# A Green-aware Strategy for Virtual Machine Placement in Cloud Datacenters

**H. Nasrolahi Matak[1], H. Motameni[2*], B. Barzegar[3], E. Akbari[4], H. Shirgahi[5]**

**Abstract**–This paper presents a method for optimizing the dual-target virtual machine provisioning problem, which is a challenge in cloud data centers. In the cloud environment, it is important to balance the interests of service providers and customers. From the producers' viewpoint, optimizing energy consumption and reducing costs are essential. From the users' point of view, it is desirable to achieve an adequate level of quality of service, and network latency is one of the factors that contribute to its reduction. Therefore, optimizing bandwidth usage to reduce network delay is the second important objective considered in this study. To solve this problem, a two-objective method based on a genetic algorithm is presented, which provides near-optimal results in an acceptable time. The evaluations show the superiority of the proposed algorithm in terms of total energy consumption and total traffic in the network compared with methods based on a genetic algorithm, ant colony, greedy FFD algorithm, and randomized deployment method.

**Keywords:** Cloud computing, Virtual machine placement, Multiobjective optimization, Meta-heuristic algorithms

## 1. Introduction

Cloud computing has emerged because of the irregular growth of computers, and telecommunication systems have started to offer various services to users over the internet. Users should connect to data centers to use on-demand services. The computing needs of users have significantly increased the energy consumption of data centers that becomes a challenge for cloud computing [1]. Cloud computing is now recognized worldwide as an integral mechanism of information technology. Considering the variety of services such as infrastructure, platform and software as a service, cloud computing plays an undeniable role in hosting and providing services on the Internet. The benefits of using cloud facilities for individuals and organizations include reliability, quality of service, and strength [2]. The service producer and consumer, which play a central role in the interactions of the cloud environment, have different interests in this area. From the customer's point of view, it is desirable to receive services of the highest quality, in accordance with the service level agreement, with as few violations as possible and a minimum payment for the use of each service. From the producer's perspective, it is desirable to reduce energy consumption, reduce waste of resources, reduce costs, and comply with the provisions of the Service Level Agreement to gain and maintain customer confidence [3].

The concept of virtualization is based on the fact that different user programs (virtual machines (VMs)) can be executed on some servers (Physical Machines (PMs)). This process is called virtual machine provisioning and belongs to the category of NP-hard problems in terms of time complexity [4]. Despite the benefits of virtualization, real-world experience shows that a strict reduction in energy consumption and resource waste can pose a threat to quality of service requirements (such as throughput, response time and network latency), that are written and documented in users' service level agreements. Conversely, communication links, switching between physical machines, and the collection of data sent in different layers of the network are responsible for more than 30% of the total energy consumption of data centers[5]. Moreover, more than 70% of the data traffic in a data center is caused by data exchange between virtual machines[6]. As a rule, in large data centers communication between virtual machines not only significantly increases energy consumption, but can also prove to be a serious bottleneck for quality of service requirements (such as response time and delay) [7-9]. This omission can lead to an increased possibility of a

1,2,4 Department of Computer Engineering, Sari Branch, Islamic Azad University, Sari, Iran
3Department of Computer Engineering, Babol Branch, Islamic Azad University, Babol, Iran
5Department of Computer Engineering, Jouybar Branch, Islamic Azad University, Jouybar, Iran
*Corresponding Author Institutional Email: motameni@iausari.ac.ir
*Received: 2024.08.22; Accepted: 2024.09.29*

breach of the service level agreement, which robs the user of confidence and satisfaction and may result in the application of various types of fines by the user, depending on the nature, extent, and severity of the breach [10]. A modern and promising method to address this challenge is to place virtual machines with a lot of data exchange at the smallest possible physical distance from each other. Therefore, when using virtual machines, a trade-off must be found between reducing the energy consumption of computer and network communication devices and reducing the data exchange between virtual machines over long distances that saturate the bandwidth and network topology[11]. Furthermore, consideration should be given to the possible bandwidth depending on the network topology [12].

In [13] presented architectural principles for managing energy consumption in the cloud, as well as policies for allocating energy resources and scheduling algorithms that meet the expectations for quality of service and device power consumption. Local administrators send information about resource utilization and virtual machines selected for migration to global administrators. Various methods have been proposed to assign virtual machines to physical nodes. The assignment problem is divided into two parts. The first part refers to the replacement of virtual machines on physical hosts, and the second part refers to the optimization of virtual machine allocation. In [14] presented a new working framework called Green Cloud Computing. In their study, virtual machine management and scheduling is considered as one of the fundamental principles for reducing energy consumption. The author introduced his main method for reducing energy consumption as shutting down physical machines with low utilization and migrating virtual machines to other physical machines. In [15] investigated a physical machine integration algorithm that was periodically deployed to minimize online machines in terms of required online capacity and possible SLA violations. They analyzed different workload profiles and show that intermittent workloads are better suited for dynamic acquisition. The mapping step is performed using a heuristic first-fit method. while reducing the number of online users. To optimize the allocated resources in cloud computing, In [16] presented a scheduling and resource allocation method based on the PSO method, in which multiple queues are considered for resource allocation. Queues are used to manage and schedule tasks that are scarce at the time of resource allocation and are actually placed in the line. Green computing is a trend in computer science that seeks to reduce the energy consumption and carbon footprint of computers in distributed platforms such as clusters, networks, and clouds. Recent studies have estimated that data centers account for approximately 1.5%–2 % of the total energy consumption. This energy demand has sharply increased because of the generalization of Internet services and distributed computing platforms such as clusters, networks, and clouds. Regarding the efficiency of data centers, studies show that approximately 55% of the energy used in a data center is consumed by the computer system and the rest by the support system. For this reason, green cloud computing is essential to make the future growth of cloud computing sustainable. Users also want the services they require to be completed faster and in less time. Therefore, we are trying to find a suitable solution to solve these problems in cloud computing. The goal of this research is to optimize the dual objectives of energy consumption and network traffic load sharing when deploying virtual machines in cloud data centers with tree topologies.

## 2. Instrumentation

### 2-1- Formulation of the proposed method

To construct a static two-objective VMP model for a DC with n VM and m PM, the mathematical model for each of the objectives such as the energy consumption of servers and network switches and the bandwidth consumption, is first formulated along with the corresponding constraints. The comprehensive model of dual-objective optimization is presented. Then, the presented model is coded and implemented using a genetic algorithm. The results of the GA for the local search for optimal answers are fed into the complementary algorithm of the local search, and the improved results resulting from the combination of GA and the complementary algorithm are analyzed and evaluated.

### 2-2- Energy consumption model of the servers

It is obvious that the physical equipment of DCs consists of electrical and electronic parts, all of which consume electricity. Most of the total DC energy consumption is caused by the operation of the PMs, but other equipment such as switches, routers, and cooling devices also consume energy, which is also considered in this research, the energy consumption model of the switch. This section focuses on PM energy consumption in two states: full state (when PM hosts VMs so that all CPU capacity is occupied) and idle mode (when PM is not assigned to any VM). It should be noted that there is a linear relationship between energy consumption and CPU efficiency [74]. Therefore, the $PM_j$ energy consumption can be expressed as Eq. (1)

$$P_j = \begin{cases} \left(P_j^{full} - P_j^{idle}\right) \times U_j^{Cpu} + P_j^{idle} & if\ U_j^{Cpu} > 0 \\ 0 & otherwise \end{cases} \quad (1)$$

Where $P_j^{full}$ and $P_j^{idle}$ represent the average values of energy consumption in the full efficiency state and the idle state $PM_J$, respectively.

The total power consumption (TPC) of servers in DC with m PM can be calculated from Eq. (2):

$$TPC = \sum_{j=1}^{m} \left(\left(P_j^{full} - P_j^{idle}\right) \times U_j^{Cpu} + P_j^{idle}\right).x_j \quad (2)$$

Where $x_j \in \{0,1\}$ a binary decision variable is equal to one if $PM_j$ is on and zero otherwise.

### 2-3-Bandwidth consumption model

The bandwidth consumption model presented in this section can be extended to other types of topologies without losing its generality. The tree model consists of identical switches with four ports and homogeneous communication lines that communicate with each other between the three layers (Core-Aggregation/Aggregation-Access). In the VMP, any VM can be placed on any PM. Therefore, the physical distance between PMs hosting interdependent VMs can be calculated by counting the hops. This concept refers to the number of communication lines that must be crossed to route data from the source PM to the destination PM [1]. On the other hand, different VMs can be independent or dependent on each other in terms of data. For example, suppose that $VM_i$ and $VM_j$ are located on $PM_k$ and $PM_\ell$ respectively, with an average data dependency rate of dv. The best scenario is when $PM_k$ and $PM_\ell$ are the same, meaning k=l. Otherwise, $PM_k$ and $PM_\ell$ are connected to each other in one of the following ways:

- Access switch: 2 hops for data transmission
- Aggregation switch: 4 hops for data transmission
- Core switch: 6 hops for data transmission.

The DC network can therefore be divided into 4 virtual zones, as described in Table ۱. The basis for this segmentation is to monitor the positions of the PMs hosting dependent VMs. Figure 1 shows the coverage areas of each zone.

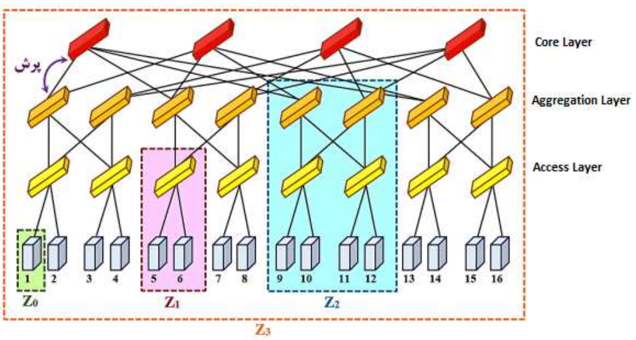


**Figure 1**: Sections covered by virtual regions in a tree-structured grid [1]

**Table 1:** Virtual regions of the DC network [1]

| Zone | $PM_k$ & $PM_\ell$ | Hops |
|---|---|---|
| $Z_0$ | Same () | 0 |
| $Z_1$ | Under the access switch | 2 |
| $Z_2$ | Under the aggregation switch | 4 |
| $Z_3$ | Under the core switch | 6 |

For each connection from $VM_i$ located on $PM_k$ to $VM_j$ located on $PM_\ell$, a connection is defined as $C_{k,\ell} \in Z_q$. The volume of traffic between $VM_i$ and $VM_j$ and generally between any pair of VMs can be extracted from the TPM. An example of a TPM is shown in equation (3).

$$TPM = \begin{array}{c} \\ VM_1 \\ VM_2 \\ \vdots \\ VM_n \end{array} \begin{array}{c} \begin{array}{cccc} VM_1 & VM_2 & \cdots & VM_n \end{array} \\ \begin{bmatrix} 0 & 250 & \cdots & 320 \\ 125 & 0 & \cdots & 260 \\ \vdots & \vdots & \ddots & \vdots \\ 400 & 112 & \cdots & 0 \end{bmatrix} \end{array} \quad (3)$$

In the matrix, all data are given in megabytes per second (MBps). TPM information can be obtained in various ways, for example, by checking the DC profile or by extracting the behavior of VMs using data mining techniques. Since this study refers to static $VM_P$, a specific TPM is used for the calculations, whose elements are all extracted from the history of data exchanged between VMs and are known in advance. According to the previous definitions, the DT between $VM_i$ is based on $PM_k$, and $VM_j$ based on $PM_\ell$ can be expressed as follows:

$$DT\left(C_{k,\ell}\right) = hop\left(C_{k,\ell}\right) \times TPM\left(VM_i, VM_j\right) \quad (4)$$

where $hop\left(C_{k,\ell}\right)$ is calculated according to the virtual region covering $PM_k$ and $PM_\ell$.

### 2-4- Model of the energy consumption of switches

The energy consumption model for switches presented in this section can be used in all tree topologies, considering the communication structure of the connections. If the communication data are located on a server, the ToR switch does not need to be switched on. In this particular case, all network switches are therefore idle. Note that the inactive switches in the DC are not switched off because restarting and configuring them is time-consuming. In DCs that should be constantly ready to receive and process VMs, this waste of time is not logical. The switches consume little energy when idle and in a half-lit state, which we ignore. If the communication data are in zone 1 (Figure 1), step 2 is activated. Communication data in zone 2 requires 4 steps and three switches to be switched on. Similarly, communication data in zone 3, which must pass through the core switch, requires 6 steps and five switches to be switched on. Therefore, the number of switches required to establish a connection between the data can be calculated from equation Eq. (5):

$$Count\_sw_s^{Full} = hop - 1 \tag{5}$$

Therefore, the energy consumption of all switches can be calculated from Eq. (6) as follows:

$$TPSW = \begin{cases} P\_sw_s^{Full} \times Count\_sw_s^{Full} & if\ hops \geq \\ 0 & otherwi \end{cases} \tag{6}$$

The mathematical model of communication-driven two-objective optimization is defined in the form of Eq. (7) to (15).

$$\min \sum_{j=1}^{m} P_j \times x_j = \sum_{j=1}^{m} \left( \left(P_j^{full} - P_j^{idle}\right) \times \sum_{i=1}^{n} ReqCpu_i \times y_{ij} + P_j^{idle} \right) \times x_j \tag{7}$$

$$min \sum_{i=1}^{n} \sum_{j=1}^{n} hop\left(C_{k,\ell}\right) \times TPM\left(VM_i, VM_j\right) \tag{8}$$

$$min \sum_{i=1}^{n} \sum_{j=1}^{n} Count\_sw_s^{Full} \times P\_sw_s^{Full} \tag{9}$$

$subject\ to$:

$$x_j \in \{0,1\} \tag{10}$$

$$k, \ell \in \{1, \dots, m\} \tag{11}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} = 1, where \quad x_{ij} \in \{0,1\} \tag{12}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m} y_{ij} = 1, where \quad y_{ij} \in \{0,1\} \tag{13}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m} ReqCpu_i \times y_{ij} \leq CpuThr \times x_j \tag{14}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m} ReqMem_i \times y_{ij} \leq MemThr \times x_j \tag{15}$$

To solve this hybrid two-objective optimization problem, which requires simultaneous optimization of three functions of equal importance, we present a genetically based hybrid metaheuristic algorithm, the details of which are explained below.

**2-5- Genetic Algorithm (GA)**

Conventional genetics was originally designed for the optimization of single-purpose functions. Therefore, to solve some equally important target functions with genetic algorithm, changes must be made to it. In addition, the necessary data building must be defined in which the information of the servers is stored. the structure of genes and chromosomes is also defined for problem coding.In the problem statement, it was specified that n requested VM numbers should be inserted on m numbers; therefore, the length of the chromosome is equal to n, and the genes are natural numbers belonging to $\{1, \dots, m\}$. To create the initial population, a two-dimensional matrix is defined as $Pop[1..PopSize][\,1..n]$, where PopSize is a constant indicating the size (number of chromosomes) of the initial population. Each individual $Pop[\mathrm{i}][1..\mathrm{n}]$of the population is represented as chromosome $\mathrm{Ch_i}$ and is a random solution to the problem. Therefore, $Pop[i][j] = k$ in a solution means that $\mathrm{VM_j}$ is placed on $\mathrm{PM_k}$. Each $\mathrm{PM_j}$ has a data structure called $\mathrm{DS_j}$, so that all individuals in the initial population (Pop) have their own specific data structure. As a result, the two-dimensional matrix $\mathrm{DS}[1..PopSize][1..m]$ can be extracted from $Pop[1..PopSize][\,1..n]$and vice versa.

**3- Experiments**

**3-1- Scenarios, datasets, and parameter settings**

The performance and efficiency of the proposed method should be tested and analyzed under different scenarios to ensure its accuracy. Because the performance of the proposed method depends on the size and number of VMs and PMs, different scenarios are defined to cover different situations. Under the conditions where the number of PMs is fixed and the number of requested VMs increases, and under the conditions where the number of VMs and PMs gradually increases, the results of the algorithm are checked according to the specified parameters and the changes in the results are monitored. To perform the evaluations, we used a family of Dell servers with 6000 MIPS processing power and 8 GB memory capacity in a homogeneous DC, which consumes energy from any server in idle mode. The active mode is $P_j^{Idle} = 162\ and\ P_j^{full} = 215$ . We have opted for a fat tree and a homogeneous DC topology, where the switches can cover up to 16 physical servers with some ports p=4s. Moreover, this algorithm can be extended to heterogeneous DCs and other topologies on a larger scale. Different scenarios were designed to obtain reliable results. Here, we design five different scenarios, each of which is implemented on 3 different datasets with different correlation coefficients between the source vectors. Considering the appropriate value for the variable $par \in [0..1.0]$,we obtain the source vectors with correlation coefficients in the range {-0.85,-0.07,+0.85}. In addition, the threshold of physical server resources for hosting VMs is set at 90%. In addition, the traffic pattern matrix (TPM) variable determines the amount of data flow between each pair of VMs, which can be [25-250 MB]. Of course, the volume of data sent between independent VMs is assumed to be zero. Note that the values of the components of this matrix can be extracted from the switches and DC or from data mining techniques.

### 3-2- Evaluation

To evaluate the proposed GA-based two-objective algorithm (BOGA), we compared the results of its implementation with the GA-based multi-objective VMP method (MOGA) in [1], the ACO-based two-objective method (BOACO) [48], the famous innovative FFD method [35], and the random VMP method in terms of the total energy consumption of servers and switches and the total bandwidth consumption. The results of the implementations and evaluations are listed in Tables 2 -6. The following scenarios are designed for a fixed number of PMs (16 PMs), with the number of requested VMs increasing gradually.

- Scenario 1 contains 15 requested VMs to be deployed on 16 PMs.
- Scenario 2 contains 20 requested VMs to be deployed on 16 PMs.
- Scenario 3 contains 25 requested VMs to be deployed on 16 PMs.
- Scenario 4 contains 30 VMs to be deployed on 16 PMs.
- Scenario 5 contains 35 requested VMs to be deployed on 16-PM.

**Table 2**: Comparison and evaluation of scenario 1

| Number of VM | Correlation | Algorithm | Band Width | Total Energy |
|---|---|---|---|---|
| 20 | -0.85 | **BOGA** | **8375۲** | **604.8933** |
| | | MOGA | 63800 | 613.5497 |
| | | BOACO | 70900 | 627.8030 |
| | | FFD | 71200 | 789.8030 |
| | | RD | 72400 | 789.8030 |
| | −0.07 | **BOGA** | **67000** | **604.8431** |
| | | MOGA | 67100 | 616.9292 |
| | | BOACO | 68500 | 616.9292 |
| | | FFD | 73150 | 778.9292 |
| | | RD | 73650 | 778.9292 |
| | 0.85 | **BOGA** | **56560** | **580.1452** |
| | | MOGA | 56700 | 588.3252 |
| | | BOACO | 57150 | 588.3252 |
| | | FFD | 57150 | 616.9292 |
| | | RD | 58400 | 616.9292 |

**Table 3**: Comparison and Evaluation of Scenario 2

| Number of VM | Correlation | Algorithm | Band Width | Total Energy |
|---|---|---|---|---|
| 25 | -0.85 | **BOGA** | **120100** | **770.1005** |
| | | MOGA | 121450 | 798.1011 |
| | | BOACO | 134400 | 798.2607 |
| | | FFD | 171850 | 850.3300 |
| | | RD | 164050 | 960.2607 |
| | −0.07 | **BOGA** | **151150** | **1010.900** |
| | | MOGA | 151100 | 1011.300 |
| | | BOACO | 155550 | 1011.300 |
| | | FFD | 148700 | 1011.300 |
| | | RD | 163600 | 1011.300 |
| | 0.85 | **BOGA** | **111200** | **800.0100** |
| | | MOGA | 114500 | 800.0486 |
| | | BOACO | 116800 | 800.0486 |
| | | FFD | 119050 | 800.0486 |
| | | RD | 124150 | 800.0486 |

**Table 4**: Comparison and evaluation of scenario 3

| number of VM | Correlation | Algoritm | Band Width | Total Energy |
|---|---|---|---|---|
| 30 | -0.85 | **BOGA** | **271500** | **943.7210** |
| | | MOGA | 271500 | 969.7314 |
| | | BOACO | 276400 | 979.4034 |
| | | FFD | 286800 | 1141.400 |
| | | RD | 317800 | 1141.400 |
| | −0.07 | **BOGA** | **292800** | **973.9090** |
| | | MOGA | 292800 | 973.9280 |
| | | BOACO | 297600 | 973.9280 |
| | | FFD | 293100 | 1135.900 |
| | | RD | 314350 | 1135.900 |
| | 0.85 | **BOGA** | **245940** | **1001.150** |
| | | MOGA | 246000 | 1015.400 |
| | | BOACO | 259250 | 1015.400 |
| | | FFD | 290500 | 1015.400 |
| | | RD | 277650 | 1015.400 |

**Table 5**: Comparison and Evaluation of Scenario 4

| Number of VM | Correlation | Algorithm | Band Width | Total Energy |
|---|---|---|---|---|
| 35 | -0.85 | **BOGA** | **418510** | **1012.340** |
| | | MOGA | 418500 | 1173.800 |
| | | BOACO | 427100 | 1183.800 |
| | | FFD | 466950 | 1345.800 |
| | | RD | 454900 | 1345.800 |
| | -0.07 | **BOGA** | **410900** | **1190.200** |
| | | MOGA | 410950 | 1202.800 |
| | | BOACO | 445500 | 1202.800 |
| | | FFD | 448050 | 1364.800 |
| | | RD | 480750 | 1364.800 |
| | 0.85 | **BOGA** | **375900** | **1161.020** |
| | | MOGA | 376000 | 1175.600 |
| | | BOACO | 377800 | 1175.600 |
| | | FFD | 439800 | 1175.600 |
| | | RD | 423400 | 1175.600 |

**Table 6**: Comparison and Evaluation of Scenario 5

| Number of VM | Correlation | Algorithm | Band Width | Total Energy |
|---|---|---|---|---|
| 40 | -0.85 | **BOGA** | **638920** | **1124.404** |
| | | MOGA | 639900 | 1426.900 |
| | | BOACO | 655200 | 1442.100 |
| | | FFD | 662650 | 1766.100 |
| | | RD | 680500 | 1880.500 |
| | -0.07 | **BOGA** | **627700** | **1213.500** |
| | | MOGA | 628150 | 1398.300 |
| | | BOACO | 653000 | 1398.300 |
| | | FFD | 656300 | 1560.300 |
| | | RD | 660250 | 1570.300 |
| | 0.85 | **BOGA** | **565650** | **1232.300** |
| | | MOGA | 565650 | 1241.800 |
| | | BOACO | 588200 | 1403.800 |
| | | FFD | 646250 | 1403.800 |
| | | RD | 653250 | 1565.800 |

As can be seen from the tables and diagrams, the three metaheuristic methods at the top of the tables are very close to each other in terms of optimizing total energy consumption and bandwidth. In terms of bandwidth consumption, the MOGA algorithm is very similar to the proposed algorithm because the solution to reduce traffic and bandwidth in both algorithms follows a similar strategy. However, as far as the energy consumption of all servers and switches is concerned, the MOGA algorithm is more efficient because of its focus on the energy of the switches; the proposed algorithm of this research shows certain superiority.

**4-Results**

The results of the mathematical model of the proposed method consist of the energy consumption model of servers, the energy consumption model of switches, and the bandwidth consumption model, which has been formulated and presented in the form of a comprehensive two-objective optimization model. To solve this hybrid problem with two objectives, a hybrid meta-heuristic algorithm based on a genetic algorithm was presented. In this way, first the genetic algorithm was implemented and its results were applied to the complementary algorithm for local search and discovery of the possible optimal answers in the neighborhood of the answers and finally the relative optimal final results were computed. The calculations in the scenarios and the correlation coefficients {-0.85, -0.07, +0.85} between the processor and memory resource requirements were performed as follows:

- Scenario 1: 20 VM on 25 PM
- Scenario 2: 25 VM on 30 PM
- Scenario 3: 30 VM on 35 PM
- Scenario 4: 35 VM on 40 PM
- Scenario 5: 40 VM on 45 PM.

The efficiency of the proposed algorithm (BOGA) has been compared with the meta-heuristic three-objective method based on GA (MOGA), the meta-heuristic two-objective method based on the ACO algorithm (BOGA), the heuristic FFD method and the random search method (RD). The results of the simulations have shown the significant superiority of the proposed method compared to the other mentioned methods in terms of saving the total energy consumption and reducing the amount of data stored on the shared channels of the data center network. Furthermore, the results of the simulations on the tree topology of the fat-tree network show that the proposed algorithm has a high scalability for development and implementation on tree networks compared to other tested methods. With considering given results, the following items were also proposed to improve the model in future. 1- considering the possibility of new virtual machines entering the system at any time and calculating the cost and energy of migration to provide a dynamic model for virtual machine deployment. 2- Presenting a comprehensive model for energy computation in wired and wireless networks b) Practical suggestions 1- Computing the

possibility of traffic distribution in heterogeneous networks with asymmetric topologies to reduce the number of servers and switches required to increase the size of the network. 2- Using data mining to extract communication patterns in wireless networks to calculate the sent traffic in the comprehensive network model.

**References**

[1]J. Masoudi, B. Barzegar and H. Motameni, "Energy-Aware Virtual Machine Allocation in DVFS-Enabled Cloud Data Centers," in IEEE Access, vol. 10, pp. 3617-3630, 2022, doi: 10.1109/ACCESS.2021.3136827.

[2]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.

[3]A. Gopu and N. Venkataraman, "Optimal VM placement in distributed cloud environment using MOEA/D," Soft Computing, vol. 23, no. 21, pp. 11277-11296, 2015.

[4]F. L. Pires and B. Barán, "Multi-objective virtual machine placement with service level agreement: A memetic algorithm approach," IEEE, pp. 203-210, 2013

[5]D. Kliazovich, P. Bouvry, and S. U. Khan, "GreenCloud: a packet-level simulator of energy-aware cloud computing data centers," The Journal of Supercomputing, vol. 62, no. 3, pp. 1263-1283, 2012.

[6]D. Serrano et al., "SLA guarantees for cloud services," Future Generation Computer Systems, vol. 54, pp. 233-246, 2016.

[7]M.-H. Malekloo, N. Kara, and M. El Barachi, "An energy efficient and SLA compliant approach for resource allocation and consolidation in cloud computing environments," Sustainable Computing: Informatics and Systems, vol. 17, pp. 9-24, 2018.

[8]G. Cao, "Topology-aware multi-objective virtual machine dynamic consolidation for cloud datacenter," Sustainable Computing: Informatics and Systems, vol. 21, pp. 179-188, 2019.

[9]A. Jobava, A. Yazidi, B. J. Oommen, and K. Begnum, "On achieving intelligent traffic-aware consolidation of virtual machines in a data center using Learning Automata," Journal of computational science, vol. 24, pp. 290-312, 2018.

[10]G. L. Stavrinides and H. D. Karatza, "An energy-efficient, QoS-aware and cost-effective scheduling approach for real-time workflow applications in cloud computing systems utilizing DVFS and approximate computations," Future Generation Computer Systems, vol. 96, pp. 216-226, 2019.

[11]J. Krzywda, A. Ali-Eldin, T. E. Carlson, P.-O. Östberg, and E. Elmroth, "Power-performance tradeoffs in data center servers: DVFS, CPU pinning, horizontal, and vertical scaling," Future Generation Computer Systems, vol. 81, pp. 114-128, 2018.

[12]P. Festa, "A brief introduction to exact, approximation, and heuristic algorithms for solving hard combinatorial optimization problems," IEEE, pp. 1-20, 2014

[13]Randy L.Haupt and sue Ellen Haupt. "Parctical Genetic Algorithm" (2nd ed),USA:Wiley. 2004.

[14]A.Alkan and E.Ozcan. “Memetic Algorithms for Timetabling", Evolutionary Computation, 2003. CEC '03. The 2003 Congress on, 3, pp 1796-1802. 2003.

[15]Shaw, R., Howley, E., & Barrett, E. (2019). An energy efficient anti-correlated virtual machine placement algorithm using resource usage predictions. Simulation Modelling Practice and Theory, 93, 322-342. 2019.

[16]Tordsson, J., Montero, R. S., Moreno-Vozmediano, R., & Llorente, I. M. Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers. Future generation computer systems, 28(2), 358-367. 2012.