \left| \vec{C}_1 - \vec{X}_\alpha - \vec{X} \right| \tag{14}$$

$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1.\left(\vec{D}_\alpha\right) \quad \vec{X}_2 = \vec{X}_\beta - \vec{A}_2.\left(\vec{D}_\beta\right) \quad \vec{X}_3 = \vec{X}_\delta - \vec{A}_\delta.\left(\vec{D}_\delta\right) \tag{15}$$

$$\vec{X}_{(t+1)} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{16}$$

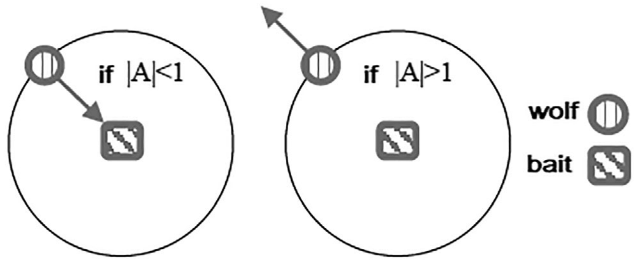

FIGURE 3 Exploration vs. exploitation phase

The implementation of the exploitation or attack phase [71], which occurs if the bait stops, is done by reducing the variable a value from 2 to 1. The value of A is also dependent on a, so it decreases. As the value of A drops, wolves are forced to attack the bait. An exploration phase is also provided to prevent getting stuck in the local minimum. Wolves distance themselves from each other in search of bait and approach and work together to attack. A is used with random values above 1 or below −1 [72]. Figure 3 illustrates this point.

Another influential component of the process is the value of C. The value of this vector is a random number among [0, 2]. This random value affects the position of the bait in determining distance, intensity ($C > 1$), or weakness ($C < 1$). This vector can also be thought of as the influence of natural impediments that hinder bait from being approached.

4.1 | Steps of the proposed method

The population of grey wolves progresses towards ideal replies, or Alpha, Beta, and Delta wolves, in the GWO algorithm. The top Alpha particle is found in each iteration using the fitness function to select the best particles. Each dimension of the new location is equal to the average of the corresponding dimensions of the superior particles, which were fully described in the previous section. Particle movement at any stage is done regardless of the fitness degree; in other words, it is non-greedy. This algorithm aims to change the initial population to have a population with more fitness. For this purpose, Omega wolves are identified and removed from the original population at each iteration. To have a population with high fitness, Omega wolves were replaced with the best particles or with other wolves randomly selected from the population in order to converge faster to the response. Random particles can be of any fitness degree. In the first iterations, the wolves are more likely to be replaced randomly, and in the last iterations, it is more likely to be based on fitness. Also, in each iteration of particle displacement, the fitness of the novel position is compared to the fitness of the last suitable location with the highest fitness. If the fitness of the novel position is high, displacement will occur. Otherwise, the location with the most fitness will be the new place. The pseudo-code of this algorithm can be seen in Figure 4, and it will be used as IGWO from now on.

```
1. Set the initial values of the population size n, parameter a, coefficient vectors A and C, and the maximum number of iterations Maxiter.
2. Set t = 0.
3. For (i = 1 : n) do
4. Generate an initial population Xi (t) randomly.
5. Evaluate the fitness function of each search agent (solution) f (Xi).
6. End for
7. Assign the values of the first, second and the third best solution Xα,Xβ and Xδ , respectively.
8. r is random number.
9. If (r<1-t/ Maxiter)
10. Replace the values of the forth solution with Random Selected solution, respectively.
11. Else Replace the values of the forth solution with Xα solution, respectively.
12. End if
13. Repeat
14. For (i = 1 : n) do
15. Set temp = Calculate each search agent in the population as shown in Eq. (7).
16. If (fitness (temp)>fitness(pBest))
17. Set pBest=temp.
18. Set newPosition=pBest.
19. End if
20. Decrease the parameter a from 2 to 0.
21. Update the coefficients A, C as shown in Eq. (3) and (4), respectively.
22. Evaluate the fitness function of each search agent (vector) f(Xi).
23. End for
24. Update the vectors Xα, Xβ and Xδ .
25. Set t = t + 1.
26. Until (t ≥ Maxiter). {Termination criteria are satisfied}
27. Print the best solution Xα.
```

FIGURE 4 Pseudo-code of the proposed algorithm

4.2 | Complexity of the proposed algorithm

The following formula may be used to estimate the time complexity of the described algorithm.

Let the highest number of iterations be t. In each iteration, the fitness value is calculated with a time complexity of O(OF) [73]. If there are P solutions, the total complexity will be O(t (P * OF)).

5 | PERFORMANCE EVALUATION

The present section compares the performance of the suggested strategy with the PSO algorithm [74], GA [57], and GCS [54]. Also, several investigations are conducted to assess the proposed algorithm's correctness, performance, and properties. The algorithm is simulated using the MATLAB simulator 2016b. Table 2 shows the important variables and parameters used for simulation. In order to assess the suggested algorithm, three scenarios have been considered, which are specified in Table 3. The population size and maximum iteration in all scenarios have been taken into account, respectively 50 and 500.

Figures 5–7 illustrates the fitness value of the algorithms in 3 scenarios (the number of tasks is 100, 200, and 300). The results represent that our algorithm outperforms other algorithms.

TABLE 2 Experimental parameters

Parameters	Values
PMs utilization	[50–800]
VMs utilization	[20–300]
Amount of energy consumption (Watt) in 100% CPU utilization	[100–1000]
Amount of time to use PMs (ms)	[0.1–10]
Number of tasks	100, 200, 300
Primary energy	0.5–1 J

TABLE 3 Details of scenarios

	First scenario	Second scenario	Third scenario
The number of PMs	10	50	100
The number of VMs	50	300	500

The confidence interval is calculated by Equation (17) [75]:

$$\mathrm{CI} = \bar{x} \pm z\frac{s}{\sqrt{n}} \tag{17}$$

- CI = confidence interval
- $\bar{x}$ = sample mean
- z = confidence level value
- s = sample standard deviation
- n = sample size

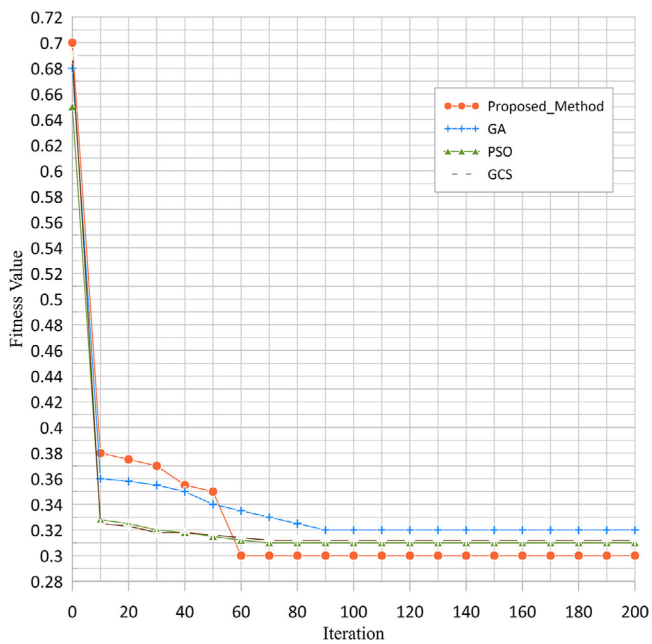

FIGURE 5 Fitness comparisons with 100 tasks

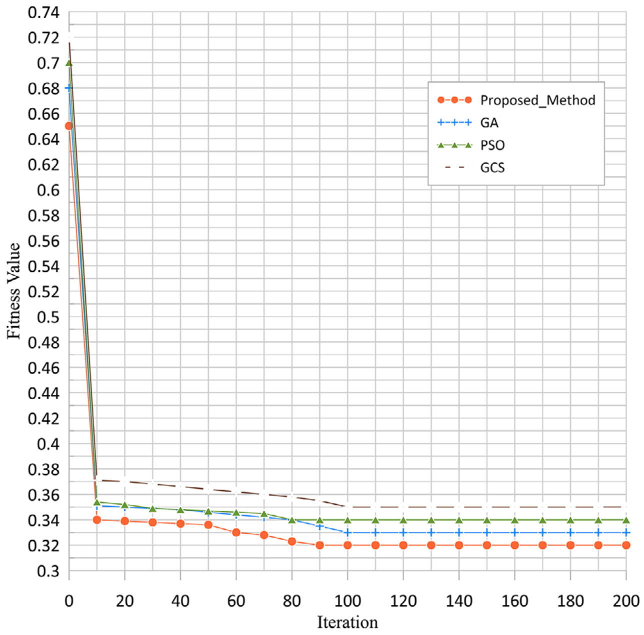

FIGURE 6 Fitness comparisons with 200 tasks

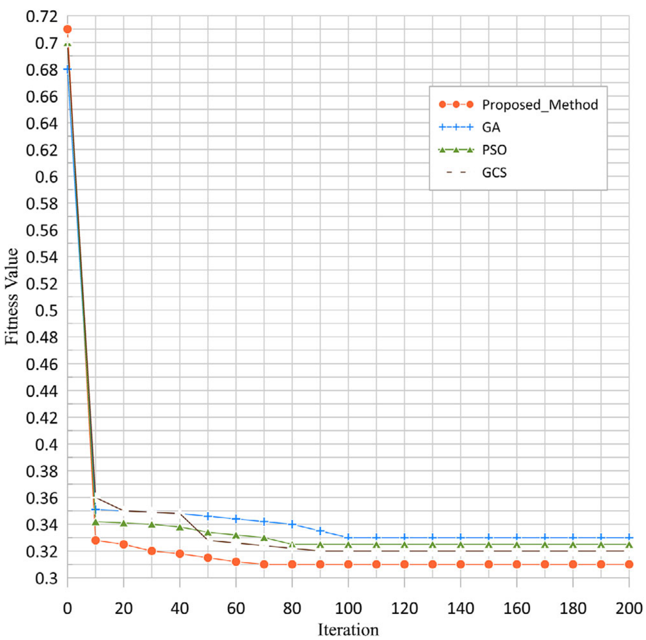

FIGURE 7 Fitness comparisons with 300 tasks

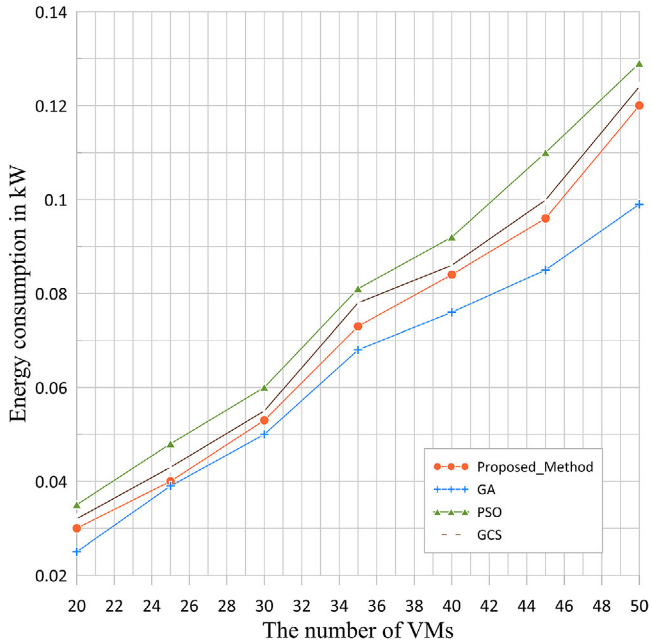

FIGURE 8 Comparison of energy consumption of the first scenario with 10 PMs

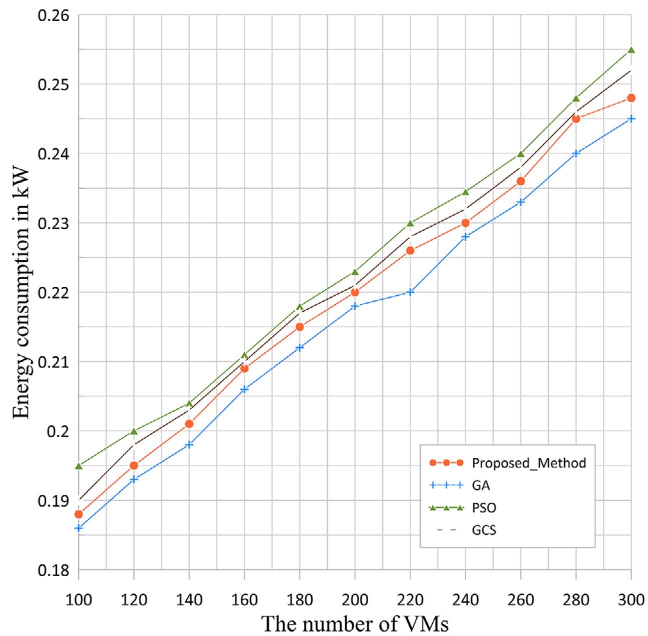

FIGURE 9 Comparison of energy consumption of the second scenario with 50 PMs

Table 4 shows the confidence interval between lower and upper endpoints for an average of ten iterations. These results confirm the minimum variation among the results, even with random input. Also, with the increase in the population, the difference between the lowest and highest point becomes short; therefore, GA is better than other algorithms. Then, the proposed method is better than PSO and GCS.

Also, Figures 8–10 illustrate the amount of energy consumption in each algorithm when PMs are 10, 50, and 100, respectively. The results demonstrate that our proposed method consumes less energy compared to other works.

Also, Figures 11–13 illustrate the amount of time allocation in each algorithm when the number of PMs is 10, 50, and 100, respectively. The results demonstrate that our proposed method consumes less time compared to other works.

As can be seen in Figure 14, the fitness results were obtained for 30 iterations for the numbers between 0.3 and 0.32. Therefore, the best and most frequent answer is chosen as fitness, which is 0.3.

TABLE 4 The confidence interval results for the proposed method and other algorithms

Technique	Performance measuring parameter			
	Cost		Peak Average Ration	
Endpoint	Lower	Upper	Lower	Upper
GA	1955	1962.5	2.3614	2.5952
PSO	1968.1	1980.6	2.4106	2.6532
GCS	1949	1998	2.0365	2.4974
Proposed method	1959.2	1967.8	2.3751	2.6185

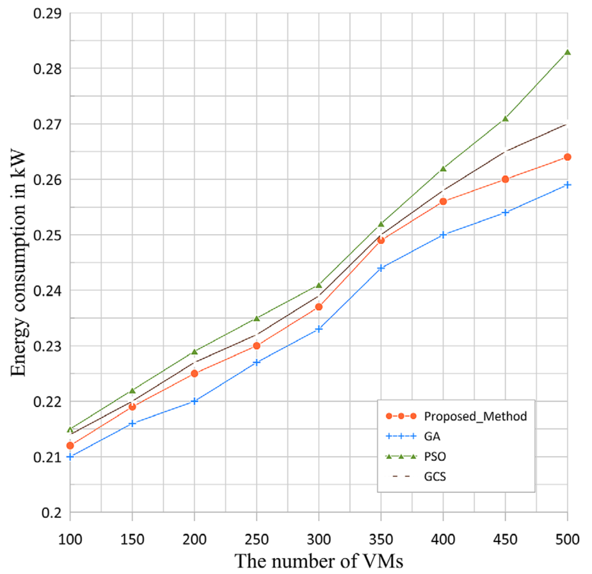

FIGURE 10 Comparison of energy consumption of the third scenario with 100 PMs

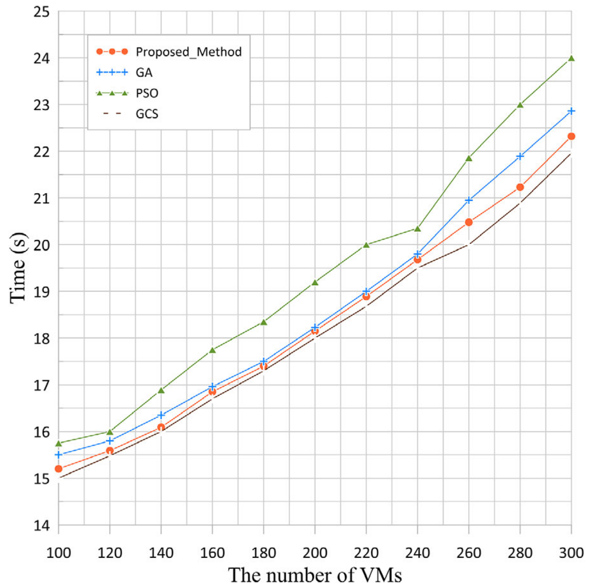

FIGURE 12 Comparison of allocation time of the second scenario with 50 PMs

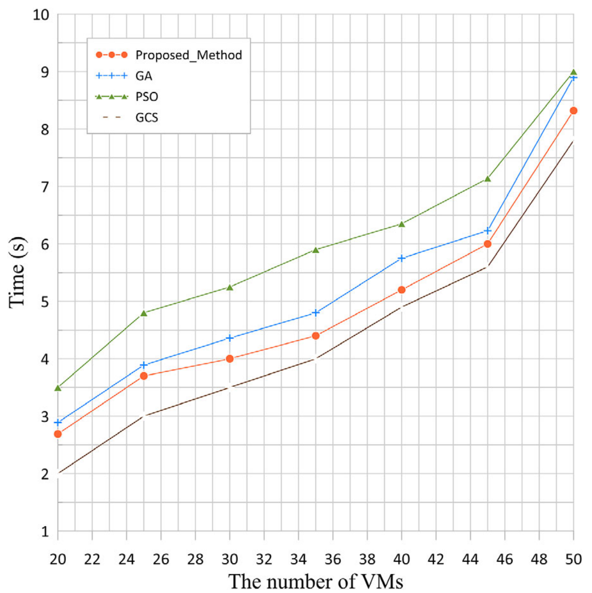

FIGURE 11 Comparison of allocation time of the first scenario with 10 PMs

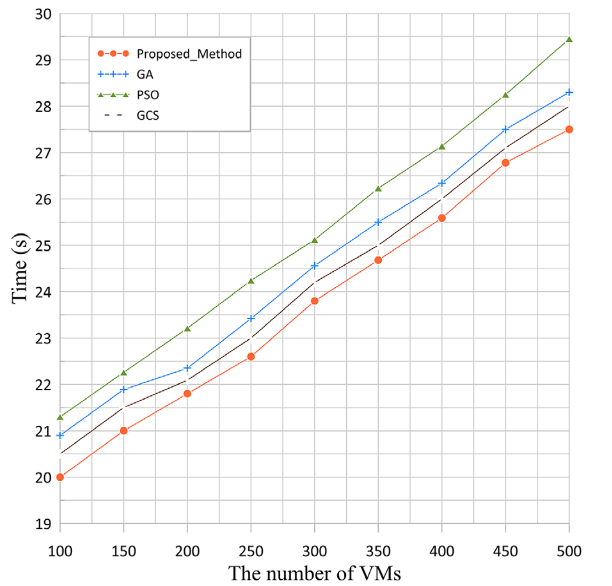

FIGURE 13 Comparison of allocation time of the third scenario with 100 PMs

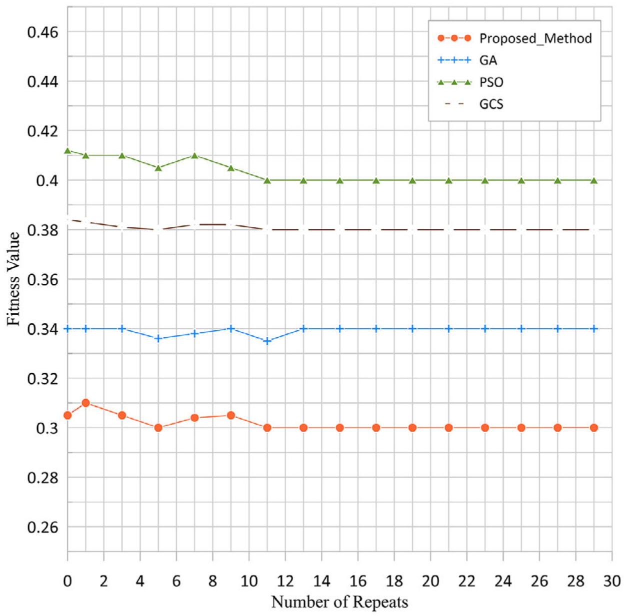

FIGURE 14 Stability of the first scenario

6 | CONCLUSION

Customers can supply resources on-demand on a pay-as-you-go basis using the cloud computing concept, which uses the virtualization of computational resources. Virtualization technology enables cloud suppliers to solve energy inefficiencies by creating several virtual machine instances on a single physical server, allowing them to use resources better and increase their return on investment (ROI). Since VM allocation in cloud environments is an NP-hard problem, many researchers have attempted to offer an efficient solution to this issue by utilizing meta-heuristic algorithms. This paper formulated VM allocation as a multi-objective optimization issue and tackled this problem using the IGWO algorithm. Since VMs allocation is sensitive to initial conditions, we combined VM allocation and IGWO to achieve robust search capability and fast convergence. The suggested algorithm's performance is evaluated and compared to PSO and GAs. The simulation outcomes prove that our algorithm consumes less energy and requires less allocation time. Meanwhile, it offers fast convergence. These heuristic techniques reduce CDC energy usage while using the minimum SLA. Since the static data set is used in this study, future work can also consider the dynamic networks and scenarios [76]. As today's data centre has changed to use containers rather than VMs, so in the future, we will update our method to fit the new container scenarios. In addition, utilizing the provided method in other distributed environments such as differential evolution (DE) [77], Markovian complex networks [78, 79], and fuzzy-based neural networks [80–83] are very interesting lines for future research. Furthermore, using the provided cloud-based framework to compress the video [84, 85] and increase its quality could be investigated in the future. Finally, considering other factors such as fault [86, 87] and risk mitigation [88] to develop secure systems [87] can be done in future research.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

DATA AVAILABILITY STATEMENT

All data are reported in the paper.

ORCID

Danial Javaheri https://orcid.org/0000-0002-7275-2370
Parisa Sabbagh https://orcid.org/0000-0001-5342-6994
Behdad Arandian https://orcid.org/0000-0002-4984-7224
Karlo Abnoosian https://orcid.org/0000-0002-6080-9202

REFERENCES

1. Heidari, A., Navimipour, N.J.: A new SLA-aware method for discovering the cloud services using an improved nature-inspired optimization algorithm. PeerJ Comput. Sci. 7, e539 (2021)
2. Shen, H., Zhang, M., Wang, H., Guo, F., Susilo, W.: A cloud-aided privacy-preserving multi-dimensional data comparison protocol. Inf. Sci. 545, 739–752 (2021)
3. Abdel-Basset, M., Mohamed, M., Chang, V.: NMCDA: A framework for evaluating cloud computing services. Future Gener. Comput. Syst. 86, 12–29 (2018)
4. Jahantigh, M.N., Rahmani, A.M., Navimipour, N.J., Rezaee, A.: Corrigendum: Integration of Internet of Things and cloud computing: a systematic survey. IET Commun. 14(21), 3944–3944 (2020)
5. Kumar, P.R., Raj, P.H., Jelciana, P.: Exploring data security issues and solutions in cloud computing. Procedia Comput. Sci. 125, 691–697 (2018)
6. Ghomi, E.J., Rahmani, A.M., Qader, N.N.: Load-balancing algorithms in cloud computing: A survey. J. Network Comput. Appl. 88, 50–71 (2017)
7. Zanbouri, K., Jafari Navimipour, N.: A cloud service composition method using a trust-based clustering algorithm and honeybee mating optimization algorithm. Int. J. Commun. Syst. 33(5), e4259 (2020)
8. Ding, L., et al.: Definition and application of variable resistance coefficient for wheeled mobile robots on deformable terrain. IEEE Trans. Rob. 36(3), 894–909 (2020)
9. Weng, L., He, Y., Peng, J., Zheng, J., Li, X.: Deep cascading network architecture for robust automatic modulation classification. Neurocomputing 455, 308–324 (2021)
10. Kunwar, V., Agarwal, N., Rana, A., Pandey, J.: Load balancing in cloud—a systematic review. In: Big Data Analytics, Springer, Singapore (2018)
11. Yadav, R., Zhang, W.: MeReg: Managing energy-SLA tradeoff for green mobile cloud computing. Wireless Commun. Mobile Comput. 2017, 1-11 (2017)
12. Yadav, R., Zhang, W., Kaiwartya, O., Singh, P.R., Elgendy, I.A., Tian, Y.-C.: Adaptive energy-aware algorithms for minimizing energy consumption and SLA violation in cloud computing. IEEE Access 6, 55923–55936 (2018)
13. Naseri, A., Navimipour, N.J.: A new agent-based method for QoS-aware cloud service composition using particle swarm optimization algorithm. J. Ambient Intell. Hum. Comput. 10(5), 1851–1864 (2019)
14. Naik, B.B., Singh, D., Samaddar, A.B.: FHCS: Hybridised optimisation for virtual machine migration and task scheduling in cloud data center. IET Commun. 14(12), 1942–1948 (2020)
15. Buyya, R., et al.: A manifesto for future generation cloud computing: Research directions for the next decade. ACM Comput. Surv. (CSUR). 51(5), 1–38 (2018)
16. Ma, F., Liu, F., Liu, Z.: Distributed load balancing allocation of virtual machine in cloud data center. In: 2012 IEEE International Conference on Computer Science and Automation Engineering, Beijing, China, 22–24 June 2012
17. He, Y., Dai, L., Zhang, H.: Multi-branch deep residual learning for clustering and beamforming in user-centric network. IEEE Commun. Lett. 24(10), 2221–2225 (2020)
18. Goudarzi, P., Hosseinpour, M., Ahmadi, M.R.: Joint customer/provider evolutionary multi-objective utility maximization in cloud data center

networks. Iran. J. Sci. Technol., Trans. Electr. Eng. 45(2), 479–492 (2021)
19. Computing, S.I.C., Masdari, M., Nabavi, S.S., Ahmadi, V.: Computing. J. Network Comput. Appl. (2016)
20. Piraghaj, S.F., Calheiros, R.N., Chan, J., Dastjerdi, A.V., Buyya, R.: Virtual machine customization and task mapping architecture for efficient allocation of cloud data center resources. The Comput. J. 59(2), 208–224 (2016)
21. Aggarwal, A., Dimri, P., Agarwal, A., Bhatt, A.: Self adaptive fruit fly algorithm for multiple workflow scheduling in cloud computing environment. Kybernetes (2020)
22. Aslani, R., Rasti, M.: A distributed power control algorithm for energy efficiency maximization in wireless cellular networks. IEEE Wireless Commun. Letters. 9(11), 1975–1979 (2020)
23. Sharma, N.K., Guddeti, R.M.R.: On demand virtual machine allocation and migration at cloud data center using hybrid of cat swarm optimization and genetic algorithm. in 2016 Fifth International Conference on Eco-friendly Computing and Communication Systems (ICECCS), pp. 27–32, IEEE, (2016)
24. Aslani, R., Saberinia, E., Rasti, M.: Resource allocation for cellular V2X networks Mode-3 With underlay approach in LTE-V standard. IEEE Trans. Veh. Technol. 69(8), 8601–8612 (2020)
25. Abdel-Basset, M., Abdle-Fatah, L., Sangaiah, A.K.: An improved Lévy based whale optimization algorithm for bandwidth-efficient virtual machine placement in cloud computing environment. Cluster Comput. 22(4), 8319–8334 (2019)
26. Zaman, S., Grosu, D.: A combinatorial auction-based mechanism for dynamic VM provisioning and allocation in clouds. IEEE Trans. Cloud Comput. 1(2), 129–141 (2013)
27. Donyagard Vahed, N., Ghobaei-Arani, M., Souri, A.: Multiobjective virtual machine placement mechanisms using nature-inspired metaheuristic algorithms in cloud environments: A comprehensive review. Int. J. Commun. Syst. 32(14), e4068 (2019)
28. Chen, H., Heidari, A.A., Chen, H., Wang, M., Pan, Z., Gandomi, A.H.: Multi-population differential evolution-assisted Harris hawks optimization: Framework and case studies. Future Gener. Comput. Syst. 111, 175–198 (2020)
29. Zhang, Y., Liu, R., Wang, X., Chen, H., Li, C.: Boosted binary Harris hawks optimizer and feature selection. Eng. Comput. 1–30 (2020)
30. Shan, W., Qiao, Z., Heidari, A.A., Chen, H., Turabieh, H., Teng, Y.: Double adaptive weights for stabilization of moth flame optimizer: balance analysis, engineering cases, and medical diagnosis. Knowledge-Based Syst. 214, 106728 (2021)
31. Wang, M., et al.: Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses. Neurocomputing 267, 69–84 (2017)
32. Xu, Y., Chen, H., Luo, J., Zhang, Q., Jiao, S., Zhang, X.: Enhanced Moth-flame optimizer with mutation strategy for global optimization. Inf. Sci. 492, 181–203 (2019)
33. Shen, L., et al.: Evolving support vector machines using fruit fly optimization for medical data classification. Knowledge-Based Syst. 96, 61–75 (2016)
34. Yu, H., et al.: Dynamic Gaussian bare-bones fruit fly optimizers with abandonment mechanism: method and analysis. Eng. Comput. 1–29 (2020)
35. Tu, J., et al.: Evolutionary biogeography-based whale optimization methods with communication structure: towards measuring the balance. Knowledge-Based Syst. 212, 106642 (2021)
36. Wang, M., Chen, H.: Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosis. Appl. Soft Comput. 88, 105946 (2020)
37. Xu, X., Chen, H.-l.: Adaptive computational chemotaxis based on field in bacterial foraging optimization. Soft Comput. 18(4), 797–807 (2014)
38. Yu, C. et al.: SGOA: annealing-behaved grasshopper optimizer for global tasks. Eng. Comput. 1–28 (2021)
39. Zhong, S.-Q., Zhao, S.-C., Zhu, S.-N.: Photovoltaic properties enhanced by the tunneling effect in a coupled quantum dot photocell. Results Phys. 24, 104094 (2021)
40. Zhao, X., Li, D., Yang, B., Ma, C., Zhu, Y., Chen, H.: Feature selection based on improved ant colony optimization for online detection of foreign fiber in cotton. Appl. Soft Comput. 24, 585–596 (2014)
41. Heidari, A., Navimipour, J.N.: A new SLA-aware method for discovering the cloud services using an improved nature-inspired optimization algorithm. PeerJ Comput. Sci. 7, e539, (2021)
42. Satpathy, A., Addya, S.K., Turuk, A.K., Majhi, B., Sahoo, G.: Crow search based virtual machine placement strategy in cloud data centers with live migration. Comput. Electr. Eng. 69, 334–350 (2018)
43. Kashiwagi, T., Kourai, K.: Flexible and Efficient Partial Migration of Split-memory VMs. In: 2020 IEEE 13th International Conference on Cloud Computing (CLOUD), pp. 248–257, IEEE, (2020)
44. Pişirir, E., Uçar, E., Chouseinoglou, O., Sevgi, C.: Structural equation modeling in cloud computing studies: a systematic literature review. Kybernetes 49, 982–1019, (2019)
45. Yadav, R., Zhang, W., Chen, H., Guo, T.: Mums: Energy-aware vm selection scheme for cloud data center. In: 2017 28th International Workshop on Database and Expert Systems Applications (DEXA), Lyon, France, 28–31 August 2017
46. Addya, S.K., Satpathy, A., Ghosh, B.C., Chakraborty, S., Ghosh, S.K.: Power and time aware vm migration for multi-tier applications over geo-distributed clouds. In: 2019 IEEE 12th International Conference on Cloud Computing (CLOUD), Milan, Italy, 8–13 July 2019
47. Addya, S.K., Turuk, A.K., Satpathy, A., Sahoo, B., Sarkar, M.: A strategy for live migration of virtual machines in a cloud federation. IEEE Syst. J. 13(3), 2877–2887 (2018)
48. Yadav, R., Zhang, W., Li, K., Liu, C., Laghari, A.A.: Managing overloaded hosts for energy-efficiency in cloud data centers. Cluster Comput. 1–15 (2021)
49. Yadav, R., Zhang, W., Li, K., Liu, C., Shafiq, M., Karn, N.K.: An adaptive heuristic for managing energy consumption and overloaded hosts in a cloud data center. Wirel. Networks 26(3), 1905–1919 (2020)
50. Satpathy, A., Sahoo, M.N., Chottray, L., Majhi, B., Mishra, A., Bakshi, S.: Vrmap: A cost and time aware remapping of virtual data centres over a geo-distributed infrastructure. In: 2020 International Conference on COMmunication Systems & NETworkS (COMSNETS), pp. 427–434, IEEE, (2020)
51. Naik, B.B., Singh, D., Samaddar, A.B.: Multi-objective Virtual Machine Selection in Cloud Data Centers Using Optimized Scheduling. Wirel. Pers. Commun. 1–24 (2020)
52. Wei, W., Gu, H., Lu, W., Zhou, T., Liu, X.: Energy efficient virtual machine placement with an improved ant colony optimization over data center networks. IEEE Access 7, 60617–60625 (2019)
53. López-Pires, F., Barán, B.: Many-objective optimization for virtual machine placement in cloud computing. In: Research Advances in Cloud Computing. pp. 291–326, Springer, Singapore (2017)
54. Satpathy, A., Addya, S.K., Turuk, A.K., Majhi, B., Sahoo, G.: A resource aware VM placement strategy in cloud data centers based on crow search algorithm. In: 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 1–6, IEEE, (2017)
55. Ye, X., Yin, Y., Lan, L.: Energy-efficient many-objective virtual machine placement optimization in a cloud computing environment. IEEE Access 5, 16006–16020 (2017)
56. Addya, S.K., Turuk, A.K., Sahoo, B., Sarkar, M., Biswash, S.K.: Simulated annealing based VM placement strategy to maximize the profit for Cloud Service Providers. Eng. Sci. Technol. Int. J. 20(4), 1249–1259 (2017)
57. Sharma, N.K., Reddy, G.R.M.: Novel energy efficient virtual machine allocation at data center using genetic algorithm. In: 2015 3rd International Conference on Signal Processing, Communication and Networking (ICSCN), pp. 1–6, IEEE, (2015)
58. Ali, H.M., Lee, D.C.: A biogeography-based optimization algorithm for energy efficient virtual machine placement. In: 2014 IEEE Symposium on Swarm Intelligence, Orlando, FL, USA, 9–12 December 2014
59. Jamali, S., Malektaji, S.: Improving grouping genetic algorithm for virtual machine placement in cloud data centers. In: 2014 4th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, 29–30 October 2014

60. Li, C., et al.: Developing a new intelligent system for the diagnosis of tuberculous pleural effusion. Comput. Methods Programs Biomed. 153, 211–225 (2018)
61. Roshani, M., et al.: Proposing a gamma radiation based intelligent system for simultaneous analyzing and detecting type and amount of petroleum by-products. Nucl.Eng. Technol. (2020)
62. Sattari, M.A., Roshani, G.H., Hanus, R., Nazemi, E.: Applicability of time-domain feature extraction methods and artificial intelligence in two-phase flow meters based on gamma-ray absorption technique. Measurement 168, 108474 (2021)
63. Bie, Y., Ji, J., Wang, X., Qu, X.: Optimization of electric bus scheduling considering stochastic volatilities in trip travel time and energy consumption. Comput.-Aided Civ. Infrastruct. Eng. (2021)
64. Hu, J., et al.: Orthogonal learning covariance matrix for defects of grey wolf optimizer: insights, balance, diversity, and feature selection. Knowledge-Based Syst. 213, 106684 (2021)
65. Zhao, X., et al.: Chaos enhanced grey wolf optimization wrapped ELM for diagnosis of paraquat-poisoned patients. Comput. Biol. Chem. 78, 481–490 (2019)
66. Sreenu, K., Malempati, S.: MFGMTS: Epsilon constraint-based modified fractional grey wolf optimizer for multi-objective task scheduling in cloud computing. IETE J. Res. 65(2), 201–215 (2019)
67. Hosseini-Hemati, S., Sheisi, G.H., Karimi, S.: Allocation-based optimal reactive power dispatch considering polynomial load model using improved grey wolf optimizer. Iran. J. Sci. Technol. Trans. Electr. Eng. 45, 921–944 (2021)
68. Mirjalili, S., Mirjalili, S.M., Lewis, A.: Grey wolf optimizer. Adv. Eng. Software. 69, 46–61 (2014)
69. Goyal, S., Yadav, N., Rani, A., Singh, V.: Classification of SGS-SRAD denoised MRI using GWO optimized SVM. IETE J. Res. 1–11 (2020)
70. Chen, J., Huang, S., Shahabi, L.: Economic and environmental operation of power systems including combined cooling, heating, power and energy storage resources using developed multi-objective grey wolf algorithm. Appl. Energy. 298, 117257 (2021)
71. Li, B., Xiao, G., Lu, R., Deng, R., Bao, H.: On feasibility and limitations of detecting false data injection attacks on power grid state estimation using D-FACTS devices. IEEE Trans. Ind. Inf. 16(2), 854–864 (2019)
72. Mohammadzadeh, A., Masdari, M., Gharehchopogh, F.S., Jafarian, A.: Improved chaotic binary grey wolf optimization algorithm for workflow scheduling in green cloud computing. Evol. Intell. 1–29 (2020)
73. Wang, T., Liu, W., Zhao, J., Guo, X., Terzija, V.: A rough set-based bio-inspired fault diagnosis method for electrical substations. Int. J. Electr. Power Energy Syst. 119, 105961 (2020)
74. Aruna, P., Vasantha, S.: A particle swarm optimization algorithm for power-aware virtual machine allocation. In: 2015 6th International Conference on Computing, Communication and Networking Technologies (ICCCNT), NW, Washington, DC, USA, 13–15 July 2015
75. Selvaraj, A., Patan, R., Gandomi, A.H., Deverajan, G.G., Pushparaj, M.: Optimal virtual machine selection for anomaly detection using a swarm intelligence approach. Appl. Soft Comput. 84, 105686 (2019)
76. Zhao, C., Zhong, S., Zhang, X., Zhong, Q., Shi, K.: Novel results on nonfragile sampled-data exponential synchronization for delayed complex dynamical networks. Int. J. Robust Nonlinear Control 30(10), 4022–4042 (2020)
77. Sun, G., Li, C., Deng, L.: An adaptive regeneration framework based on search space adjustment for differential evolution. Neural Comput. Appl. 33, 9503–9519 (2021)
78. Zhao, C., Zhong, S., Zhong, Q., Shi, K.: Synchronization of Markovian complex networks with input mode delay and Markovian directed communication via distributed dynamic event-triggered control. Nonlinear Anal. Hybrid Syst. 36, 100883 (2020)
79. Xie, W., Zhang, R., Zeng, D., Shi, K., Zhong, S.: Strictly dissipative stabilization of multiple-memory Markov jump systems with general transition rates: A novel event-triggered control strategy. Int. J. Robust Nonlinear Control. 30(5), 1956–1978 (2020)
80. Wang, T., et al.: A weighted corrective fuzzy reasoning spiking neural P system for fault diagnosis in power systems with variable topologies. Eng. Appl. Artif. Intell. 92, 103680 (2020)
81. Luo, J., Li, M., Liu, X., Tian, W., Zhong, S., Shi, K.: Stabilization analysis for fuzzy systems with a switched sampled-data control. J. Franklin Inst. 357(1), 39–58 (2020)
82. Roshani, M., et al.: Combination of X-ray tube and GMDH neural network as a nondestructive and potential technique for measuring characteristics of gas-oil–water three phase flows. Measurement 168, 108427 (2021)
83. Roshani, M., et al.: Application of GMDH neural network technique to improve measuring precision of a simplified photon attenuation based two-phase flowmeter. J Flow Meas. Instrum. 75, 101804 (2020)
84. Xu, M., Li, C., Chen, Z., Wang, Z., Guan, Z.: Assessing visual quality of omnidirectional videos. IEEE Trans. Circuits Syst. Video Technol. 29(12), 3516–3530 (2018)
85. Yang, R., Xu, M., Liu, T., Wang, Z., Guan, Z.: Enhancing quality for HEVC compressed videos. IEEE Trans. Circuits Syst. Video Technol. 29(7), 2039–2054 (2018)
86. Chen, X., Wang, T., Ying, R., Cao, Z.: A fault diagnosis method considering meteorological factors for transmission networks based on P systems. Entropy 23(8), 1008, (2021)
87. Huang, Z., Wang, T., Liu, W., Valencia-Cabrera, L., Pérez-Jiménez, M.J., Li, P.: A fault analysis method for three-phase induction motors based on spiking neural P systems. Complexity 2021, 1–19 (2021)
88. Hu, B., Wu, Y., Wang, H., Tang, Y., Wang, C.: Risk mitigation for rockfall hazards in steeply dipping coal seam: a case study in Xinjiang, northwestern China. Geomatics Nat. Hazards Risk 12(1), 988–1014 (2021)

How to cite this article: Hashemi, M., Javaheri, D., Sabbagh, P., Arandian, B., Abnoosian, K.: A multi-objective method for virtual machines allocation in cloud data centres using an improved grey wolf optimization algorithm. IET Commun. 15, 2342–2353 (2021). https://doi.org/10.1049/cmu2.12274