Internet of Things and Cyber-Physical Systems 3 (2023) 28–36

Contents lists available at ScienceDirect

Internet of Things and Cyber-Physical Systems

journal homepage: www.keaipublishing.com/en/journals/internet-of-things-and-cyber-physical-systems

Seagull optimization algorithm based multi-objective VM placement in edge-cloud data centers

Sayyidshahab Nabavi [a], Linfeng Wen [b], Sukhpal Singh Gill [c,*], Minxian Xu [b]

[a] Institute of Electronic Engineering, FH JOANNEUM, University of Applied Sciences, Graz, Austria
[b] Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
[c] School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK

ARTICLE INFO

Keywords:
Cloud computing
Seagull optimization
VM placement
Power consumption
Network traffic
Edge cloud data centers

ABSTRACT

Edge-Cloud Datacenters (ECDCs) have been massively exploited by the owners of technology and industrial centers to satisfy the user demand. At the same time, the amount of energy used by these data centers is considerable. To address this challenge, Virtual Machine (VM) placement of the ECDCs plays an important role; therefore, assigning VM properly to physical machines (PM) can significantly decrease the amount of energy consumption. The applied assigning technique simultaneously must consider additional objectives involving traffic and power usage of the network elements, which makes it a challenging problem. This paper proposes a multi-objective VM placement approach in edge-cloud data centers, which uses Seagull optimization to optimize power and network traffic together. In this strategy, the network traffic among PMs is reduced by concentrating the communications of VMs on the same PMs to reduce the amount of transferred data through the network and reduce the PMs' power consumption by consolidating VMs to fewer PMs, which consumes less energy. We evaluate with simulations in CloudSim and test two different network topologies, VL2 (Virtual Layer 2) and three-tier, to validate that the proposed approach can effectively reduce traffic and power consumption in ECDCs. The experimental results show that our proposed method can decrease energy consumption by 5.5% while simultaneously reducing network traffic by 70% and the power consumption of the network components by 80%.

1. Introduction

Nowadays, services of Cloud computing have become prevalent and adapted widely across various industries and fulfill numerous Internet services [1]. Edge-computing technology [2] seems to be an appealing alternative, particularly for hosting compute workloads as near as possible to the data sources and end-users [3,4]. As indicated in Fig. 1, the edge can be considered as the improved version of cloud computing. These improvements in comparison with cloud computing are as follows [3].

- *Relieving pressure of backbone network.* Since the nodes of edge computing are able to conduct huge computations without main data center assistance, The transferred data on the backbone network reduces dramatically.
- *Improved quality of service (QoS).* As the amount of transferred data is less than cloud computing. Therefore, the QoS would be more efficient, and the service time is increased.
- *Collaborative assistance.* Robust cloud backup can give a high level of processing power and large storage when the edge is unable to afford it.

In 2017, data centers used 416.2 billion kilowatt-hours of power, or about 2% of the world's total electricity usage [5]. As a result of the rapid increase in the data centers power consumption without losing the quality of service of the data centers became a hot research topic [6,7]. The same essential technology behind cloud computing and edge computing concept is based on resource virtualization, which abstracts hardware resources from software to support numerous task on the same PMs [8,9]. Edge computing, on the other hand, differs significantly from cloud computing in a number of key respects, including mobility-assisted services' knowledge of their physical location, devices' being resource-constrained and diverse, and the devices' being widely dispersed [10,11]. Heavyweight virtualization (directly running a virtual machine (VM) monitor may not be suitable for all use cases due to these differences [12,13]. Examples of Internet of Things (IoT) applications that might benefit from edge computing [14]. This restriction necessitates the use of lightweight virtualization [15].

* Corresponding author. School of Electronic Engineering and Computer Science, Queen Mary University of London, London, E1 4NS, UK.
E-mail addresses: shahab.nab@gmail.com (S. Nabavi), lf.wen@siat.ac.cn (L. Wen), s.s.gill@qmul.ac.uk (S.S. Gill), mx.xu@siat.ac.cn (M. Xu).

https://doi.org/10.1016/j.iotcps.2023.01.002
Received 25 December 2022; Received in revised form 31 December 2022; Accepted 15 January 2023
Available online 24 January 2023

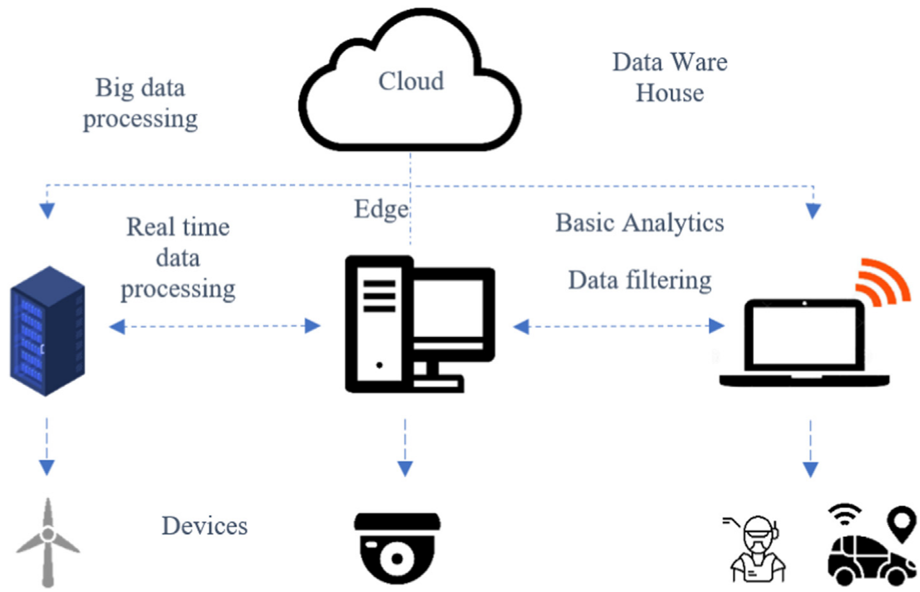

Fig. 1. Paradigm of Edge-Cloud Computing.

Two major sources of energy consumption come from the servers and network communications, which can be directed influenced by location of placed VMs on the hosts [16]. In this work, we proposed an algorithm to optimize the placement of VMs to reduce the energy consumption and network traffics. We modeled the scheduling problem as a multi-objective optimization problem and use seagull based approach to solve it. The results have validated that our proposed approach can outperform the baselines in energy consumption and network traffics significantly under different network topologies.

The rest of the article is as follows. Section 2 overviews and compares the related work in network-aware and energy-aware VM placement. Section 3 presents the Seagull optimization based algorithm for VM placement reduce network traffics and energy consumption. In sections 4 and 5, the VM placement method and evaluation of proposed scheme are described respectively. Section 6 concludes the paper and highlights the future directions.

2. Related works

In this section, we present a summary of the previous investigations in the context of VM placement in Edge cloud data centers. The virtual machine placement in cloud data centers is considered to be NP-Hard [17], so numerous approaches and methods have been investigated to address this problem. These plans in this section are categorized based on their aim as follow.

2.1. Network-aware virtual machine placement

Darrous et al. [18] developed a VM management system capable of reducing the time required to assign VMs over a heterogeneous network. The proposed method follows two complementary strategies to achieve this goal. Firstly, it enjoys the deduplication of similar data to reduce communication traffic. Secondly, the strategy forwards data communication to the links with high bandwidth, which minimizes the provisioning time. The suggested method compared with existing methods, namely Interplanetary File System (IPFS), BitTorrent, and OpenStack Swift, and the results indicate that the performance of their approach is considerably better than the overmentioned methods. Abdel-Basset et al. [19] developed a virtual machine placement issue defined as a bin packing problem with the goal of reducing traffic communication. Moreover, they developed a Whale optimization algorithm (WOA) using the Lévy flight technique in Cloudsim simulator [20]. The proposed approach is compared with the first fit, Particle Swarm Optimization (PSO), best fit, intelligent tuned harmony search, genetic algorithm, and using 25 random datasets. Results demonstrate that in large-scale problems, the proposed method performs more efficiently. However, this line of research only focuses on the optimization of network communications.

2.2. Power-aware virtual machine placement

Abohamama et al. [21] developed a hybrid Virtual Machine Placement (VMP) method based on evolutionary algorithms and optimized resource utilization by using a best-fit allocation technique; moreover, they lowered data center energy usage by reducing the number of enabled servers. Additionally, this research demonstrates that the suggested VMP algorithm achieves a balanced use of active servers' available resources. Ghobaei-Arani et al. [22] presented a method considering electrical consumption also Service Level Agreement Violation (SLAV). This approach enjoys the Bidirectional Forwarding Detection (BFD) algorithm, which applies automata correlation, theory, prediction algorithms, coefficient and predicts the incoming load VMs of the next future to assign VMs to PMs effectively to reduce service level agreement violation (SLAV) and energy consumption. Alharbi et al. [23] proposed a dynamic VM placement in cloud data centers over a period of time, using a constrained combinatorial optimization model. In this model, they developed the Ant Colony Optimization algorithm and used the information of PMs and VMs characteristics to minimize the energy consumption of PMs. Compared with two other Ant Colony System (ACS) approaches and the First Fit Decreasing (FFD) algorithm, their model functions more efficiently in small, medium, and large test scales. Shaw et al. [24] presented a novel strategy known as predictive anti-correlated VM placement algorithm (PACPA) that can improve power efficiency and enhance the capability to reach considerable improvements in the quality of the provided services. They also compared ML methodologies, and after comparison their accuracy, and based on their performance, Artificial Neural Networks (ANN) methodology is chosen to predict the demanding resources, including CPU and RAM; consequently, it improves scheduling time of migrations and overall placement strategy and aims to consolidate VMs to the same PM. However, this category of work does not consider the energy consumption due to varied traffics.

2.3. Multi-objective virtual machine placement

To reduce the load on the data center's physical equipment and network infrastructure, earlier we proposed [25] an adaptation of the Artificial Bee

Colony (ABC) algorithm that takes into account the interactions between virtual machines. The suggested strategy for VM placement is evaluated by simulating the VL2 (Virtual Layer 2) and three-tier network architecture in the CloudSim toolkit and demonstrating the resulting decrease in network traffic and power consumption by the Edge-Cloud Datacenter (ECDC). Adyson et al. [26] explored the multi-objective optimization problem of deploying IoT applications and balancing the load on an edge computing system. The suggested genetic algorithm takes inspiration from the NSGAII and BRKGA[1] in order to provide solutions that are both realistic and close to the Pareto optimal solution. They want to improve the model so that it can take into account application migration, user mobility, and other dynamic changes at the network level, and the formulation is intended to consider a wide range of application aspects such as the response time demand for time-critical applications.

2.4. Critical analysis

Table 1 shows the summary of related works. Our approach advances the relevant area by considering the traffic and energy together and utilizes seagull optimization algorithm (SOA) to find the optimized decisions for VM placement. SOA gives better performance compared to other popular meta-heuristic optimization approaches because SOA is very effective in solving large-scale constrained problems such as resource scheduling in Cloud/Edge Computing [27]. Further, SOA has less computational complexity and it achieves global minima very quickly because of its better exploration and exploitation ability [28].

Table 1
Comparison of related works.

Study	Traffic	Energy	MOA	Algorithm Type	EE
[19]	✓			WOA	Cloudsim
[5]		✓		PABF algorithm	Cloudsim
[6]		✓		Ant Colony System (ACS)	Cloudsim
[24]		✓		PACPA	Cloudsim
[25]			✓	ABC algorithm	Cloudsim
Proposed (this work)	✓	✓	✓	Seagull optimization algorithm	CloudSim

Abbreviations- MOA: Multi-objective Optimization Approach, EE: Experimental Environment.

3. Seagull VM placement optimization algorithm

Seagulls, also scientifically known as Laridae, are spread worldwide. There are several seagull species, each with its own distinct qualities [29]. Seagulls, being sociable birds, naturally take a strategic approach to finding and attacking their haunt. The seagulls' migratory and assault strategy is the most fascinating aspect of their activity (Fig. 2). Seasonal migration is defined as an organism's movement from one habitat to another in quest of the best available food supply.

– At the beginning of their migration journey, seagulls obtain different positions to avoid collisions.
– During the Migration journey, seagulls move toward the route of the fittest seagull.

The list of variables used in this research work is described in Table 2. The following mathematical models of migration and hunting are discussed as follows: during the migration procedure, the algorithm imitates the behavior of the seagull's relocation. In this step, three conditions should be considered: Avoiding collisions: To avoid collisions (Fig. 3), parameter A is defined to justify the new position of the seagull.

Table 2
Variable description.

Variable	Definition
α	Applied Coefficient for F_1
β	Applied Coefficient for F_2
π	Applied Coefficient for F_3
P_i	Power usage of i^{th} PM
n	Size of PMs
$Cost_{ij}$	The cost of data transmission between PM_i and PM_j
Ps_i	energy consumption of i-th switch
Bs_i	binary decision element
l_j	energy expenditure of i-th link
Bl_j	binary decision element If $Bl_j = 0$, link is inactive If $Bl_j = 1$, link is active
ns	Size of switches
nl	Size of links
F	The cost function for each Seagull

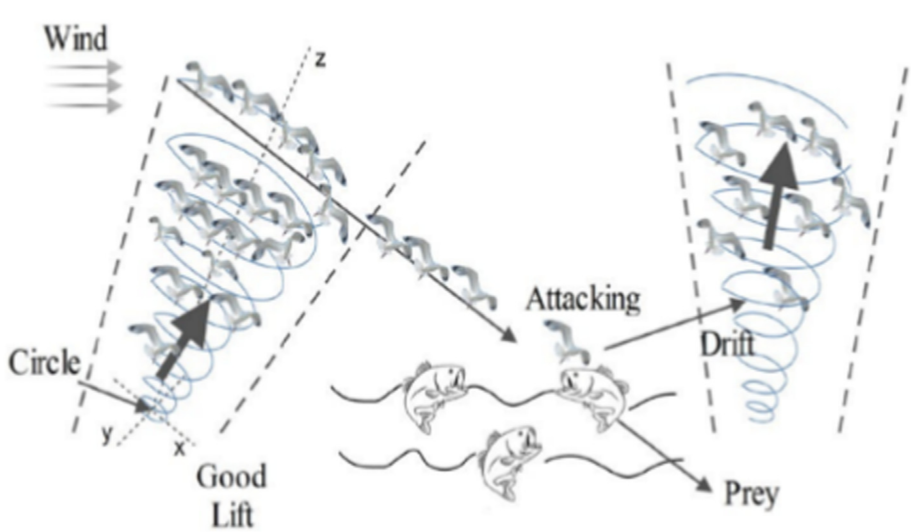

Fig. 2. Migration and attack models of the seagull.

[1] NSGA-II is Nondominated Sorting Genetic Algorithm II and BRKGA is Biased Random Key Genetic Algorithm.

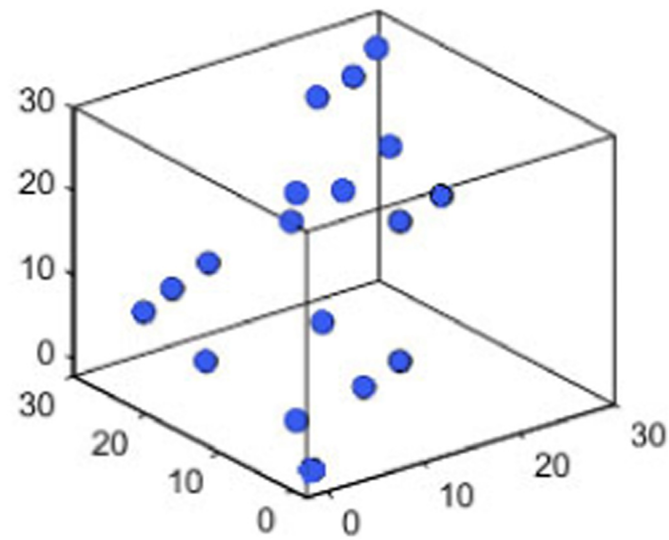

Fig. 3. Avoiding collision among seagulls.

$$\overrightarrow{C_s} = A \times \overrightarrow{p_s}(x), \tag{1}$$

where $\overrightarrow{C_s}$ shows the next search agent position without collision with other seagulls. $\overrightarrow{p_s}(x)$ is the recent position of the search agent. In this expression, x represents the current index of the iteration, and A represents the search agent's mobility pattern within the search region, which is defined as:

$$\begin{aligned} A &= f_C - (x \times (f_C/\mathrm{Max}_{\mathrm{iteration}})) \\ x &= 0, 1, 2, \ldots, \mathrm{Max}_{\mathrm{iteration}} \end{aligned}, \tag{2}$$

where f_C is an element which justifies the frequency of applied element A, which gradually decreased f_C to 0.

Moving Toward best neighbor's path: In order to find the healthiest neighbor, search agents must first avoid collisions:

$$\overrightarrow{M_s} = B \times (\overrightarrow{p_{bs}}(x) - \overrightarrow{p_s}(x)), \tag{3}$$

where $\overrightarrow{M_s}$ is the contemporary position of search agent $\overrightarrow{p_s}$, which moves toward the fittest search agent. B is a random element that keeps the equilibrium of exploitation and exploration. B is computed as follow:

$$B = 2 \times A^2 \times rd, \tag{4}$$

where rd is a random element in the range of [0, 1]. Eventually, as shown in Fig. 4, the search agent is able to revise its assessment of its own relative ranking among the best search agents.

$$\overrightarrow{D_s} = \left| \overrightarrow{C_s} + \overrightarrow{M_s} \right| \tag{5}$$

D_{ij}	VM_0	VM_1	VM_2	VM_3	VM_4
VM_0	0	D_{01}	D_{02}	D_{03}	D_{04}
VM_1	0	0	D_{12}	D_{13}	D_{14}
VM_2	0	0	0	D_{23}	D_{24}
VM_3	0	0	0	0	D_{34}
VM_4	0	0	0	0	0

Fig. 4. Required VM-to-VM data transfer.

where $\overrightarrow{D_s}$ indicates that region between the best-fitting search agent and the other agents.

3.1. Attack (Exploitation)

In the exploitation phase, the background and experience of the search agents are exploited. Seagulls are able to alter the angle of attack and speed constantly. They preserve their altitude by employing their wings and weight. As indicated in Fig. 2, seagulls attack the prey through a spiral movement in the air. The three dimensions of this behavior are calculated by applying the following equations:

$$\begin{aligned} x' &= r \times \cos(k), \\ y' &= r \times \cos(k), \\ z' &= r \times \cos(k), \\ r &= u \times e^{kv}, \end{aligned} \tag{6}$$

in which r equals the radius of spiral turn, k is a random element lay in $[0 < k < 2\pi]$, u and v are consonant elements to characterize the spiral form, also e is the natural base logarithm. The following equation computes the search agent's updated location:

$$\overrightarrow{P_s}(x) = \left(\overrightarrow{D_s} \times x' \times y' \times z' \right) + \overrightarrow{P_{bs}}(x). \tag{7}$$

In this equation, $\overrightarrow{P_s}(x)$ retains the best available option and keeps track of where other search bots currently stand.

4. Proposed VM placement method

Here, we devised a multi-objective VM placement approach with the goals of reducing the number of active PMs, connections, and switches to cut network traffic and power consumption. On the other hand, it tries to aggregate interacting VMs on the same PMs aim to reduce the amount of transferred data. Also, resources limitations such as CPU, storage, and the RAM of the PMs are considered in this plan. To achieve mentioned objectives, the cost function should contain three components, PM and network elements power consumption, network traffic data as proposed in Equation (8):

$$\mathrm{F} = \alpha \mathrm{F}_1 + \beta \mathrm{F}_2 + \pi \mathrm{F}_3. \tag{8}$$

In this equation, alpha (α), beta (β) and gamma (π) are coefficients that are used to keep the equilibrium among objectives. Also, F_1 equals the amount of power consumption of PMs. F_2 is the value of Network elements power consumption, and F_3 stands for the network traffic of the data center.

4.1. Power consumption

To determine the overall energy consumption of the physical machine, our VM Placement strategy uses Equation (9):

$$F_1 = \sum_{i=1}^{n} p_i. \tag{9}$$

In Equation (9), F_1 specified the sum of energy usage of the PMs, p_i equals the electricity usage of the ith physical machine, n equals the number of available physical machines in the datacenter. In Ref. [30], Gao et al. proposed an equation to calculate the amount of power consumption of PMs. Here, we see that the PMs' electric energy consumption and CPU use are directly proportional to one another. Power consumption is represented by Equation (10):

$$P_i = \begin{cases} \left(P_i^{\max} - P_i^{idle}\right) \times U_i^c + P_i^{\mathrm{idle}} & U_c^i > 0 \\ 0 & \text{otherwise} \end{cases}. \tag{10}$$

In this calculation, $P_i^{\max}$ denotes the highest possible power usage of the *ith* physical machine when it is fully used, P_i^{idle} denotes the *ith* physical machine's idle power usage, and U_i^c is the *ith* physical machine's CPU usage. Notably, the CPU frequency in sleep mode is around half that of the highest possible power usage. However, by hosting certain VMs on a physical computer, the physical system's CPU rate is increased in response to the VMs' CPU requirements. Equation (11) is used to determine the usage of each physical machine:

$$U_i^c = \frac{\sum_{j=0}^{m}(X_j \times R_j)}{A_i}. \tag{11}$$

In this equation, m denotes the count of virtual machines installed on the PMi, A_i denotes the exist CPU on the PMi, $R_j$is the CPU requested by the VMj, and X_j denotes a binary value generated [1] using Equation (12):

$$X_j = \begin{cases} 1 & \text{if VMj is assigned to the PMi} \\ 0 & \text{if VMj is not assigned to the PMi} \end{cases}. \tag{12}$$

4.2. Network data transmission

Due to the fact that virtual machines may perform interactive programs such as workflows, they constantly need to communicate with one another. As seen in Fig. 4, an upper triangular matrix is used to specify the quantity of data sent between VMs. D_{ij} denotes the quantity of data that should be exchanged among the VM_i and VM_j in this matrix. The sum of all the entries in the matrix in Fig. 5 reflects the total quantity of data that must be transmitted across all VMs. Additionally, another higher triangular matrix is constructed in this system to represent the connection capacity between distinct physical machines. Data transfer across VMs is restricted by the available bandwidth among physical machines, dictated by the network architecture and switches utilized.

The available bandwidth between the physical equipment is shown in Fig. 5. In Fig. 5: C_{it} is the connection capacity between the *ith* physical machine (PM_i) and the *jth* physical machine (PM_j) in megabits per second, as shown in this matrix. After virtual machines are installed on PMs, PMs interchange data among each other. Fig. 6 shows that the amount of information that must be sent between the PMs and the VMs is represented by the matrix *X*. This matrix clearly demonstrates that X_{ij} represents the quantity of data that will be exchanged among the *ith* physical machine (PM_i) and the *jth* physical machine (PM_j), expressed in megabits per second. Equation (13) is applied to compute the overall network cost when a solution for VM placement has been found:

$$F_2 = \sum_{i=0}^{n-1} \sum_{j=i+1}^{n-1} \mathrm{Cost}_{ij}, \tag{13}$$

where F_2 is the total network traffic cost, and n is the number of physical computers in the data center, $Cost_{ij}$ denotes the cost of data transmission among thePM_i and the PM_j, which may be calculated using the following equation:

C ij	PM0	PM1	PM2	PM3	PM4
PM0	0	C01	C02	C03	C04
PM1	0	0	C12	C13	C14
PM2	0	0	0	C23	C24
PM3	0	0	0	0	C34
PM4	0	0	0	0	0

Fig. 5. N/w Connectivity among PMs.

X ij	PM0	PM1	PM2	PM3	PM4
PM0	0	X01	X02	X03	X04
PM1	0	0	X12	X13	X14
PM2	0	0	0	X23	X24
PM3	0	0	0	0	X34
PM4	0	0	0	0	0

Fig. 6. Data communication map between all of the PMs.

$$Cost_{ij} = \begin{cases} \dfrac{X_{ij}}{C_{ij}} & i \neq j \\ 0 & \mathrm{i = j} \end{cases}. \tag{14}$$

To recap, X_{ij} equals the quantity of data transferred between the PM_i and PM_j, and C_{ij} is their network connection capacity (as previously stated).

4.3. Consumption of network power

The suggested VM placement scheme's third purpose is to save energy by reducing the power consumption of network components. The overall power usage of the network switches or F_3 is computed using Equation (15) in this VM placement strategy:

$$F_3 = \sum_{i}^{ns} Ps_i \times Bs_i + \sum_{j}^{nl} pl_j \times Bl_j, \tag{15}$$

where *ns* denotes the number of switches in this equation. Ps_i is the energy usage of the *ith* switch in the network, whereas Bs_i is a binary number indicating whether the *ith* switch is on or off. Additionally, *nl* is the sum of number of links in the switches, while pl_j is a binary value indicating whether or not the *jth* link is active. Thus, pl_j is 1 while the *jth* switch is active; otherwise, it is zero. Due to additional power is provided to each enabled switch connection, the second portion of this equation verifies the topology of all activated links.

4.4. Proposed VM placement algorithm

In this scheme, we define a determined number of agents to find an optimum solution. In Fig. 7, a sample solution for the problem, representing which VM is assigned to the PM. For instance, in this sample solution, VM (1) and VM (3) are assigned to PM (a) to PM(c), respectively.

To address VM placement problem, we employed a discrete version of seagull algorithm which is proposed by Dhiman et al. [31]. Algorithm 1 shows the pseudocode of multi-objective Seagull optimization algorithm in VMP. In this version of seagull algorithm, first we initialize the position of the agents randomly and the feasibility of each solutions because of the resource constraints, if the proposed solution exceeds the resource limitation, it will be replaced with a possible one, also the corresponding objective of each agent is calculated. In the next step to imitate the attack and migration behavior of seagulls, we employed the crossover and mutation strategies which is proposed in NSGA-II algorithm.

VM (1)	VM (2)	VM (3)	VM (4)	…	VM (n)
PM (a)	PM (a)	PM (c)	PM (d)	…	PM (z)

Fig. 7. A sample solution to VMP.

Algorithm 1
Multi-objective Seagull algorithm in VMP Pseudocode.

```
Input: Initializing the agent Population
Output: Best-found solution by the algorithm
Initialize the agents
Compute cost function of each agent
while y < max iteration do
    for each search agent do
        Apply mutation and crossover on the solutions as migration and seagull
        behavior
    end for
    Compute the costs involved with using all search agents.
    Find the most effective solutions by employing the most recent search agents.
    Hold fast to the most effective solution.
end while
return Best-found solution by the algorithm
```

In this step, the outcome solution is checked with the aim of resource constraints and the cost of each solution is calculated using Equation (8). In each iteration the cost of each agent is compared with the best ever found solution and if a solution with lower cost value recognized, the best solution will be replaced by the new efficient one. Consequently, the best solution of the all iterations is returned as the optimum solution.

5. Performance evaluation

To investigate the developed virtual machine placement method, comprehensive simulations are performed using Matlab. In this simulation, all aspects of the Edge data centers, including network architecture, available and consumed resources, are considered to simulate the real edge computing environment. To test our plan, we compared our approach with VM allocation policy simple and CPU greedy, which are adapted from Cloudsim simulator [20]. Our strategy considers the power usage of network switches while making placement selections. To consider this element, the network topology should be specified. As it can be seen in Fig. 8, in our scenario, the data centers are assumed to employ Three-Tier and VL2 topologies, and the electricity consumption of the switches is calculated in both cases. Table 3 illustrates the number of components in each network topology in our simulation.

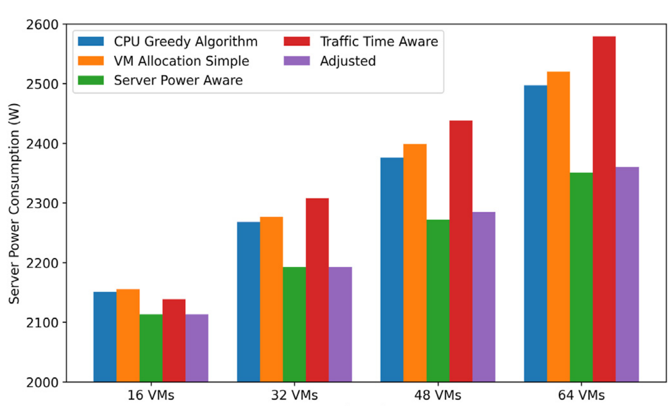

Fig. 8. VL2 server power consumption.

Table 3
Network elements.

Topology	VL2	Three-Tier
PM	16	16
Switch(1 Gigabit)	4	6
Switch(10 Gigabit)	8	12
Link (1 Gigabit)	16	16
Link (10 Gigabit)	16	32

The following scenarios evaluate the energy consumption of network components.

– A single gigabit switch consumes 30 Watts.
– A ten gigabit switch consumes 200 Watts.
– A single gigabit link consumes 0.4 Watts.
– A ten gigabit link consumes 6 Watts.

This section offers simulation results for the data center's network using VL2 and three-tiered topologies. Our suggested VM placement approach is compared to both the Greedy VM placement algorithms and VM Allocation Policy in these simulations. The simulated scenarios used in these evaluations assess the following primary factors.

– Power consumption of physical machines.
– Power usage of network switches.
– Total active connections in network switches.
– # of active switches in the data center network.
– % rate of CPU.

16 physical machines were examined in each of these simulations as part of the data center's network. Additionally, PMs in both in these scenarios are heterogeneous; therefore, two distinct types of physical machines are specified, eight of which own the following specifications.

– CPU: 117160 MIPS.
– Pmax: 129 Watts.
– Pidle: Pmax/2 Watts.

The remaining eight physical machines are defined as follows.

– CPU: 97125 MIPS.
– Pmax: 210 Watts.
– Pidle: 129 Watts.

Additionally, each simulation is performed for 16 VMs, 32 VMs, 48 VMs, and 64 VMs, and in every scenario, two distinctive kinds of virtual machines are considered.

– CPU speed of 10000 MIPS.
– CPU speed of 8000 MIPS.

The second VM category relates to an edge device, while the first VM type refers to a cloud node. In our approach, we proposed three different methods based on the optimization objectives as indicated in Table 4.

Each approach focuses on different goals depending on the value of each coefficient: the power-aware approach focuses on the power usage of the PMs, the traffic time aware method aims to reduce the amount of data transmission time, and finally, the adjusted version uses a balanced coefficient to keep the goals in balance. Figs. 8 and 9 compare the power usage (wattage) of physical equipment in the middle of the data. As previously stated, each PM's energy consumption is directly related to its CPU load, which grows as additional VMs are added to the network. So, as can be seen in these charts, the power consumption of real computers rises as the number of virtual machines grows. This illustration shows how the proposed system, by utilising different values for the cost function's coefficients, has the potential to significantly cut the energy consumption of the physical machines during the simulation. Our

Table 4
Coefficients description.

Name Of Approach	Coefficients
Server power aware	$\alpha = 1, \beta = 0, \pi = 0$
Traffic time aware	$\alpha = 0, \beta = 0, \pi = 1$
Adjusted	$\alpha = 0.001, \beta = 0.005, \pi = 1$

method results in a 5.5% percent reduction in server power usage. In each case, the entire time when network switches are enabled is shown in Figs. 10 and 11. Network switches, as mentioned in the previous sections, offer a way for VMs on different physical machines to communicate with one other. The outcomes of these simulations are provided for 16 VMs, 32 VMs, 48 VMs, and 64 VMs, and in most situations, our technique delivers better results and requires less time to apply network changes. In the instance of 16 VMs, our technique lowered the overall activation time of network components by 5 and 3 s, respectively, when compared to VM allocation simple and CPU greedy for VL2 and three-tier. In our approach, this value in 32 and 48 takes 70% less time. Finally, our method saves more than 48 s for 64 virtual machines.

Figs. 12 and 13 evaluate the relative power requirements of various network nodes across four different virtual machine (VM) loads on the 16 physical machines (PMs). The proposed method is tested in four separate cases with different values for the cost function's coefficients, similar to the prior simulation findings. As was previously indicated, the aforementioned coefficients modify the effect of network traffic and the energy used by PMs and network switches. As this diagram shows, using our solution will drastically reduce the energy consumption of every network switch. These simulations are run in four scenarios in which 16 VMs, 32 VMs, 48 VMs, and 64 VMs are examined for placement on the ECDC's sixteen physical computers. The quantity of energy used by the network is proportional to the number of virtual machines. In both Figs. 12 and 13, the energy usage of 16 and 32 VMs is lowered by about 90%. In the

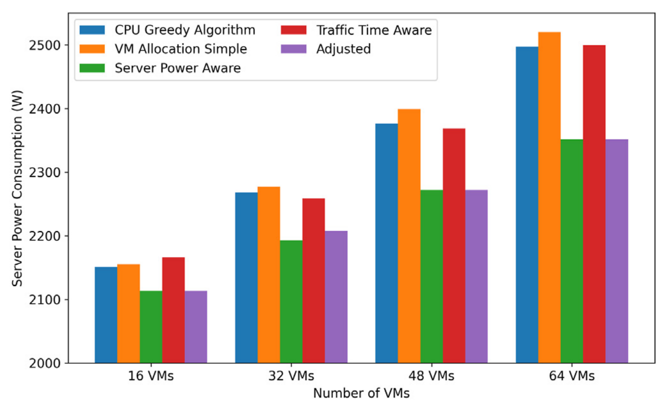

Fig. 9. Three-tier server power consumption.

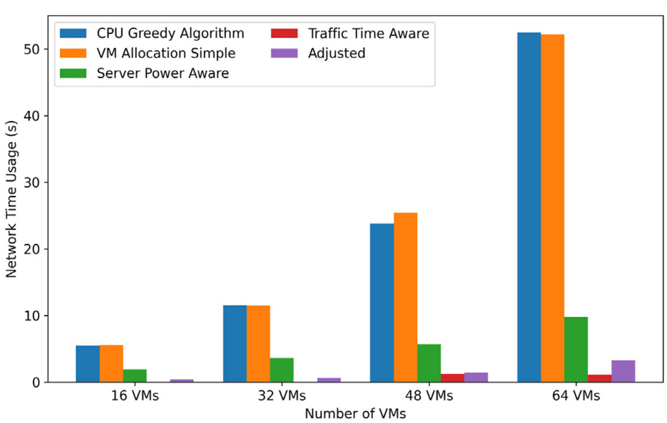

Fig. 10. VL2 network time usage.

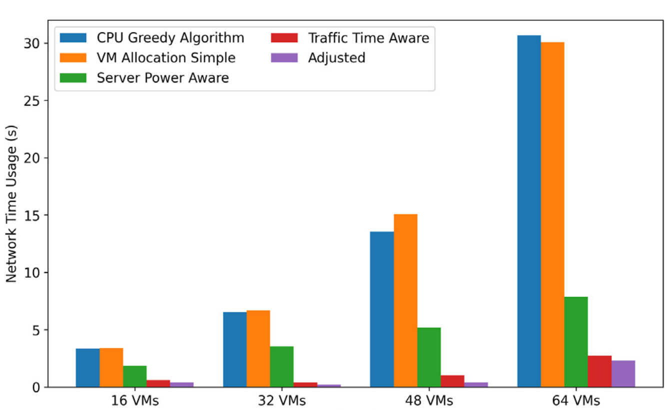

Fig. 11. Three-tier network time usage.

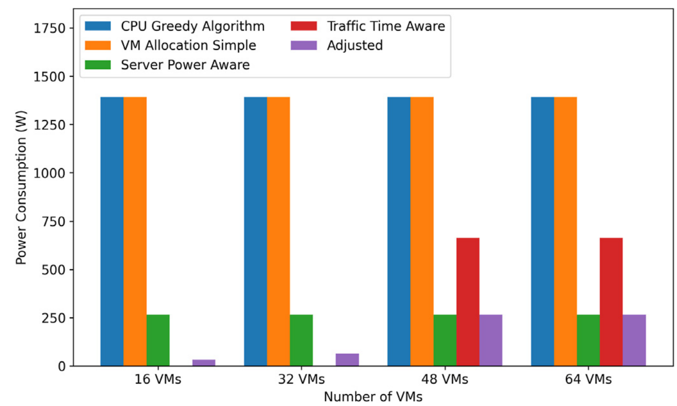

Fig. 12. VL2 network power consumption.

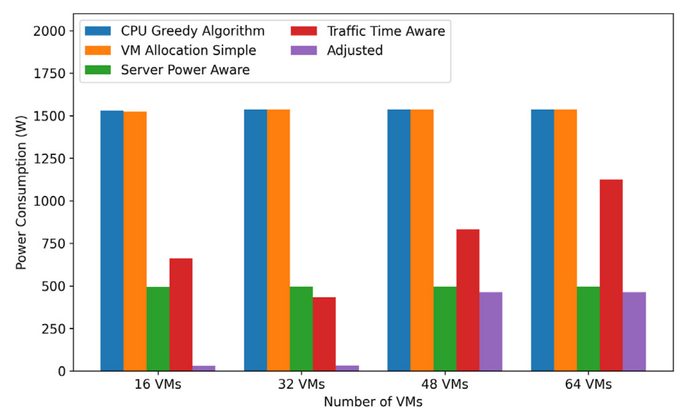

Fig. 13. Three-tier network power consumption.

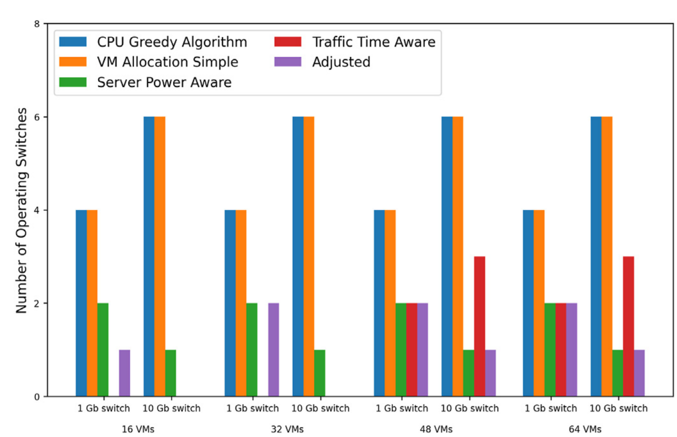

Fig. 14. VL2 active switches.

instances of 48 and 64 virtual machines, our suggested solution uses 80% less energy than the standard VM allocation algorithm and the CPU Greedy algorithm. Figs. 14 and 15 show how many 10-gigabit and 1-gigabit switches are operational in each case for the examined approaches, respectively. It is important to mention that the number of active switches is related to the quantity of data transferred between physical computers and the VM placement process, not to the number of employed VMs. The proposed solution results from the simulation are displayed in these graphs for four unique scenarios where alternative values for the cost function coefficients are utilized. As seen in these figures, the suggested seagull optimization VM placement algorithm outperforms the other two VM placement techniques in the majority of circumstances.

Figs. 16 and 17 show the number of active connections in 10 gigabit and 1 gigabit switches of the network in four scenarios with varying amounts of virtual machines. As a result of these facts, it can be stated that as the number of virtual machines and their transferred data rises, the number of active connections in network switches increases proportionately. As seen in these numbers, our system is far more effective in reducing the total active connections than the other two scheme. The reason that our approach can achieve good performance results from the large solution space that seagull based approach can search, and the proposed approach can balance the trade-offs between traffic and energy consumption via optimized VM placement decisions.

6. Conclusions and future scope

Cloud data centers are comprised of a variety of energy-intensive technologies, including physical servers and switch devices. However,

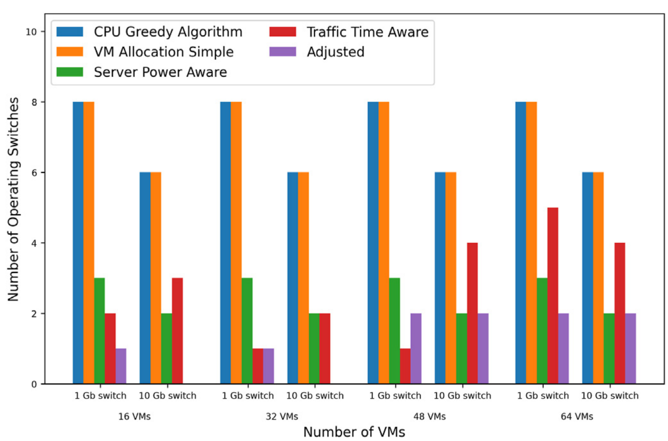

Fig. 15. Three-tier active switches.

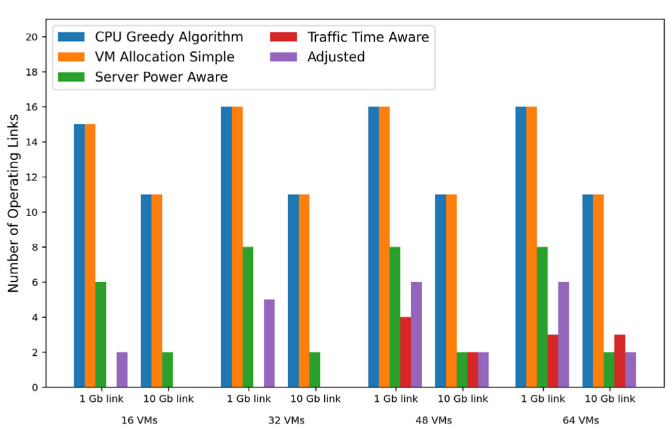

Fig. 16. VL2 active links.

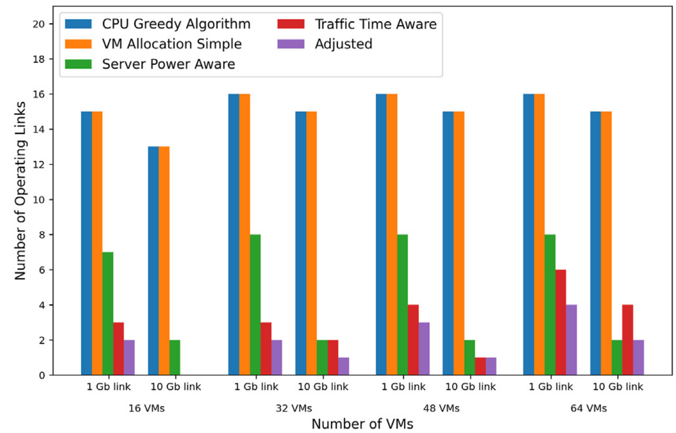

Fig. 17. Three-tier active links.

the power usage of these devices raises the ongoing data centers' expenses and may possibly pose major difficulties due to excessive heat. In this paper, we proposed an algorithm to optimize the placement of VMs to reduce the energy consumption and network traffics. We modeled the scheduling problem as a multi-objective optimization problem and use seagull based approach to solve it. Our solution not only saves power for physical equipment and network parts, but it also reduces network traffic through concentrating interacting VMs on the same PMs. This minimizes the power usage of switches and their connections while boosting the data center's output. Finally, the suggested Seagull VM placement approach was compared to the Greedy VM placement and baseline VM placement algorithms in the Three-Tier and VL2 network topologies. By implementing this strategy, we can cut down on network traffic by 70% and electricity usage by 80% without compromising on any other QoS metrics. Both network topologies were found to reduce the amount of time switches are activated and the amount of power they consume thanks to our system's simulation results.

Even though the proposed technique addressed a variety of objectives with efficient performance, it is still open of being enhanced. Containers will eventually replace virtualization technology, which is now utilized in our method. Application virtualization is made possible by the light-weight container technology [32,33]. Containerization can increase CPU speed and energy efficiency. Using AI, fog nodes may learn about their surrounding workloads and change on the fly to improve quality of service. It also decreases the need for electricity and the overall price of the supporting infrastructure. As a last step, this strategy may be applied via serverless computing or function as a service to increase scalability and decrease cost due to the lack of requirement for server configuration during application deployment. A serverless computing architecture makes it possible for applications to scale with little effort.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work is partially funded by Chinese Academy of Sciences President's International Fellowship Initiative (Grant No. 2023VTC0006), National Natural Science Foundation of China (No. 62102408), Shenzhen Industrial Application Projects of undertaking the National key R & D Program of China (No.CJGJZD20210408091600002) and Shenzhen Science and Technology Program (Grant No. RCBS20210609104609044).

References

[1] M. Masdari, S.S. Nabavi, V. Ahmadi, An overview of virtual machine placement schemes in cloud computing, J. Netw. Comput. Appl. 66 (2016) 106–127 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804516000291.

[2] W. Shi, J. Cao, Q. Zhang, Y. Li, L. Xu, Edge computing: vision and challenges, IEEE Internet Things J. 3 (5) (2016) 637–646.

[3] X. Wang, Y. Han, V.C.M. Leung, D. Niyato, X. Yan, X. Chen, Convergence of edge computing and deep learning: a comprehensive survey, IEEE Communications Surveys & Tutorials 22 (2) (2020) 869–904.

[4] N. Hassan, K.-L.A. Yau, C. Wu, Edge computing in 5g: a review, IEEE Access 7 (2019), 127 276–127 289.

[5] Q. Huang, S. Shao, H. Zhang, C. Tian, Development and composition of a data center heat recovery system and evaluation of annual operation performance, Energy 189 (2019), 116200 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S036054421931895X.

[6] H. Zhao, J. Wang, F. Liu, Q. Wang, W. Zhang, Q. Zheng, Power-aware and performance-guaranteed virtual machine placement in the cloud, IEEE Trans. Parallel Distr. Syst. 29 (6) (2018) 1385–1400.

[7] M. Dayarathna, Y. Wen, R. Fan, Data center energy consumption modeling: a survey, IEEE Communications Surveys & Tutorials 18 (1) (2016) 732–794.

[8] I. Alam, K. Sharif, F. Li, Z. Latif, M.M. Karim, S. Biswas, B. Nour, Y. Wang, A survey of network virtualization techniques for internet of things using sdn and nfv, ACM Comput. Surv. 53 (2) (apr 2020), https://doi.org/10.1145/3379444 [Online]. Available:.

[9] M.S. Bonfim, K.L. Dias, S.F.L. Fernandes, Integrated nfv/sdn architectures: a systematic literature review, ACM Comput. Surv. 51 (6) (feb 2019), https://doi.org/10.1145/3172866 [Online]. Available:.

[10] K. Dolui, S.K. Datta, Comparison of edge computing implementations: fog computing, cloudlet and mobile edge computing, in: 2017 Global Internet of Things Summit (GIoTS), 2017, pp. 1–6.

[11] C.Z. Zhao Ziming, Fang Liu, X. Nong, Edge computing: platforms, applications and challenges, J. Comput. Res. Dev. 55 (2) (2018) 327 [Online]. Available: https://crad.ict.ac.cn/EN/abstract/article_3633.shtml.

[12] H. Shukur, S. Zeebaree, R. Zebari, D. Zeebaree, O. Ahmed, A. Salih, Cloud computing virtualization of resources allocation for distributed systems, J. Appl. Sci. Technol. Trends 1 (3) (2020) 98–105.

[13] J. Singh, et al., Fog computing: A taxonomy, systematic review, current trends and research challenges, J. Parallel Distr. Comput. 157 (2022) 56–85 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0743731521001349.

[14] S. Barshandeh, M. Masdari, G. Dhiman, V. Hosseini, K.K. Singh, A range-free localization algorithm for iot networks, Int. J. Intell. Syst. (2021).

[15] S. Iftikhar, et al., Ai-based fog and edge computing: a systematic review, taxonomy and future directions, Internet of Things 21 (2023), 100674.

[16] W. Lin, C. Xiong, W. Wu, F. Shi, K. Li, M. Xu, Performance interference of virtual machines: a survey, ACM Comput. Surv. (nov 2022), https://doi.org/10.1145/3573009 just Accepted. [Online]. Available:.

[17] K. Li, H. Zheng, J. Wu, Migration-based virtual machine placement in cloud systems, in: 2013 IEEE 2nd International Conference on Cloud Networking (CloudNet). Plus 0.5em Minus 0.4emIEEE, 2013, pp. 83–90.

[18] J. Darrous, S. Ibrahim, A.C. Zhou, C. Perez, Nitro: network-aware virtual machine image management in geo-distributed clouds, in: 2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), 2018, pp. 553–562.

[19] M. Abdel-Basset, L. Abdle-Fatah, A.K. Sangaiah, An improved lévy based whale optimization algorithm for bandwidth-efficient virtual machine placement in cloud computing environment, Cluster Comput. 22 (4) (2019) 8319–8334.

[20] R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A. De Rose, R. Buyya, Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Software Pract. Ex. 41 (1) (2011) 23–50.

[21] A.S. Abohamama, E. Hamouda, A hybrid energy–aware virtual machine placement algorithm for cloud environments, Expert Syst. Appl. 150 (2020), 113306.

[22] M. Ghobaei-Arani, A.A. Rahmanian, M. Shamsi, A. Rasouli-Kenari, A learning-based approach for virtual machine placement in cloud data centers, Int. J. Commun. Syst. 31 (8) (2018) e3537.

[23] F. Alharbi, Y.-C. Tian, M. Tang, W.-Z. Zhang, C. Peng, M. Fei, An ant colony system for energy-efficient dynamic virtual machine placement in data centers, Expert Syst. Appl. 120 (2019) 228–238 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417418307498.

[24] R. Shaw, E. Howley, E. Barrett, An energy efficient anti-correlated virtual machine placement algorithm using resource usage predictions, Simulat. Model. Pract. Theor. 93 (2019) 322–342, modeling and Simulation of Cloud Computing and Big Data. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569190X18301448.

[25] S.S. Nabavi, et al., Tractor: traffic-aware and power-efficient virtual machine placement in edge-cloud data centers using artificial bee colony optimization, Int. J. Commun. Syst. 35 (1) (2022) e4747.

[26] A.M. Maia, Y. Ghamri-Doudane, D. Vieira, M.F. de Castro, A multi-objective service placement and load distribution in edge computing, in: 2019 IEEE Global Communications Conference (GLOBECOM), 2019, pp. 1–7.

[27] G. Dhiman, V. Kumar, Seagull optimization algorithm: theory and its applications for large-scale industrial engineering problems, Knowl. Base Syst. 165 (2019) 169–196.

[28] V. Kumar, D. Kumar, M. Kaur, D. Singh, S.A. Idris, H. Alshazly, A novel binary seagull optimizer and its application to feature selection problem, IEEE Access 9 (2021), 103 481–103 496.

[29] G. Dhiman, K.K. Singh, M. Soni, A. Nagar, M. Dehghani, A. Slowik, A. Kaur, A. Sharma, E.H. Houssein, K. Cengiz, Mosoa: a new multi-objective seagull optimization algorithm, Expert Syst. Appl. 167 (2021), 114150.

[30] Y. Gao, H. Guan, Z. Qi, Y. Hou, L. Liu, A multi-objective ant colony system algorithm for virtual machine placement in cloud computing, J. Comput. Syst. Sci. 79 (8) (2013) 1230–1242 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0022000013000627.

[31] G. Dhiman, et al., Emosoa: a new evolutionary multi-objective seagull optimization algorithm for global optimization, Int. J. Mach. Learn. Cybern. 12 (2) (2021) 571–596.

[32] M. Xu, C. Song, H. Wu, S.S. Gill, K. Ye, C. Xu, esdnn: deep neural network based multivariate workload prediction in cloud computing environments, ACM Trans. Internet Technol. 22 (3) (2022) 1–24.

[33] S.S. Gill, et al., Ai for next generation computing: emerging trends and future directions, Internet of Things 19 (2022), 100514 [Online]. Available: https://www.sciencedirect.com/science/article/pii/S254266052200018X.