TRACTOR: Traffic-aware and Power-efficient Virtual Machine Placement in Edge-Cloud Data Centers Using Artificial Bee Colony Optimization

Sayyid Shahab Nabavi[1], Sukhpal Singh Gill [2], Minxian Xu[3], Mohammad Masdari[1], Peter Garraghan[4]

[1]Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran
[2]School of Electronic Engineering and Computer Science, Queen Mary University of London, UK
[3]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
[4]School of Computing and Communications, Lancaster University, UK

shahab.nab@gmail.com, s.s.gill@qmul.ac.uk, mx.xu@siat.ac.cn, m.masdari@iaurmia.ac.ir, p.garraghan@lancaster.ac.uk

Abstract: Technology providers heavily exploit the usage of Edge-Cloud Data Centers (ECDC) to meet user demand while the ECDC are large energy consumers. Concerning the decrease of the energy expenditure of ECDCs, task placement is one of the most prominent solutions for effective allocation and consolidation of such tasks onto physical machine (PM). Such allocation must also consider additional optimizations beyond power and must include other objectives, including network-traffic effectiveness. In this study, we present a multi-objective Virtual Machine (VM) placement scheme (considering VMs as fog tasks) for ECDCs called TRACTOR, which utilizes an Artificial Bee Colony optimization algorithm for power and network-aware assignment of VMs onto PMs. The proposed scheme aims to minimize the network traffic of the interacting VMs and the power dissipation of the data center's switches and physical machines. To evaluate the proposed VM placement solution, the VL2 (Virtual Layer 2) and three-tier network topologies are modeled and integrated into the CloudSim toolkit to justify the effectiveness of the proposed solution in mitigating the network traffic and power consumption of the ECDC. Results indicate that our proposed method is able to reduce power energy consumption by 3.5% whilst decreasing network traffic and power by 15% and 30%, respectively, without affecting other Quality of Service parameters.

Keywords: Cloud Computing, VM Placement, Artificial Bee Colony, Power Consumption, Network Traffic, Cloud Data Centers

1. Introduction

Cloud computing provides the ability for omnipresent, accessible, upon request network accessibility to a group of shared adjustable virtualized resources, which include network bandwidth, CPU, storage, and various other services. Such virtualized resources are able to be accessed rapidly and released with a minimum management endeavor or service provider communication [1, 2]. Moreover, these resources are provided to customers as a service on the Internet-based on the pay-for-use pricing design [3, 4]. This provisioning model allows cloud users to reduce operation and maintenance costs [5, 6].

Virtualization is considered as an indispensable specification of cloud computing, provides enhancements to location independence, resource pooling [7], rapid elasticity, and security [8, 9]. By allocating Virtual Machines (VMs) to physical machines, data center administrators are able to assign multiple tasks/users onto the same host for better data center resource utilization [10, 11]. However, to be more effective, fog tasks should be placed on appropriate Physical Machines (PMs) using a process called VM placement [12, 13] shown in Figure 1. Proper placement of VMs is very important in the cloud environment to effectively improve the power effectiveness and resource utilization in the edge-cloud infrastructure [14, 15]. However, inefficient VM placement can increase the number of active PMs, which leads to more energy consumption and network traffic [16, 17]. To prevent these problems, numerous researches have been conducted to find optimal VM placement on data centers' physical machines [18, 19].

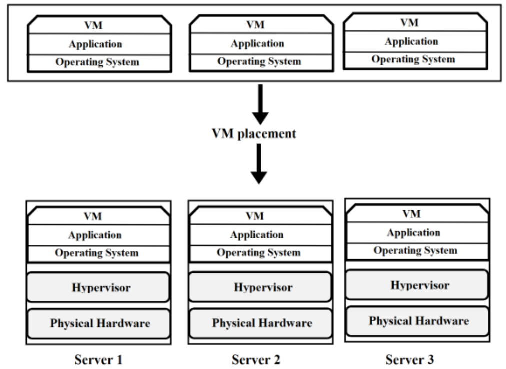

Figure 1: VM Placement

In [3], we have studied VM placement schemes and analyzed their properties, features, and limitations. Nowadays, due to the rising concerns of energy costs in the Edge-Cloud Data Center (ECDC) [53] and emerging bandwidth-intensive applications, power consumption and network traffic of VMs should be taken into account in the VM placement operations [20]. Swarm Intelligence algorithms can be used to design VM placement policy, and it has been reported that the optimization algorithm of Artificial Bee Colony (ABC) is the foremost optimization algorithm among all the others [21] to find an optimum solution for multi-objective optimization and NP-complete problem such as resource scheduling, resource allocation, etc. There is a need for an effective technique for VM placement while reducing network traffic and energy consumption simultaneously [22]. In this paper, a **TRA**ffic-aware and power effi**C**ient VM placemen**T** algorithm for edge-cl**O**ud data cente**R** called ***TRACTOR*** is proposed by using the ABC optimization algorithm to improve the effectiveness of the network traffic and power consumption. ABC algorithm is an optimization algorithm proposed for solving numerical problems [21]. This algorithm is on the basis of the intelligent behavior of the honeybee swarms and replicates their foraging behaviors [23, 24]. The purpose of this research work has been the formulation of the VM placement problem as a multi-objective optimization problem to lower the power consumption of the PMs and data center switches. Moreover, it considers the amount of data transfer between VMs and is aimed to decrease the network traffic between PMs by the assignment of the interacting VMs to the appropriate machines in an efficient manner. Extensive simulations in the CloudSim toolkit indicate the effectiveness of our proposed solution in increasing the performance of the ECDC. The ***main contributions*** of the paper are mentioned as below:

- Modeling VM placement as a multi-objective problem aimed to decrease the network traffic and power expenditure of the PMs and network switches.
- Presenting an ABC optimization algorithm based VM placement policy for ECDCs.
- Extensive simulation of the proposed VM placement algorithm in the CloudSim toolkit based on the VL2 (Virtual Layer 2) and three-tiered network topologies.
- The performance of the proposed VM placement solution and the CloudSim's VM allocation policy, and the Greedy VM placement algorithms are compared.

The rest of the article is structured as follows: Section 2 displays a brief overview of the state-of-the-art VM placement schemes presented for cloud computing. Section 3 briefly illustrates the algorithm of the artificial bee colony optimization, and section 4 presents our proposed multi-objective VM placement solution. Section 5 demonstrates the performance assessment and the results of the experiment. And at last, in section 6, the concluding remarks and future research works are depicted.

2. Related Works

In this section, a review of the existing research work in the area of VM placement for ECDCs is presented. VM placement in ECDCs is an NP-complete problem, and, in the literature, there is the presentation of numerous placement schemes [3]. This section is categorized into three groups based on their aims:

2.1 Power and Energy-efficiency based VM Placement Policies

In [25], Huang et al. provided an energy efficient VM placement solution on the basis of the model of multi-dimensional space partition. In this scheme, when a new VM placement should be performed, the resource consumption for each feasible physical machine is checked, and then the most appropriate physical machine is selected according to their proposed model. Thus, the data center resource is able to be utilized in an equitable way and create smaller resource fragments, consuming less energy. In [26], Yang et al. presented VM Placement and Traffic Configuration Algorithm (VPTCA). In order to minimize the power consumption in the data center, they use the genetic algorithm. As decisions are being made on VM placement, VPTCA takes the three above mentioned factors and their relationship into account. VPTCA outperforms the First Fit and Elastic Tree placement algorithms in the light traffic and heavy traffic load in the data center. Also, good network performance having a low average end-to-end delay, a low ratio of the dropped packets, and high transmission throughput can be achieved by VPTCA. In [29], Xu et al. presented a heuristic scheme called PSABC (Power Saving ABC) that applies an improved ABC approach, binary search, and Boltzmann selection policy to provide a reduction of power. In the proposed PSABC algorithm, the Bayes theorem is used to converge rapidly to the optimal of the globe and making the performance of live VM migration better. In comparison with the random technique and other available optimization approaches, this scheme has lowered extra power consumption and breakdown events of the live VM migration to provide green ECDCs.

EAGLE is another energy-efficient online VM placement scheme proposed in [27], which provides balanced resource utilization. A multi-dimensional space partition model is proposed with the aim of designing the resource-balanced placement. In comparison with the traditional greedy VM placement algorithms, including the first-fit algorithm, placing each VM on any available physical machine, this algorithm creates fewer fragments of cloud resources and uses less energy. In [28], Zhao et al. investigated the equilibrium between reducing PM power and securing VM performance and put a power-aware and performance-guaranteed VMP (PPVMP) forward. At first, they examine the relationship between the consumption power and CPU utilization to make a non-linear power method, which is essential for the VMP. In this paper, a multi-objective plan for VM placement is implemented, whose objective is to minimize PM power consumption and ensure VM performance. In [34], Alharbi et al. have presented a new model of the Ant Colony Scheme (ACS) so that energy usage in cloud data centers is lowered. To examine the efficiency of the proposed algorithm, various simulation-based experiments have been conducted on small, medium, and large scale data centers. They also compared the results with two different ACS methods and the FFD algorithm. After comparing, it is proven that their new approach exhibits acceptable results.

2.2 Network-Awareness based VM Placement Policies

Piao et al. [29] presented a policy in which the VM is placed based on the network I/O and provide a VM migration and placement approach to handle the conditions where the network connection's condition reduces the application performance. In this scheme, the placement policy improves the data access by placing a VM on a PM where the data will be transferred faster. In the proposed scheme, VM migration is initialized when the period of the data transfer surpasses a certain threshold owing to the unstable network. Afterward, another improved machine is selected considering the present network status, and the VM is migrated to this physical machine for higher efficiency. In terms of the proposed VM allocation policy, the data can be accessed faster by the task, as the hosted VM is placed on the PM that utilizes a better network bandwidth. In [30], Fang et al. proposed VM Planner, which handles a network-wide power that tries to improve the VM placement and traffic flow routing. They consider the power consumption in VM placement and traffic routing. VM placement is designed as a combinatorial optimization problem, and decomposes it into three separate classic NP-complete problems, including power-aware inter-VM traffic flow routing, traffic, and distance aware VM grouping to server-rack mapping.

2.3 Multi-objective based VM Placement Policies

In [12], the VM placement is considered as a multi-objective optimization problem to cut the wastage of the resource and power usage. An adjusted version of the ant colony system algorithm is presented to be dealt with the big solution space in large ECDCs. They have made a comparison between the performance of their proposed approach and a multi-objective genetic algorithm, two single-objective algorithms, and a bin-packing algorithm. In [31], the authors presented a VM placement scheme to boost the data center's resource utilization and reduce the runtime reconfiguration costs. They applied the vector arithmetic for modeling multi-dimensional resource usage balancing and increasing resource utilization. After that, they provide a two-level runtime reconfiguration method to include local adjustment and VM parallel migration and satisfy the migration of VM and total migration time. In [34], Abdel-Basset et al. concentrated on how to solve the VM placement problem regarding the existing bandwidth being formulated as a variable-sized bin packing problem. Also, a novel bandwidth allocation approach is engendered and combined with an optimized Whale Optimization algorithm variant, which is called an improved levy-based whale optimization algorithm. The results obtained were equivalent to many optimizations.

In [32], Satpathy et al. proposed a comprehensive model including a scheduling structure to arrange and queue a considerable set of VMs and Crow Search based VM Placement, where a multi-objective VM placement algorithm is used to utilize resource allocation and reduce power consumption at ECDCs. Further, three VM migration approaches, including serial, parallel, improved serial as an inseparable part of the cloud computing environment, is considered. In [7], a novel algorithm was proposed for the Infrastructure as a Service (IaaS) cloud to achieve two important objectives. The first objective of the proposed approach is minimizing power consumption in cloud data centers by reducing active PMs. The second objective concentrates on resource utilization, and the strategy that they conducted is able to find unbalanced VMs conveniently. They have proposed a multi-objective PSO algorithm in

[33]. It considers physical resource utilization and link failure rate as an optimization target; also, user requests are satisfied efficiently, and results illustrate that link failure and energy consumption are decreased notably.
In [34], Gharehpasha et al. proposed a novel approach using a combination of the Sine–Cosine Algorithm and Swarm Algorithm for virtual machine placement. In the proposed method, reducing power consumption and resource wastage are considered by reducing the number of active physical machines. Tuli et al.[35] offered an A3C real-time scheduler for stochastic Edge-Cloud environments. Also, for capturing a vast number of host and task parameters together with temporal patterns, R2N2 architecture is used to give useful scheduling determinations. The conducted experiment on a real-world data set indicates a notable enhancement in energy expenditure, latency, Service-Level-Agreement comparison with previous works. Azizi et al.[36] proposed a large-scale CDC with heterogeneous and multidimensional resources using a Greedy Randomized VM placement (GRVMP) algorithm. The method is established on two characteristics, including placing VMs on more power-efficient PMs to reduce power consumption and resource utilization. The experiment results illustrate that the technique performs effectively in the mentioned metrics. A framework named FogBus that helps end-to-end IoT-Fog(Edge)-Cloud integration is proposed by Tuli et al. [37]. The introduced model promotes IoT application deployment and resource utilization in comparison with existing frameworks and the influence of different FogBus adjustments on system efficiency. The results indicate that the FogBus is lightweight and responsive in comparison to Fog and Cloud infrastructures and it can be modified based on situation requirements.

In [38], Li et al. introduced a novel approach to minimize the consumption of energy in order to maximize resource utilization in VM placement. There have been used two algorithms on the basis of chemical reaction optimization (CRO), that's to say CVP (CRO-VMP-Permutation) and CVV (CRO-VMP-Vector). The outcome results are compared with Best Fit Decreasing (BFD), First Fit Decreasing (FFD), Cuckoo Search Optimization (CSO), and Reordered Grouping Genetic Algorithm (RGGA). Consequently, it is proven that their approach can mostly perform better. In [39], Mohammad hosseini et al. presented a VM placement approach, namely a balance-based cultural algorithm for virtual machine placement (BCAVMP) where a different fitness function was applied. This approach considers the balanced utilization of resources through the sum of the balance vector length and fitness function to reduce energy consumption. Outcome results prove that resource utilization and power consumption can be enhanced in cloud data centers. In [40], an Energy-oriented Flower Pollination Algorithm (E-FPA) for VM placement in cloud data centers was introduced by Yau Gital et al. with the purpose of decreasing data center energy consumption and enhancing resource utilization of the available physical resources.

Table 1: Comparison of the Existing Techniques with Proposed Scheme (TRACTOR)

Scheme	Traffic-aware	Energy-aware	Algorithm Type	Multi-objective Optimization Problem	Cloud Computing	Edge Computing
[12]		✓	Ant Colony Optimization	No	✓	
[25]		✓	Heuristic Technique	No	✓	
[26]		✓	Genetic Algorithm	No	✓	
[18]		✓	Heuristic Technique	No	✓	
[29]	✓		Banker Algorithm	No	✓	
[30]	✓	✓	Heuristic Technique	Yes	✓	
[21]		✓	Artificial Bee Colony	No	✓	
[31]		✓	Heuristic	No	✓	
[41]		✓	Ant Colony Optimization	No	✓	
[42]	✓	✓	Memetic Algorithm	Yes	✓	
[43]		✓	Learning Automata	No	✓	
TRACTOR (this work)	✓	✓	**Artificial Bee Colony**	**Yes**	✓	✓

Table 1 presents a comparison of the proposed VM placement investigations. As it is depicted in the Table 1, there are few schemes which take both traffic of the network and power consumption in the VM placement into account. Also, few power-aware placement schemes even consider the network switches and their links to power consumption. Our adjustable multi-objective approach considers the power consumption of servers and network elements and the network traffic. It is feasible to consider each goal separately or consider a balance between them. Moreover, the type of applied algorithm, mostly bio inspirational and metaheuristic algorithms, is specified in each investigation. Among these schemes, it can be seen that our proposed approach is implemented in the edge computing environment while the other research works are studied in the cloud computing environment.

3. Artificial Bee Colony Algorithm

Karaboga [44] et al. compared the performance of the ABC algorithm with other bio-inspired algorithms such as GA, PSO, and DE using 50 different functions. The outcome of this study indicates that the ABC algorithm performs better in more than half of these scenarios. The proposed algorithm is adjusted to enable a Traffic-Aware and Power Efficient approach to heterogeneous VMs on PMs. The framework determines an optimal solution that also establishes equilibrium between the global search exploration and the local search exploitation. Resource constraints, including processor, storage, and memory of PMs, are considered in this framework.

We used the ABC algorithm for virtual machine placement in the following steps:1: we define a certain number of scout bees to search the environment, each bee proposes a solution by an array with the size of the number of VMs, which is indicated in Figure 2.

Solution=[V_1 V_2 . . . V_n] where V_i is between the minimum and the maximum index of PM.

VM(1)	VM(2)	VM(3)	VM(4)	VM(5)	…	VM(n)
a	b	c	d	e	….	z

Figure 2:solution representation

For instance:solution [a x f g h]

represents the VM(3) will be placed on the host (f).

This is the way of representing a placement solution in the algorithm, which should be checked for constraints of PMs' resources. Afterward, the cost of the proposed solution by a function that we will explain in the following sections, a solution with lower cost chosen as better one, solutions with a high rate of cost rejected and replaced by new random one.

The function is based on two-part power and traffic, as discussed below:

The ABC algorithm [45] applies three sorts of bees named employed bees, scouts bees, and onlookers. Onlooker is a bee that stands by the dance area to determine choosing a food source, and the one which goes to the food source visited before being named is an employed bee. The scout bees search randomly to discover new sources. Each food position indicates a solution which is possible to the problem, and the value of food source nectar specifies the fitness value or quality of the solution, which can be computed using equation 1:

$$Fit_i = \frac{1}{1+f_i} \qquad (1)$$

Algorithm 1: Artificial Bee Colony

```
Begin
    Initial Pop()
    While remain iterations do
        Site selection for local search
        assign bees for the selected sites
        and to compute fitness
        find the bee with the best fitness
        Allocate the remaining bees to look
        for randomly
        compute remaining bees fitness
        Update Optimum Solution()
    End While
    Return BestSolution
End
```

Figure 3: The pseudocode of ABC Algorithm

In the ABC algorithm, the food position indicates a feasible solution, and at first, a fortuitously allotted initial population is created. Also, employed or onlooker bees must be initialized corresponding to the solutions in the mentioned population. In this algorithm, only one employed bee uses each food source, and the fitness of the solution specifies the amount of nectar in a food source. The food resource is selected by an onlooker bee based on the calculated probability value *P* related to the food source that is possible to be determined as indicated in equation 2:

$$P_i=\frac{Fit_i}{\sum_{n=1}^{SN} Fit_n} \qquad (2)$$

In this equation, Fit_i is the calculated fitness value of the *ith* solution, and the food source number is specified by *SN*, which needs to be equal to the number of the onlooker or employed bees [46]. The first positions of the food sources are fortuitously made, in which each employed bee is designated to a random food source. Afterward, each of the employed bees, using equation 3, determines a novel neighboring food source and calculates the amount of the nectar of the novel food source for each of them. If the new food source fitness value is greater than the old one, then the employed bee will start moving to the new food source, or else it will use the previous one.

$$V_{ij} = x_{ij} + \varnothing_{ij}\,(x_{ij} - x_{kj}) \qquad (3)$$

In equation (3), $\varnothing$ stands for an accidental number between [-1 and 1], v_i is a candidate solution, x_i for the existing solution, and x_k for a neighbor solution and $j \,\varepsilon\, \{1, 2, \ldots, D\}$z is an accidentally chosen index where D stands for the dimension of the solution vector. In this algorithm, when a position cannot be optimized after the specified number of repetitions, the food source needs to be regarded as discarded, and the employed bee of the same food source gets a scout. Then, in that position, the scout bee, using equation 4, produces a new solution randomly:

$$x^j_i = x^j_{min} + rand(0,1)(X_{max} - x^j_{min}) \qquad (4)$$

Where x^j_i is the new position of the scout bee (*i*), x^j_{min} and X_{max} are the minimum and maximum values for the dimension (*j*), respectively. *V* is produced after each candidate source position; then, the artificial bee evaluates it, and its fitness is compared with that of its old one. The new food source substitutes the old one in the memory when it provides equal or better nectar. If not, the old one will be retained in the memory. To put it differently, a voracious selection technique is applied as the selection operation between the former and the nominated one [47]. Figure 3 indicates the artificial bee colony pseudocode.

4. Levy Flight

Levy flight is an accidental model of walk that expresses the anomalous nature diffusion [48]. The Levy distribution is satisfied by its distribution, which has long-tail characteristics. Since similar methods of Levy flight are used by the flying insects in nature, such as bees and fruit flies, they have inspired several scholars to bring in Levy flight in the design of evolution to optimize the achievement of the algorithm to reach good results. As the specific method is located in the bees' collecting phase of the ABC algorithm, there will be two parts of *SN*; one part will have the job of collecting the region's optimal information based on the original algorithm, and the other one does Levy flight near the existing best global solution to optimize the capabilities of the global search. And here is its adapted equation as below:

$$x'_{ij} = x_{ij} + c_1 \frac{\varphi u}{|V|^{1/\beta}} (x_{ij} - x_{best}) \qquad (5)$$

where c_1 stands for the step adjustment factor, and normally is considered to be as 0.01; u, V are random numbers meeting usual standard distribution; θ is a constant satisfying $0 < \theta < 2$; $\varphi = \left[\frac{\Gamma(1+\theta)\sin(\pi\theta/2)}{\Gamma(\frac{1+\theta}{2}\,\theta.2^{(\theta-1)/2})}\right]^{1/\theta}$; Γ is the standard Gamma function.

5 Proposed VM Placement Solution

This section presents the proposed multi-objective VM placement approach designed to lower the power consumption of cloud data centers. This scheme is aimed to reduce the energy consumption of the PMs and the network switches by decreasing the number of active switches and active links required for the VMs' interactions. An ABC-based VM placement algorithm to achieve the mentioned objectives is offered in the present section. In the proposed ABC-based VM placement approach, a cost function is needed to compute the fitness value of each solution and make decisions about the solutions. To achieve the objectives of the proposed solution, this cost function should consist of three parts, which are physical machines' power consumption; network switches power consumption and network traffic, as discussed below. This cost function is indicated in equation 6:

$$F = \alpha F_1 + \beta F_2 + \pi F_3 \qquad (6)$$

In this equation, α, β, and π are the coefficients that are used to tune the impact of each objective in the Fitness value. Also, F_1 is the physical machines' total power consumption in the data center, and F_2 is the total network cost, which depicts the data transfer cost between physical machines. Moreover, F_3 specifies the total power consumption of the switches of the network.

5.1 Power Consumption

In the VM placement scheme, which has been put forward, the total electricity consumption of the physical machines can be completed using Equation 7:

$$F_1 = \sum_{i=1}^{n} p_i \qquad (7)$$

where F_1 is the total amount of the physical machines' power consumption, P_i is the power consumption of the *ith* physical machine, and n is the number of the physical machines of the data centers. Recent studies indicate a linear relationship existing between the physical machine's power consumption and utilization of its CPU [12]. By using this, the power consumption of each physical machine will be possible to be computed, as indicated in equation 8:

$$P_i = \begin{cases} (P_i^{max} - P_i^{idle}) * U_i^c + P_i^{idle} & U_c^i > 0 \\ 0 & otherwise \end{cases} \qquad (8)$$

In this equation, the maximum power consumption of the *ith* physical machine is P_i^{max} , in its full utilization, P_i^{idle} is the power consumption of the *ith* physical machine in idle mode, and U_i^c is the CPU utilization of the *ith* physical machine. It is worth noting that the CPU rate in the sleep mode is about half of the maximum rate. But, by placing some VMs on a physical machine, its CPU rate increases based on the VMs' CPU requests. The utilization of each physical machine can be calculated using equation 9:

$$Uic = \frac{\sum_{j=0}^{m} (xj * Rj)}{Ai} \qquad (9)$$

In this equation, *m* is the number of Virtual machines placed on the PM_i, A_i is the available CPU for PM_i, R_j is the requested CPU of the VM_j and x_j is a binary value which can be computed by equation 10:

$$Xj = \begin{cases} \text{if VMj is placed on the PMi} & 1 \\ \text{if VMj is not placed on the PMi} & 0 \end{cases} \qquad (10)$$

5.2 Network Traffic

Because VMs may execute interacting applications such as workflows, they often need to interact with each other. As indicated in Figure 4, the amount of data transmission between VMs can be specified by an upper triangular matrix. In this matrix, D_{ij} indicates the amount of the data which should be transferred between the VM_i and VM_j. In Figure 4, the sum of all elements in the matrix represents the total amount of the data that are needed to be exchanged between all VMs. Moreover, in this scheme, another upper triangular matrix is defined to indicate the link capacity between different physical machines. The speed of data transfer between VMs depends on the available bandwidth between the physical machines, which depends on the applied network topology and switches.

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

Figure 4: Required data transmission between VMs

Figure 5 indicates the available bandwidth between the physical machines. In this matrix, C_{ij} is the link capacity between the *ith* physical machine(PM_i) and *jth* physical machine(PM_j) in megabit per second.

C_{ij}	PM_0	PM_1	PM_2	PM_3	PM_4
PM_0	0	C_{01}	C_{02}	C_{03}	C_{04}
PM_1	0	0	C_{12}	C_{13}	C_{14}
PM_2	0	0	0	C_{23}	C_{24}
PM_3	0	0	0	0	C_{34}
PM_4	0	0	0	0	0

Figure 5: Link capacity between PMs

After VMs are placed on the physical machines, they may interact with each other. The matrix *X* indicated in Figure 6 shows the amount of the data that must be transferred between the PMs because of their VMs interactions.

X_{ij}	PM_0	PM_1	PM_2	PM_3	PM_4
PM_0	0	X_{01}	X_{02}	X_{03}	X_{04}
PM_1	0	0	X_{12}	X_{13}	X_{14}
PM_2	0	0	0	X_{23}	X_{24}
PM_3	0	0	0	0	X_{34}
PM_4	0	0	0	0	0

Figure 6: Required data transfer between the PMs

In this matrix, X_{ij} indicates the amount of the data that will be transferred between the *ith* physical machine(PM_i) and *jth* physical machine(PMj) in megabit per second. After a solution is found for VM placement, the total network cost can be calculated by equation 11:

$$F_2 = \sum_{i=0}^{n-1} \sum_{j=i+1}^{n-1} Cost_{ij} \qquad (11)$$

In this equation, F_2 is the total network traffic cost, and *n* shows the number of physical machines in the data center. Also, $Cost_{ij}$ indicates the cost of the data transfer between the PM_i and PM_j that can be computed by equation 12:

As outlined before, x_{ij} is the amount of the data transfer between the PM_i and PM_j and C_{ij} is the link capacity between them.

$$Cost_{ij} = \begin{cases} \frac{X_{ij}}{C_{ij}} & i \neq j \\ 0 & i = j \end{cases} \qquad (12)$$

5.3 Network Power Consumption

The third objective of the proposed VM placement scheme is the mitigation of the power consumption of the network switches. In this VM placement scheme, the total amount of the power consumption of the network switches or F_3 can be computed by equation 13:

$$F_3 = \sum_{i}^{ns} Ps_i * Bs_i + \sum_{j}^{nl} pl_j * Bl_j \qquad (13)$$

In this equation, *ns* indicates the number of switches.

In the network, Ps_i indicates the power consumption of the *ith* switch, and Bs_i is a binary value, which indicates that the *ith* switch is on or off. Also, *nl* indicates the total number of links in the switches, and pl_j is a binary value, which indicates that the *jth* link is active or not. Thus, when the *jth* switch is in use, pl_j is 1; otherwise, it is zero. Because, for each activated switch link, more power is applied, the second part of this equation checks all of the activated links in topology.

5.4 Pseudocode of the Proposed Algorithm

In this solution, a certain number of employed bees are applied to achieve an optimal solution. Also, the number of employed bees is considered to be the host number, and onlooker bee number can be calculated by equation 14:

$$N_{onlooker}=N_{\text{employed}} * P_i+1 \qquad (14)$$

In this equation, each onlooker bee represents a solution where indicates the placement of each VM on a PM. Equation 15 provides a more elaborate version of equation 6, in which F_1, F_2, and F_3 parameters are replaced by equations 7, 11, and 13, respectively. Also, equation 16 indicates a more elaborate version of equation 1, where the *F* parameter is

replaced by equation 15. On the other hand, equations 15 and 16 can better indicate which parameters are considered in our scheme for VM placement. Also, the probability value for each solution can be calculated by equation 17. Generally, there is a direct relation between the fitness value and this probability value. The result of this equation

$$F= \alpha\ (\textstyle\sum_{i=1}^{n} pi) + \beta(\sum_{i=0}^{n-1}\sum_{j=i+1}^{n-1} \quad Cost_{ij}) + \pi(\sum_{i}^{ns} Ps_i * Bs_i + \sum_{j}^{nl} pl_j * Bl_j) \qquad (15)$$

$$Fitness= \frac{1}{1+\alpha\ (\sum_{i=1}^{n} pi)+\beta(\sum_{i=0}^{n-1}\sum_{j=i+1}^{n-1} \quad Cost_{ij})+\pi(\sum_{i}^{ns} Ps_i*Bs_i+\sum_{j}^{nl} pl_j*Bl_j)} \qquad (16)$$

$$P= \frac{\alpha\ (\sum_{i=1}^{n} pi)+\beta(\sum_{i=0}^{n-1}\sum_{j=i+1}^{n-1} \quad Cost_{ij})+\pi(\sum_{i}^{ns} Ps_i*Bs_i+\sum_{j}^{nl} pl_j*Bl_j)}{1+\sum_{n=1}^{SN}\ \alpha\ (\sum_{i=1}^{n} pi)+\beta(\sum_{i=0}^{n-1}\sum_{j=i+1}^{n-1} \quad Cost_{ij})+\pi(\sum_{i}^{ns} Ps_i*Bs_i+\sum_{j}^{nl} pl_j*Bl_j)} \qquad (17)$$

for each solution is compared with a random value between this [0,1]. If the probability value is larger, then a solution is chosen in that iteration.

Figure 7 provides the pseudocode of the proposed multi-objective ABC algorithm. This algorithm works as follows: In the first loop (Line 2), the ABC parameters and list of the physical machines and VMs are initialized. Then for each bee, a new solution is produced, and the cost and the fitness are calculated by Equations 15 and 16. In the second loop, the probability value for each employed bee is calculated by Equation 17, and based on this value, and the proposed algorithm makes a decision about employed bees to search their neighborhood or abandon them. This algorithm generates a random value each time and compares it with the probability values. When for each employed bee, the probability value is less than the random value, the solution will be changed completely and replaced with the new one by scout bee, and if it is more than the random value, onlooker bees will be specified based on line 13. The neighborhood of the accepted bees will be searched by the onlooker bees to do this. Each onlooker bee changes the place of some VMs randomly and calculates the cost and fitness. If the cost of the onlooker bee is less than the cost of the employed bee, the algorithm will replace the old, employed bee with the new onlooker bee. The best answer of each iteration will be saved in the last lines, and after finishing the second loop, this algorithm will search all the best answers of each step of the second loop to choose the best answer, which has the lowest cost. Table 2 indicates the variables applied in equations (15),(16), and (17).

Table 2: Description of Variables

Variable	Definition
α	Coefficient of f1
β	Coefficient of f2
π	Coefficient of f3
pi	Power consumption of i-th PM
n	Number of PMs
$Cost_{ij}$	The cost of transferred data between PM_i and PM_j
Ps_i	Power consumption of i-th switch
Bs_i	binary decision variable If Bs_i==0 switch is disabled If Bs_i==1 switch is enabled
l_i	Power consumption of i-th link
Bl_i	binary decision variable

	If Bl_i==0 link is disabled If Bl_i==1 link is enabled
ns	Total number of switches
nl	Total number of links
F	The cost function for each bee
Fitness	Fitness function for each bee
P	The calculated probability for each bee

```
Algorithm 2: Artificial Bee Colony-based VM Placement
Input: Load VM-list, Host-list,VMrelatiom [][],SN= hostlist.size, Max-iteration
Output :Best[]//a proposed solution
Generate initial population employed bee =1….SN
For i=1 to SN
      Do
            Generate randomly solution for employed bee [i]
      While(solution is safe for employed bee [i])
      Calculate cost for employed bee [i] from equation (16)
      Evaluate fit_i for employed bee [i]from equation (17)
 End For
 For j=1 to Max-iteration
     For k=1 to SN
        Rand=new random()//generate a random value betweem 0 and 1
        calculate the probability value p_j for the solutions in this iteration from equation(2)
               If ( p_j >Rand)//onlooker bee phase
                  Onlooker_number=[fit_i*SN]+1
                  Onlooker bee[]=new bee[Onlooker_number]
                  For l=1 to Onlooker_number {
                      Produce new solution from employed bee [k] using levyflight for each onlooker bee
                      Do
                           Generate randomly solution for employed bee [i]
                      While(the solution is safe for employed bee [i])
                          If (cost onlookerbee<cost employed bee )
                             employed bee =onlooker bee
                             Calculate the value f_i
                          End If
                    End For
               Else If( p_j <Rand)//scout bee phase
                             Bee scout bee;
                             Generate randomly solution for scout bee
                              Do
                                    Generate randomly solution for scout bee
                              While(the solution is safe for scout bee )
                              Calculate cost for scout bee from equation (15)
                              Evaluate fit_i for scout bee from equation (16)
                              employed bee [i]=scout bee;
               End If
      End For
      BestsoFar[j]=FindLowestCost( employed bee [])
End For
Best=FindLowestCost(Bestsofar[])
```

Figure 7: Pseudocode of the Artificial Bee Colony-based VM placement solution

6. Performance Evaluation

To evaluate the proposed VM placement scheme, extensive simulations are conducted in the CloudSim toolkit, which is an extendable simulation framework to enable simulation and modeling of the cloud computing infrastructures and services. CloudSim toolkit offers features to support simulating the framework of large scale cloud computing, which includes the data centers on a single physical computing node and a self-contained platform for modeling the data centers, scheduling, allocations policies, and service brokers. Moreover, CloudSim provides a virtualization engine for managing multiple, co-hosted virtualized services, flexibility to change between space- and time-shared processing cores allocation to the services virtualized [49].

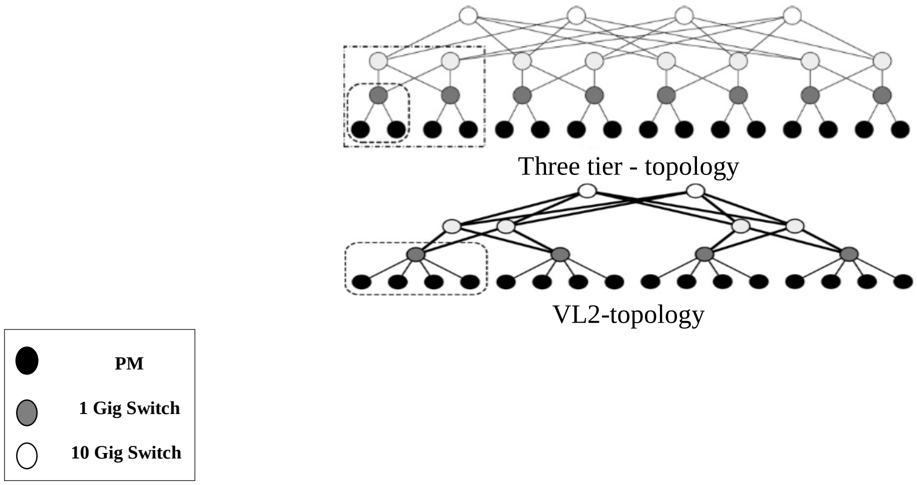

Figure 8-Network Topology

Table 3: Number of Network elements

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

In our solution, the consumption of the network switches' power is considered in the placement decisions. For this purpose, it is necessary to consider some topology for the data center's network. As indicated in Figure 8, it is assumed that the data centers use Three-Tier or VL2 [50] network topologies, and the switches' power consumption is computed in both cases. In our simulation, the number of elements in each topology is indicated in Table 3. In these scenarios, the power consumption of network components is considered as follow:

- 1 Gigabit Switch=200 Watt
- 10 Gigabit Switch=30 Watt
- 1 Gigabit Link=6 Watt
- 10 Gigabit Link=0.4 Watt

This section presents the simulation results achieved by considering VL2 and Three-Tiered topologies for the data center's network. In these simulations, our proposed VM placement solution and the CloudSim's simple VM Allocation Policy, and the Greedy VM placement algorithms are compared with each other. In these comparisons, mainly the following factors are evaluated in the simulation scenarios:

- Physical machines power consumption
- Network switches power consumption
- Number of the activated links in network switches
- Number of the activated switches in the data center network
- Percentage of CPU rate.

In all of these simulations, 16 physical machines have been considered in the data center's network. Also, two kinds of physical machines are used, 8 of which have the following characteristics:

- CPU= 117160 MIPS
- P_{max}= 210Watt and P_{idle} =P_{max} /2Watt

The specifications of the other 8 physical machines are as follows:

- CPU=97125 MIPS
- P_{max} =129watt and P_{idle}= P_{max}/2watt

Moreover, each simulation is conducted for 16VMs, 32VMs, 48VMs, and 64VMs, and in each scenario, two types of VMs with the following characteristics are considered:

- CPU= 10000 MIPS
- CPU= 8000 MIPS

Here, the second VM type corresponds to an edge device and the former as a cloud node. Figure 9 and Figure 10 present the comparison of physical machines' power consumption (watt) in the center of the data. As outlined before, the power consumption of each PM directly depends on its CPU load, and this load increases when more VMs are applied in the network. Thus, as shown in these figures, the power that the physical machines consume goes up as the number of VMs goes up. As indicated in these figures, based on the different values for coefficients of the cost function, the proposed scheme can effectively mitigate the power consumption of the physical machines during the time of the simulation. In our approach, the server power consumption is reduced by 3.5%.

Figure 11 and Figure 12 indicate the percentage of the CPU rate in the physical machines after all VMs are placed on them. As indicated in these figures, with the increase of VMs number, the percentage of CPU rate in the physical machines increases. In these figures, the proposed solution is simulated with four different sets of values for coefficients of the cost function. The results of all these executions indicate that the proposed VM placement solution is better than the other two VM placement methods.

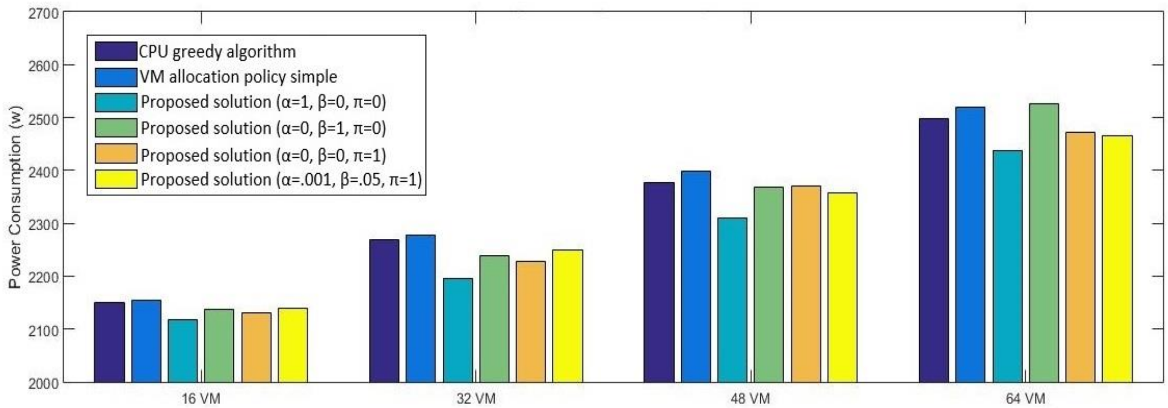

Figure 9: Comparison of the physical machines power consumption (W) in VL2 topology

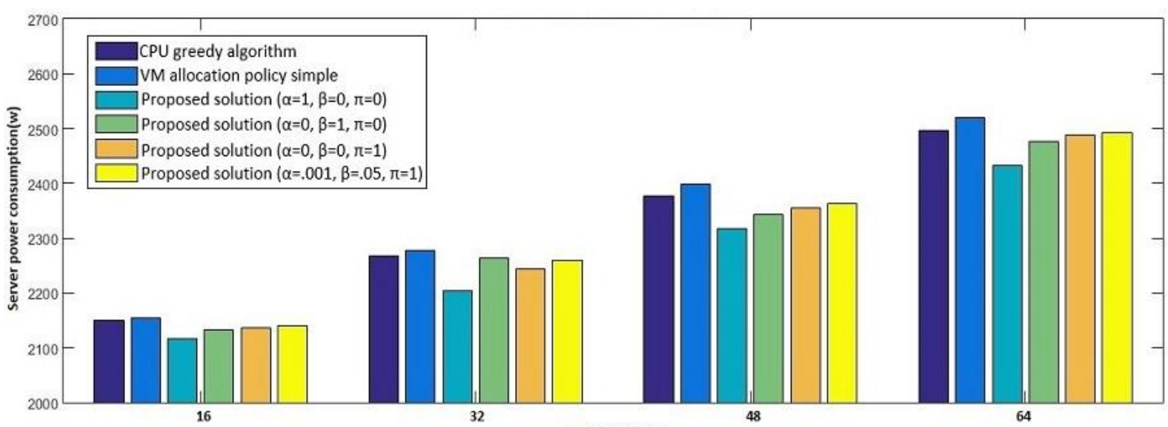

Figure 10: Comparison of the physical machines power consumption (Watt) in Three-Tier topology

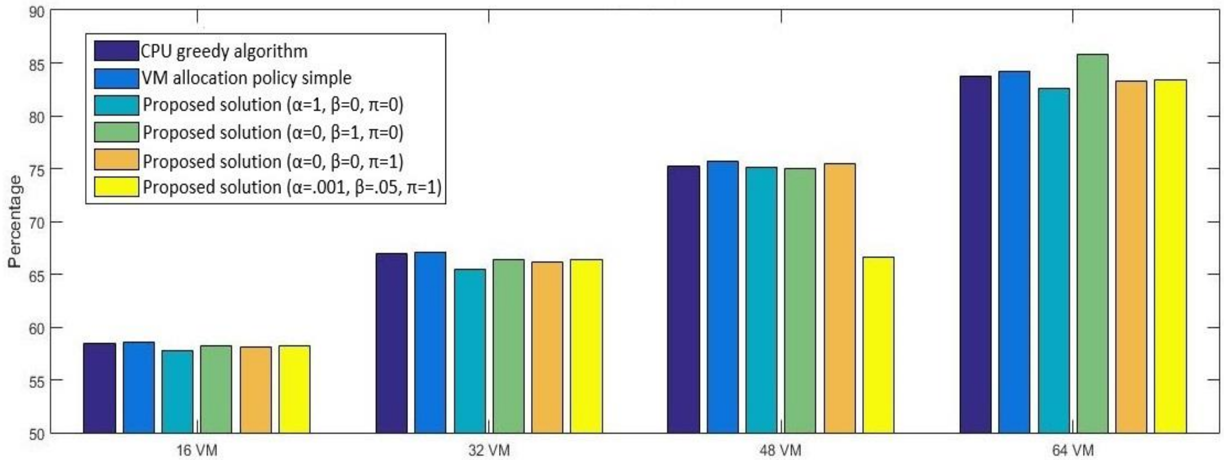

Figure 11: Comparison of the average of the maximum CPU rate in VL2 topology

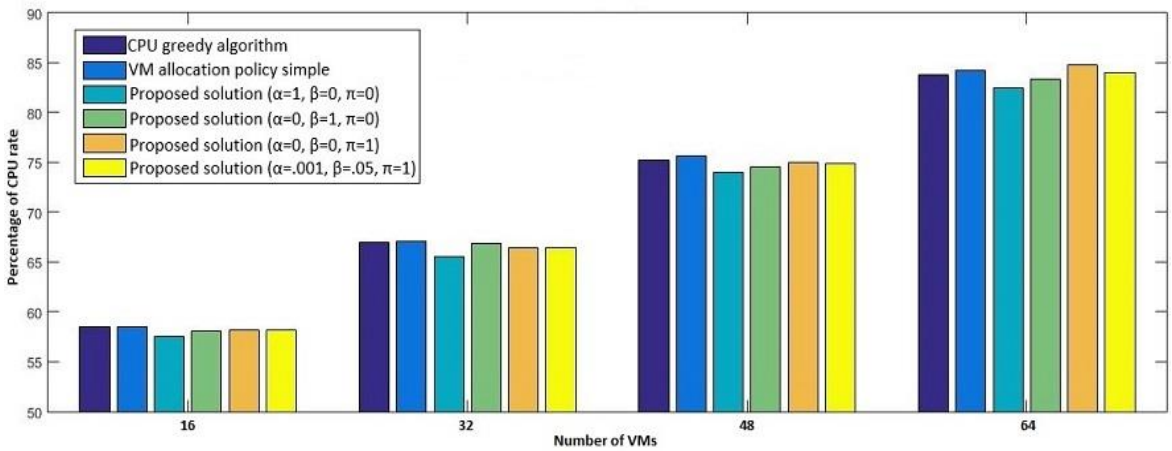

Figure 12: Comparison of the average CPU rate in the Three-Tier topology

The number of activated 1 gigabit and 10-gigabit switches in each scenario for the evaluated algorithms is shown in Figure 13 and Figure 14. To be noted, the number of activated switches depends on the amount of the data transferred between physical machines and the VM placement algorithm, not the number of the applied VMs. In these figures, the proposed solution results of the simulation are indicated in four different cases where different values are applied for the cost function coefficients. As shown in these figures, in most scenarios, the proposed ABC-based VM placement algorithm presents better results than the other two VM placement methods.

Figure 15 and Figure 16 indicate the number of the activated links in the 1 gigabit and 10-gigabit network switches applied in four cases with different numbers of VMs. It can be concluded from these figures that as the number of VMs and their transmitted data increase, the number of active links in the network switches increases accordingly. As shown in these figures, our scheme can greatly reduce the number of active links better than the other two schemes.

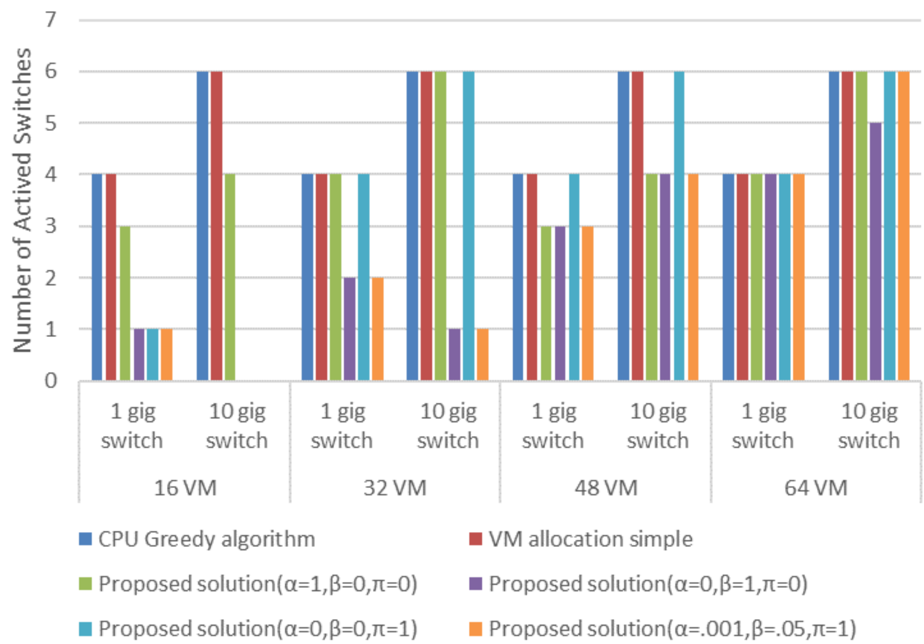

Figure 13: Number of the activated switches in the VL2 topology

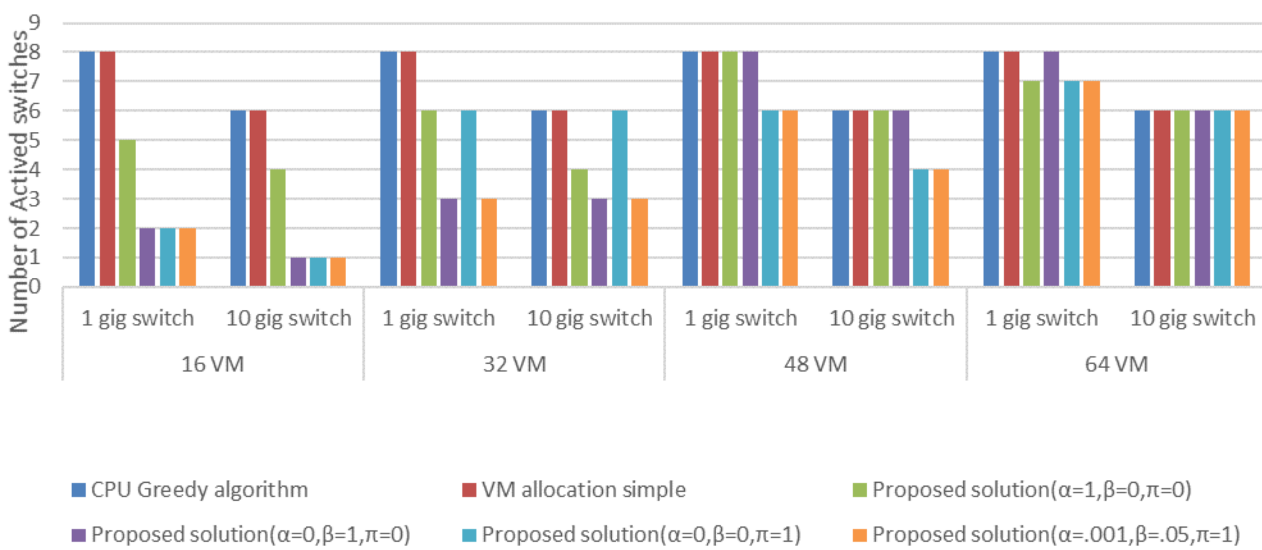

Figure 14: Number of the activated switches in the Three-Tier topology

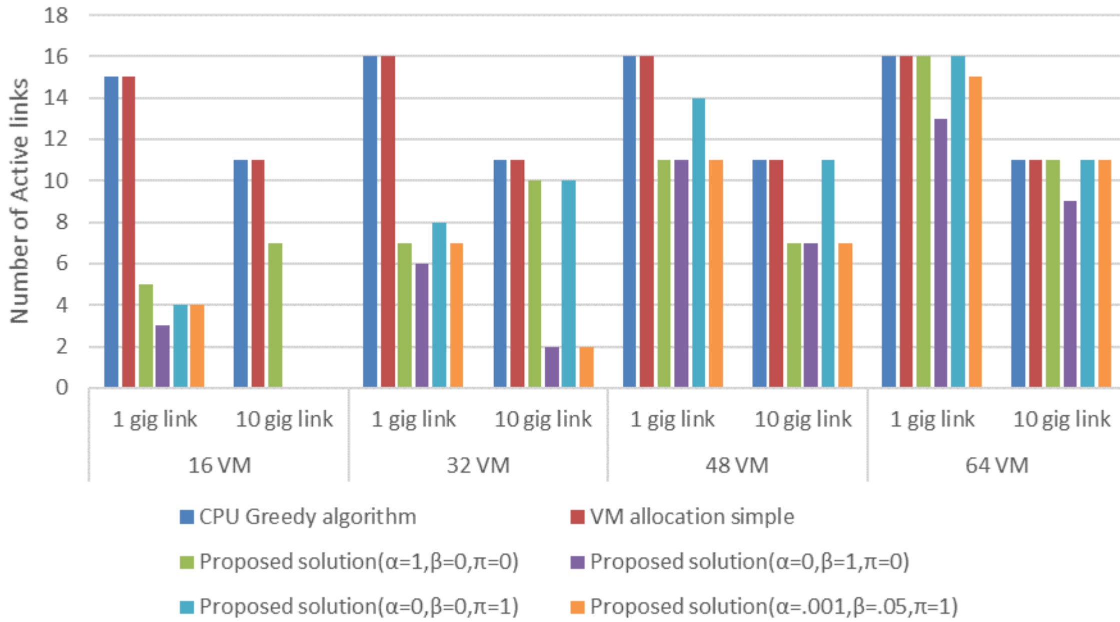

Figure 15: Number of the active links in the VL2 topology

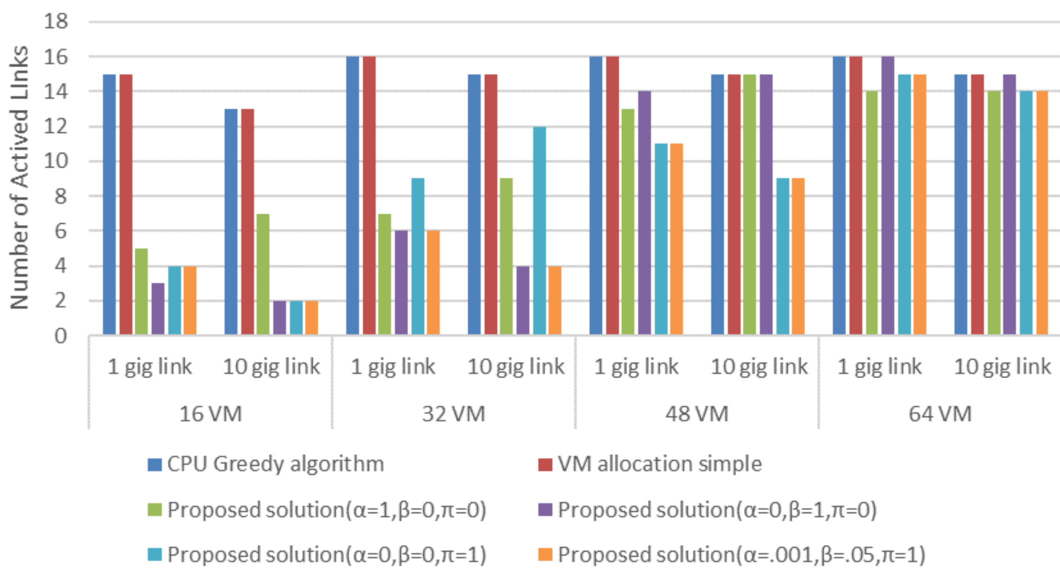

Figure 16: Number of the active links in the Three-Tier topology

Figure 17 and Figure 18 present the comparison of the consumption of the power of the network components in the 4 cases where different numbers of the VMs are placed on the 16 PMs. Again, the same as the previous simulation results, the results of the proposed solution are provided in 4 different cases where different values are applied for the coefficients of the cost function. As outlined before, these coefficients tune the effect of the network traffic and power consumption of the PMs and network switches. As indicated in this figure, except in the last scenario, our proposed solution consumes multiple times less power for all network switches. These simulations are conducted in 4 cases where 16 VMs, 32 VMs, 48 VMs, and 64 VMs are considered for placement on the 16 physical machines in the ECDC. The amount of energy consumption of the network has a direct relation to the number of VMs. For 16 and 32 VMs in both figures 17 and 18, the energy consumption is reduced by nearly 70%. In the case of 48 VMs, our proposed approach consumes 20% less energy compared to VM allocation simple and CPU Greedy algorithm.

In the last scenario, since the number of VMs is high, most of the network elements are active, and the results for all approaches are approximately the same.

Figure 19 and Figure 20 indicate the total time period when network switches are activated in each scenario. As outlined in the previous sections, network switches provide a means for the required interactions between VMs placed on different physical machines. Again, the results of these simulations are shown for 16VMs, 32VMs, 48VMs, and 64VMs, and our solution, in most cases, provides better results and applies network switches for less time. In the case of 16 VMs, the total activation time of network elements, our approach reduced by 4 and 2 seconds in comparison with CPU greedy and VM allocation simple for VL2 and three-tier, respectively. This rate in 32 and 48 consumes 50% less time in our solution. Finally, for 64 VMs, our approach saves more than 20 seconds.

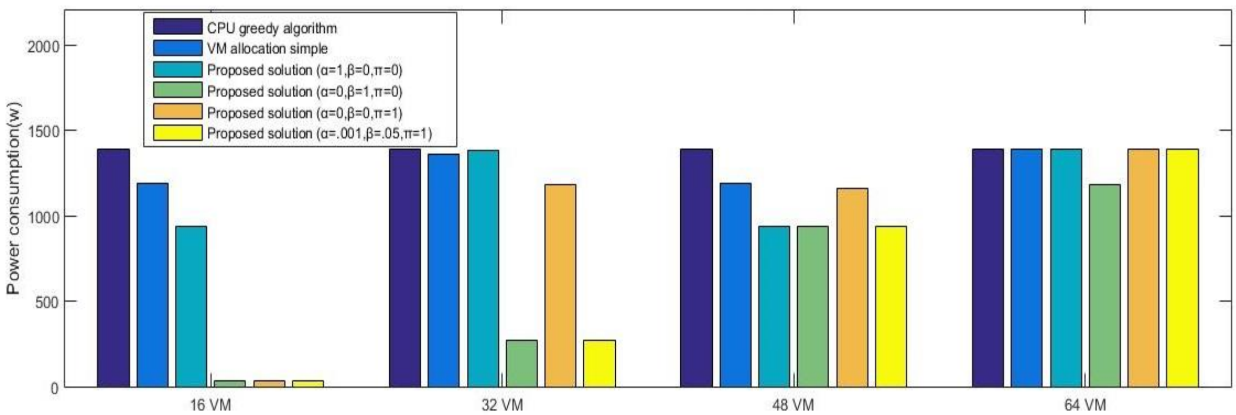

Figure 17: Network power consumption (Watt) in the VL2 topology

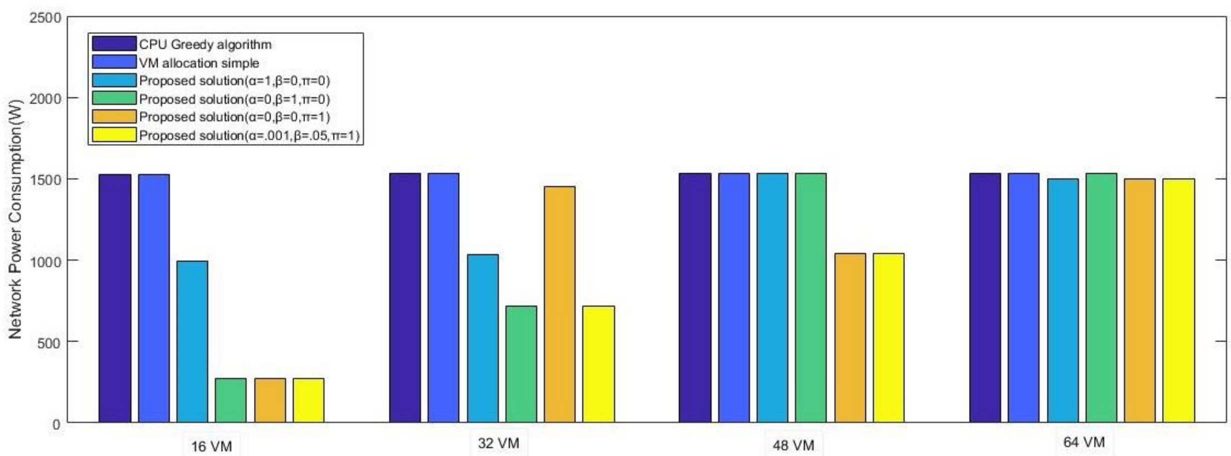

Figure 18: Network power consumption (Watt) in the Three-Tier topology

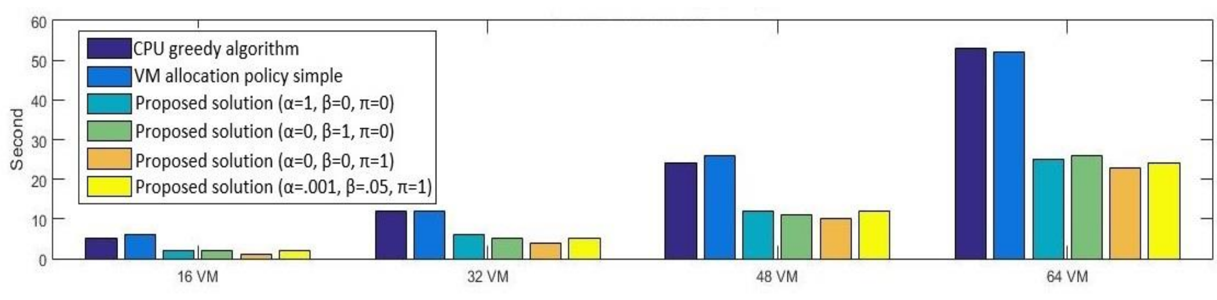

Figure 19: Network switches usage in the VL2 topology

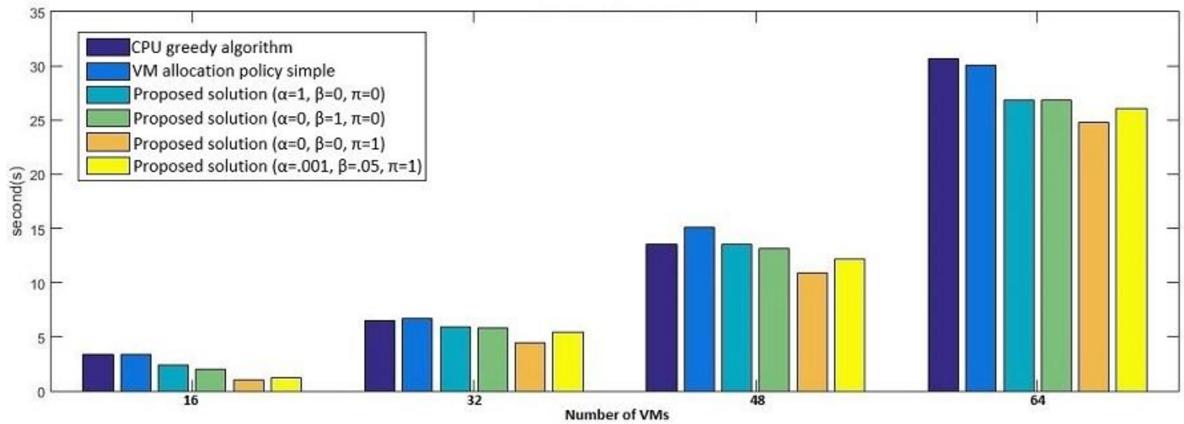

Figure 20: Network switches usage in the Three-Tier topology

7. Conclusions and Future Scope

Cloud data centers consist of various power-consuming devices such as physical machines and network switches. But, the high power consumption of these devices increases the data centers' maintenance costs and also may cause serious problems because of the excessive heat. Thus reducing the power consumption of the ECDCs is of high importance in Green cloud computing. To solve these problems, this study offers an Artificial Bee Colony (ABC)-based VM placement scheme for cloud computing data centers called TRACTOR, which tries to lower the consumption of the power of the various data center's elements such as physical machines and network switches. In addition, this VM placement solution considers the network traffic between VMs and tries to reduce the network traffic by placing interacting VMs close to each other efficiently. This reduces the power consumption of switches and their links and increases the throughput of the data centers. Finally, the proposed ABC-based VM placement solution is compared with the Greedy VM placement and VM placement simple algorithms in the Three-Tier and VL2 network topologies. Experimental results indicate that TRACTOR is able to reduce power energy consumption by 3.5% whilst decreasing Network traffic and power by 15% and 30%, respectively, without affecting other Quality of Service parameters. The simulation results indicate the effectiveness of our scheme in mitigating the consumption of the physical machines' power, network switches, and the time period when switches are activated in both network topologies.

7.1 Future Research Directions

Although the proposed approach is a comprehensive VM placement and concerns different objectives, it can be enhanced in a larger scope under the following aspects:

1. *Thermal-aware Management:* In the future, the TRACTOR can be extended by incorporating temperature as an optimization parameter along with energy consumption [52], which further helps the temperature of the ECDC to be maintained.
2. *Security:* Considering the ever-increasing security attacks such as a sprawling attack, Economic Denial of Sustainability (EDoS) attacks, and various other DoS attacks on virtualization, in future researches and studies, we will take into account the security issues in conducting VM placement. Security of the data can be optimized using Blockchain based framework called FogBus [37]. This technology ensures the reliability of the systems. In the future, this technology can be applied for handling data in edge computing.
3. *Container based Deployment:* In the future, virtualization technology, which is used in our approach, can be substituted with containers. Container is a lightweight technology that is used to virtualize applications [51]. Using containers can enhance CPU performance and improve energy consumption.
4. *Internet of Things (IoT):* IoT is the network of objects, devices, buildings, and other cases, including electronic sensors, network connectivity, and this brings the ability to gather and interchanging data between objects. Since IoT and technology applications are developing, cloud and edge would be an effective infrastructure to manage data effectively [51]. Cloud computing can fill the gap of IoT, such as limited storage and applications over the Internet. Also, the main problem of cloud computing, limited scope can be solved through IoT.

5. *Artificial intelligence (AI):* AI can be used to make fog nodes aware of the workload environment and continuously adapt to provide better QoS. In addition, it reduces power consumption and total infrastructure cost. This method provides optimum VM placement and migration effectively [54].
6. *Serverless Edge Computing:* Further, TRACTOR can be implemented using Serverless computing or Function as a Service (FaaS) to improve salability and reduce cost because there is no need of server configuration while deploying an application [55]. With the help of Serverless computing, application can be easy to scale.

Data Availability Statement

A significant amount of data is presented in this article. The remaining data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.

References

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," *National Institute of Standards and Technology,* vol. 53, no. 6, p. 50, 2009.

[2] A. Abdelaziz, M. Elhoseny, A. S. Salama, A. M. Riad, and A. E. Hassanien, "Intelligent algorithms for optimal selection of virtual machine in cloud environment, towards enhance healthcare services," in *International Conference on Advanced Intelligent Systems and Informatics*, 2017: Springer, pp. 289-298.

[3] M. Masdari, S. S. Nabavi, and V. Ahmadi, "An overview of virtual machine placement schemes in cloud computing," *Journal of Network and Computer Applications,* 2016.

[4] A. J. I. C. C. Eivy, "Be wary of the economics of" Serverless" Cloud Computing," vol. 4, no. 2, pp. 6-12, 2017.

[5] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: state-of-the-art and research challenges," *Journal of internet services and applications,* vol. 1, no. 1, pp. 7-18, 2010.

[6] J. A. González-Martínez, M. L. Bote-Lorenzo, E. Gómez-Sánchez, and R. Cano-Parra, "Cloud computing and education: A state-of-the-art survey," *Computers & Education,* vol. 80, pp. 132-151, 2015.

[7] M. K. Gupta and T. J. T. J. o. S. Amgoth, "Resource-aware virtual machine placement algorithm for IaaS cloud," vol. 74, no. 1, pp. 122-140, 2018.

[8] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," *Future Generation computer systems,* vol. 28, no. 3, pp. 583-592, 2012.

[9] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of network and computer applications,* vol. 34, no. 1, pp. 1-11, 2011.

[10] K. Fine and E. Kruglick, "Virtual machine placement," ed: Google Patents, 2018.

[11] W. Chen, D. Wang, and K. J. I. T. o. S. C. Li, "Multi-user multi-task computation offloading in green mobile edge cloud computing," 2018.

[12] Y. Gao, H. Guan, Z. Qi, Y. Hou, and L. Liu, "A multi-objective ant colony system algorithm for virtual machine placement in cloud computing," *Journal of Computer and System Sciences,* vol. 79, no. 8, pp. 1230-1242, 2013.

[13] X. Fu and C. J. I. T. o. C. C. Zhou, "Predicted affinity based virtual machine placement in cloud computing environments," 2017.

[14] G. Portaluri, D. Adami, A. Gabbrielli, S. Giordano, M. J. I. T. o. G. C. Pagano, and Networking, "Power consumption-aware virtual machine placement in cloud data center," vol. 1, no. 4, pp. 541-550, 2017.

[15] M. Xu, W. Tian, R. J. C. Buyya, C. Practice, and Experience, "A survey on load balancing algorithms for virtual machines placement in cloud computing," vol. 29, no. 12, p. e4123, 2017.

[16] S. Goel and T. Arjariya, "A Review on Different Energy Efficient VM Placement Approaches with their Anomalies in Cloud computing Environment," 2015.

[17] M. Masdari, S. ValiKardan, Z. Shahi, and S. I. Azar, "Towards workflow scheduling in cloud computing: A comprehensive analysis," *Journal of Network and Computer Applications*, 2016.

[18] S.-H. Wang, P. P.-W. Huang, C. H.-P. Wen, and L.-C. Wang, "EQVMP: Energy-efficient and QoS-aware virtual machine placement for software defined datacenter networks," in *Information Networking (ICOIN), 2014 International Conference on*, 2014: IEEE, pp. 220-225.

[19] N. M. Calcavecchia, O. Biran, E. Hadad, and Y. Moatti, "VM placement strategies for cloud scenarios," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, 2012: IEEE, pp. 852-859.

[20] X. Ye, Y. Yin, and L. J. I. a. Lan, "Energy-efficient many-objective virtual machine placement optimization in a cloud computing environment," vol. 5, pp. 16006-16020, 2017.

[21] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," *Applied soft computing*, vol. 8, no. 1, pp. 687-697, 2008.
[22] S. S. Gill *et al.*, "Holistic resource management for sustainable and reliable cloud computing: An innovative solution to global challenge," *Journal of Systems and Software*, vol. 155, pp. 104-129, 2019.
[23] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, 2005.
[24] B. Basturk and D. Karaboga, "An artificial bee colony (ABC) algorithm for numeric function optimization," in *IEEE swarm intelligence symposium*, 2006, vol. 8, no. 1, pp. 687-697.
[25] W. Huang, X. Li, and Z. Qian, "An energy efficient virtual machine placement algorithm with balanced resource utilization," in *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013 Seventh International Conference on*, 2013: IEEE, pp. 313-319.
[26] Y. Ting, L. Young Choon, and A. Y. Zomaya, "Energy-Efficient Data Center Networks Planning with Virtual Machine Placement and Traffic Configuration," in *Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on*, 15-18 Dec. 2014 2014, pp. 284-291, doi: 10.1109/CloudCom.2014.135.
[27] X. Li, Z. Qian, S. Lu, and J. Wu, "Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in a data center," *Mathematical and Computer Modelling*, vol. 58, no. 5, pp. 1222-1235, 2013.
[28] H. Zhao *et al.*, "Power-aware and performance-guaranteed virtual machine placement in the cloud," vol. 29, no. 6, pp. 1385-1400, 2018.
[29] J. T. Piao and J. Yan, "A network-aware virtual machine placement and migration approach in cloud computing," in *Grid and Cooperative Computing (GCC), 2010 9th International Conference on*, 2010: IEEE, pp. 87-92.
[30] W. Fang, X. Liang, S. Li, L. Chiaraviglio, and N. Xiong, "VMPlanner: Optimizing virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers," *Computer Networks*, vol. 57, no. 1, pp. 179-196, 2013.
[31] W. Chen, X. Qiao, J. Wei, and T. Huang, "A profit-aware virtual machine deployment optimization framework for cloud platform providers," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, 2012: IEEE, pp. 17-24.
[32] A. Satpathy, S. K. Addya, A. K. Turuk, B. Majhi, G. J. C. Sahoo, and E. Engineering, "Crow search based virtual machine placement strategy in cloud data centers with live migration," vol. 69, pp. 334-350, 2018.
[33] J. Luo, W. Song, and L. J. I. A. Yin, "Reliable virtual machine placement based on multi-objective optimization with traffic-aware algorithm in industrial cloud," vol. 6, pp. 23043-23052, 2018.
[34] S. Gharehpasha, M. Masdari, and A. Jafarian, "Power efficient virtual machine placement in cloud data centers with a discrete and chaotic hybrid optimization algorithm," *Cluster Computing*, pp. 1-23, 2020.
[35] S. Tuli, S. Ilager, K. Ramamohanarao, and R. Buyya, "Dynamic scheduling for stochastic edge-cloud computing environments using a3c learning and residual recurrent neural networks," *IEEE Transactions on Mobile Computing*, 2020.
[36] S. Azizi, M. Shojafar, J. Abawajy, and R. Buyya, "GRVMP: A Greedy Randomized Algorithm for Virtual Machine Placement in Cloud Data Centers," *IEEE Systems Journal*, 2020.
[37] S. Tuli, R. Mahmud, S. Tuli, and R. Buyya, "Fogbus: A blockchain-based lightweight framework for edge and fog computing," *Journal of Systems and Software*, vol. 154, pp. 22-36, 2019.
[38] Z. Li, Y. Li, T. Yuan, S. Chen, and S. J. A. I. Jiang, "Chemical reaction optimization for virtual machine placement in cloud computing," vol. 49, no. 1, pp. 220-232, 2019.
[39] M. Mohammadhosseini, A. T. Haghighat, and E. J. T. J. o. S. Mahdipour, "An efficient energy-aware method for virtual machine placement in cloud data centers using the cultural algorithm," pp. 1-30, 2019.
[40] M. J. Usman *et al.*, "Energy-efficient Virtual Machine Allocation Technique Using Flower Pollination Algorithm in Cloud Datacenter: A Panacea to Green Computing," vol. 16, no. 2, pp. 354-366, 2019.
[41] X.-F. Liu, Z.-H. Zhan, and J. J. E. Zhang, "An energy aware unified ant colony system for dynamic virtual machine placement in cloud computing," vol. 10, no. 5, p. 609, 2017.
[42] F. López-Pires and B. J. J. o. G. C. Barán, "Many-objective virtual machine placement," vol. 15, no. 2, pp. 161-176, 2017.
[43] M. Ghobaei-Arani, M. Shamsi, A. A. J. J. o. E. Rahmanian, and T. A. Intelligence, "An efficient approach for improving virtual machine placement in cloud computing environment," vol. 29, no. 6, pp. 1149-1171, 2017.

[44] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," *Applied mathematics and computation,* vol. 214, no. 1, pp. 108-132, 2009.
[45] D. Karaboga and B. Basturk, "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems," in *International fuzzy systems association world congress*, 2007: Springer, pp. 789-798.
[46] W. Gao and S. Liu, "Improved artificial bee colony algorithm for global optimization," *Information Processing Letters,* vol. 111, no. 17, pp. 871-882, 2011.
[47] D. Karaboga, S. Okdem, and C. Ozturk, "Cluster based wireless sensor network routing using artificial bee colony algorithm," *Wireless Networks,* vol. 18, no. 7, pp. 847-860, 2012.
[48] A. Rajasekhar, A. Abraham, and M. Pant, "Levy mutated artificial bee colony algorithm for global optimization," in *2011 IEEE International Conference on Systems, Man, and Cybernetics*, 2011: IEEE, pp. 655-662.
[49] R. N. Calheiros, R. Ranjan, C. A. De Rose, and R. Buyya, "Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services," *arXiv preprint arXiv:0903.2525,* 2009.
[50] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, pp. 63-74, 2008.
[51] S. S. Gill *et al.*, "Transformative effects of IoT, Blockchain and Artificial Intelligence on cloud computing: Evolution, vision, trends and open challenges," *Internet of Things,* vol. 8, p. 100118, 2019.
[52] S. S. Gill *et al.*, "ThermoSim: Deep learning based framework for modeling and simulation of thermal-aware resource management for cloud computing environments," *Journal of Systems and Software,* p. 110596, 2020.
[53] W. Z. Khan *et al.,* Edge computing: A survey. Future Generation Computer Systems, 97, 219-235, 2019.
[54] Q. Zhou *et al.*, Energy efficient algorithms based on VM consolidation for cloud computing: comparisons and evaluations. Proceedings of the 20th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID 2020); 2020:1-10
[55] M. S. Aslanpour *et al*, Serverless Edge Computing: Vision and Challenges, Proceedings of the 19th Australasian Symposium on Parallel and Distributed Computing (AusPDC 2021), February 1–5, 2021, Dunedin, New Zealand.