# AN OPTIMAL CUCKOO SEARCH ALGORITHM FOR VM SELECTION FOR ENERGY EFFICIENT MIGRATION IN CLOUD COMPUTING

**LOVELEENA MUKHIJA[1], ROHIT SACHDEVA[2]**

**1 .Computer Science, Punjabi University, Punjab, India**

**2. Computer Science, Punjabi University, Punjab**

rsachu.147@gmail.com

***Abstract*** *-Cloud Computing is known be a service platform for all kind of services related to Infrastructure, Platform and Software services. Virtual Machine(VM) placement over Physical Machine(PM) is one of the processes that is implemented to support elasticity. VMs are migrated from over utilized PM and underutilized PMs in order to keep a balance of load and power consumption at the data centre. This paper proposes a novel behavior in Cuckoo Search(CS) optimization from the meta-heuristic approach. A simulation architecture of VM placement and VM selection is presented and the performance of the proposed algorithm architecture is evaluated using Quality of Service(QoS) parameters. The proposed work is also compared with other state of art techniques in terms of power consumption and service level agreement violation(SLA-V). The proposed work shows significant improvement ranging from 2-16% in different aspects and the results are elaborated in the similar fashion.*

***Keywords***- *VM Selection, Cuckoo Search, meta-heuristics, Cloud Computing technique, power consumption.*

.

## 1. INTRODUCTION

An intrinsic characteristic of cloud computing is provisioning the software and hardware services to both industrial and businesses with guaranteed availability and reliability of resources. Cloud computing provides a range of services to physical servers such as termination, flexibility in deployment, services replication, and migration as shown in Fig. 1. A variety of computing services are provided in different forms: (a) Infrastructure as a service such as Amazon Elastic cloud (b) Software as a service such as Google App Engine, and (c) Platform as a service such as SalesForce.com [1], [2].

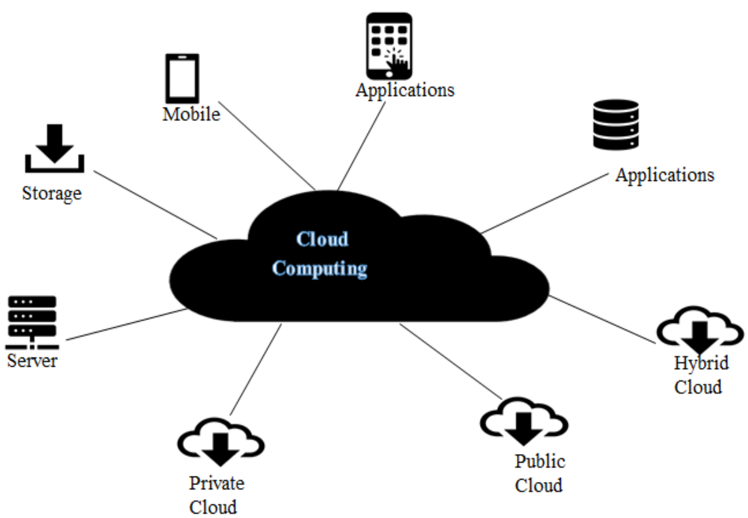

**Fig. 1 Overview of Cloud Computing**

Additionally, Cloud data centers provide a variety of add-on features and customizable settings to meet the storage requirements of the customers or end users ensuring Quality of Service (QoS). The efficient utilization and management of cloud resources are important while meeting the demands of the users. However, the work studied in literature shows that majority of the physical machines in the cloud data centers stay idle and employ a small proportion of energy (about 10% to 20%) to remain in active state. This cause many problems such as magnifies the energy consumption, overheating, violation in Service Level Agreement (SLA), decline in profit margins and increase in operational cost of the cloud environment. In such a scenario, virtualization plays a prominent role in handling the mentioned constraints by effectively migrate the Virtual Machines (VM) from highly loaded machines to underutilized machines. Virtualization act as a layer between the computer system and hardware (such as Virtual Machine Monitor (VMM)) used to host the multiple machines. It also partitions the system resources of different machines into various VM that can execute different programmes. These virtual machines(VMs) enable the execution of numerous programmes in a performance-isolated setting. As a result, cloud computing companies offer limitless virtual computers through virtualization, guaranteeing the users' satisfaction with the service. Each VM that is hosted by a physical machine (PM) needs a specific number of processors, bandwidth, and memory that is stipulated by the end client. Therefore, VM consolidation plays a significant role for the management of resources by considering the four main problems [3].

- Determination of overloaded PM’s [4]
- Determination of under loaded PM’s [5]
- Selection of VM’s for migration from overloaded PM [6]
- Finding a new place for VM placement [7]

In the past few years, researchers mainly focussed on VM placement problem to minimize the consumption of energy by optimal placement. VM placement can be static and dynamic. Static placement is the creating of VM’s on suitable PM’s while dynamic placement is the process of migrating the V from one PM to another during the running state. In this paper, we focussed on dynamic placement through VM consolidation to reduce the consumption of power. However, placing an excessive number of VM’s on same physical machine may results in substantial increase in SLA violations. The majority of cloud computing issues, particularly resource allocation and scheduling, may be expressed as conventional optimization issues. The major goal of the optimization is to increase resource utilisation by cutting costs and provide the best possible services [19]. Therefore, metaheuristic techniques have been employed that optimizes the energy consumed by the machines. The Cuckoo Search (CS) algorithm has been widely used in cloud computing to handle challenging multi-objective optimization problems throughout the past few decades [8].To find the global optimum, Levy flight distribution was used. The Genetic Algorithm (GA) and CS play significant roles in the selection of the network path and infrastructure for VM migration when compared to the other optimization methods. For multi-objective optimization problems involving network selection and resource allocation for task distribution in the topology of the cloud network, the standard Cuckoo Search algorithm has been employed [10]. Furthermore, a multi-objective load balancing algorithm based on traditional CS optimization was devised [11]. The scheduling paradigm of a directional acyclic graph of the challenge can be applied with the described laying and migration in this study. Then, utilising CSO, a new algorithm was developed with the objective of improving the efficiency of the centre while lowering cloud costs and minimising wastage of time. A strategy with strong local search typically has limited global exploration ability, according to accepted criteria. This is seen while addressing complex optimization problems; when the algorithm employs just one search method, CS quickly catches the local optimum. Though meta-heuristic algorithm architectures are not new in the field of VM selection or placement, the contributions of the proposed work are as follows.

- Design of a new behaviour in CS algorithm for a better swarming result.
- Selection of optimal VMs based on QoS parameters
- Comparison of proposed work with other state of art techniques.

The rest of the paper is organised in the following manner. Section 2 illustrates the related work whereas section 3 describes the proposed work scenario. The results are illustrated in section 4 whereas the paper is concluded in section 5.

## 2. RELATED WORK

In order to save energy and maximise resource efficiency, VM migrations are typically carried out. Analysis of the problems and difficulties associated with VM migration is done at several levels in detail. The main obstacles to VM migration are network connection continuity and memory utilization during VM migration. The prediction and monitoring strategy is typically used to reduce the load occurrences of physical equipment to order to increase energy efficiency. In other words, migration lowers the number of PM in use and lowers the energy consumption.

Naeen et al. 2018 proposed a stochastic process-based method for dynamic consolidation of VM's with dynamic workloads. By using this strategy, authors hope to cut back on expenses like energy use, the frequency of migrations, SLA breaches, and the costs associated with switching PM’s. For the placement of VM's, authors presented the stochastic-based best fit decreasing technique, which is based on stochastic processes. This approach places VM's on PM's so that their average total stochastic resource consumption is greater than that of the other physical machines. In other words, a PM is chosen as the destination if, once the VM has been moved, that PM will have the least amount of resource waste and if there is a low likelihood that it will be overloaded [12].

Barlaskar et al. 2018 suggested the Enhanced CS algorithm in this study to address the problems with VM placement with a focus on energy consumption. Three different workloads from the CloudSim tool had been used to assess how well the suggested method performs. The suggested approach was compared to the current ACS, GA, and Firefly algorithms as part of the evaluation process. The comparison findings show that while keeping a constant performance for SLA and VM migration, the developed ECS algorithm uses less energy than the previous algorithms. In comparison to GA, OFS, and AC, the ECS algorithm uses about 25%, 27%, and 26% less energy, respectively. The limitation of the study was proposed ECS technique unable to provide desired results in a heterogeneous environment and performance metrics were also limited [16].

Liu et al. 2019 suggested a multi-population ACS with lesser complexity in conjunction with Extreme Learning Machine (ELM) prediction algorithm. The proposed algorithm forecasts that the host state will be changed using ELM before the VM shifted on the busy host is relocated whereas the VM on the overloaded host will be consolidated to another, to the normal host especially host with higher usage and less load. The authors consider the Local search mechanism of ants and enhance each population's results to cut down on SLA violations. The investigation's findings have shown that proposed approach reduces energy consumption and migration times when compared to existing algorithms. But, the limitation of the ELM algorithm is increase in energy consumption for real world data centers [3].

Sayadnavard et al. 2019 suggested a method for placing VM's using failure awareness. The authors anticipated each PM a grade taking into account the costs associated with energy and reliability. The proposed policy chooses a PM with the best dependability value as the destination to avoid redundant migrations. It indicates that the intended VM might successfully complete its mission on that PM without any errors, avoiding the need for re-migration and the associated overhead. The study was limited to provide the desired results in terms of minimizing the energy [13].

Salami et al. 2021 suggested a study in which minimum number of physical machines must be employed when hosting multiple virtual machine requests. This study was based on developing an algorithm for solving the VM placement problem that is inspired by the CS. The performance of the algorithm is enhanced via new cost and perturbation functions. Two well-known benchmark datasets served as the testbeds for the proposed technique. It performed better than multi-CSA, the best-fit decreasing, the first-fit decreasing, and the GA technique. The present study was limited to two dimensions such as memory and CPU, the other dimensions need to consider for efficient placement [17].

Ghetas 2021 suggested a new technique for placing new VM’s based on the Monarch Butterfly Optimization algorithm (MBO) named as MBO-VM to increase packaging efficiency and decrease the number of active PM’s. The effectiveness of the suggested MBO-VM strategy was evaluated using both synthetic and actual cloud workloads using the CloudSim tools. The simulation results from simulator demonstrate that MBO-VM greatly outperforms conventional VM placement methods in terms of energy consumption and average usage of CPU and RAM. The suggested MBO-VM can enhance packaging efficiency and more efficiently cut down the number of active PM's. The main drawback of the study is limitation to apply in dynamic environment during the on-going VM migration process and unable to tackle task scheduling issues [20].

Venkata Subramanian and Shankar Sriram 2022 employed the CS algorithm, an intelligent metaheuristic technique, to select the safest and best dynamic network path. The majority of research has been focussed on resource load balancing and energy reduction, but authors not been able to provide safe and ideal resource use. In comparison to existing strategies, the created hybrid movement strategy improves search capability by enlarging the search field and using a combined risk score assessment of each PM as a fitness criterion for maintaining security. The Google cluster dataset was used to evaluate the suggested method's viability. But, limited results obtained for levy flight mechanism and need of different compressing techniques for SLA measures [8].

Al Wesabi et al. 2022 presented a new hybrid resource allocation for the cloud computing environment in this paper. The suggested model performs feature extraction initially in response to job requests from several clients, and principal component analysis (PCA) was then used for feature reduction. The proposed approach then makes use of the combined features for the best resource allocation. For the purpose of best resource allocation, the proposed model combines the Group Teaching Optimization Method (GTOA) with the rat swarm optimizer (RSO) algorithm. The allocation of resources across VMs in cloud datacenters is made better by the merging of GTOA and RSO algorithms. A thorough set of simulations were run using the CloudSim programme for experimental validation. But study was limited to schedule the tasks and aggregation of data in a prescribed environment [9].

## 3. PROPOSED WORK

There are a large number of server nodes in the cloud environment and assumptions has been made such that infrastructure of data center is fully virtualized and different applications are running on VM's. The assignment of VM to PM is considered as multi-dimensional problem and the characteristics of servers need to be considered as two-dimensional such as CPU utilization and memory usage. The CPU utilization is considered as sum of power utilized by Virtual Machines and the same concept applicable to memory usage [20]. In other words, many VMs are hosted by each server hosts and appropriate resources are provided to run different processes. The total resources consumed by the server are the sum of total resources utilized by the VMs and given in Eqn 1 and Eqn 2.

$$U_b^{CPU} = \sum_{c=1}^{a} y_{cb} v_c^{CPU} \quad (1)$$

$$U_b^{Memory} = \sum_{c=1}^{a} y_{cb} v_c^{Memory} \quad (2)$$

Where, $U_b^{CPU}$ and $U_b^{Memory}$ are the power consumed by the CPU and Memory of server 'b', a is the number of active servers, $y_{cb}$ indicates whether $VM_c$ hosted on server 'b' and$v_c^{CPU}$, and $v_c^{Memory}$ is the memory requirement of $VM_c$.
The utilization of resource is limited for each server by some thresholds that avoid the degradation of performance and migration of VM. For example, the threshold level has been set for CPU and utilization of memory to 80% each. Consider that 20% and 30% be the memory and CPU requirement from the any VM, and the physical server cannot meet the requirement of this VM and cannot host the same due to shortage of resources. It is seen that host 1 is 100% utilized, host 2 is 90% utilized, host 3 is 70% utilized and host 4 is 80% utilized. In such scenario, VMs are migrated from over utilized host to a host that can accommodate the VMs. In this context, two major objectives are considered in this research namely, to minimize the wastage of resources and to minimize the energy consumption considering the SLA constraint.

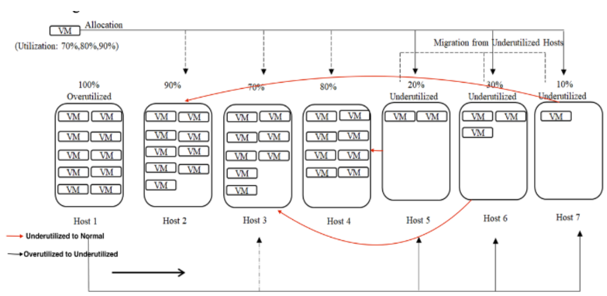


**Fig. 2 Migration and Placement of VM**

### 3.1. Resource Wastage Model

The number of active PM's that can accommodate additional VM is significantly influenced by the resources available on each server [14]. In order to host as many VM as feasible with the fewest number of active servers, an effective VM placement solution balances resource utilisation across all dimensions. In other words, the key of VM placement is to employ resources efficiently to minimise resource waste. The potential cost for each active server is calculated using the following equation.

$$R_b^{Wasted} = \frac{\left|r_b^{CPU} - r_b^{Memory}\right| + \epsilon}{U_b^{CPU} - U_b^{Memory}} \quad (3)$$

$R_b^{Wasted}$ is the wastage of resources from physical server 'b', and $r_b^{CPU}$ and $r_b^{Memory}$ is the power remained from physical server 'b' for CPU and memory resources. ∈ is the positive real number whose value is very small about 0.0001. It is added due to random uncertainties in the cloud environment. These consumption models have the base as Modified Best Fit Decreasing(MBFD) algorithm for VM placement and Minimization of Migrations(MM) as hotspot detection for VM selection.

#### *3.1.1 Energy model*

Practitioners demonstrated that there is a linkage between the CPU utilization and power consumption which may be linear [15]. For idle server, the energy consumed by the machines represents 50-70% of the $T_E$ consumed by$PM_a$. Therefore, idle servers need to switch off to reduce the consumption of energy in the data center. Ideally, energy consumption by the server in terms of CPU utilization and memory utilization is represented as given in Eqn 4.
. The potential cost for each active server is calculated using the following equation.

$$T_b^{energy} = \left(T_b^{Busy\ energy} - T_b^{idle\ energy}\right) \times U_b^{CPU} + T_b^{idle\ energy} \quad (4)$$

$T_b^{energy}$ is the total energy consumed of physical server 'b', $T_b^{idle\ energy}$and $T_b^{Busy\ energy}$ is the average power consumed by idle and average power consumed by the busy machines

## 4. VM PLACEMENT PROBLEM USING THE CS OPTIMIZATION TECHNIQUE

### 4.1 The Traditional Approach

In this section, VM migration and its placement is considered as a multi-objective optimization problem which is aimed to optimize the energy consumption by minimum violation of service level agreement. Evolutionary algorithms such as Particle Swarm Optimization (PSO), Ant Colony System (ACS), Cuckoo Search (CS), and many more that are multi-objective in nature are population based that are used to find an optimal feasible solution. For the selection of optimal solution, these evolutionary algorithms depend upon the dominant principles of the birds. The dominance concept can be expressed as multi-objective problem for minimization to place VM with x parameters and z objectives. follows.

$$Minimize\ f(y) = [f_1(y_1, \ldots, \ldots y_x), \ldots, f_z(y_1, \ldots, y_x)] \quad (5)$$

$$y = (y_1, \ldots, y_x) \in Y\ f = (f_1, \ldots, f_z) \in K \quad (6)$$

Where $y$ signifies the solution vector, $Y$ considered as parameter space, $f$ is considered as objective vector and $K$ is the objective space. To simulate the VM selection problem, the assumptions has been made that DC consists of $x$ VM's that are assigned to number of PM's. Consider VM be an array of virtual machines $VMs = \{V_1, V_2, \ldots . V_x\}$ and there is a two dimensional vector such as $v_a = \{v_a^{cpu}, v_a^{memory}\}$. Additionally, assume that number of physical machines$PM = \{PM_1, PM_2, \ldots . PM_z\}$.

Yang and Deb developed the Cuckoo Search algorithm to solve the engineering problems which may be multi-objective and tend to address the engineering problem; likewise, this article addresses the problem of VM selection from overloaded PM by finding the relevant solutions. As the application of CS algorithm is widely accepted across various domains, the researchers still focused on exploring the Cuckoo Search optimization(CSO) algorithm across different contexts for the selection of VM.

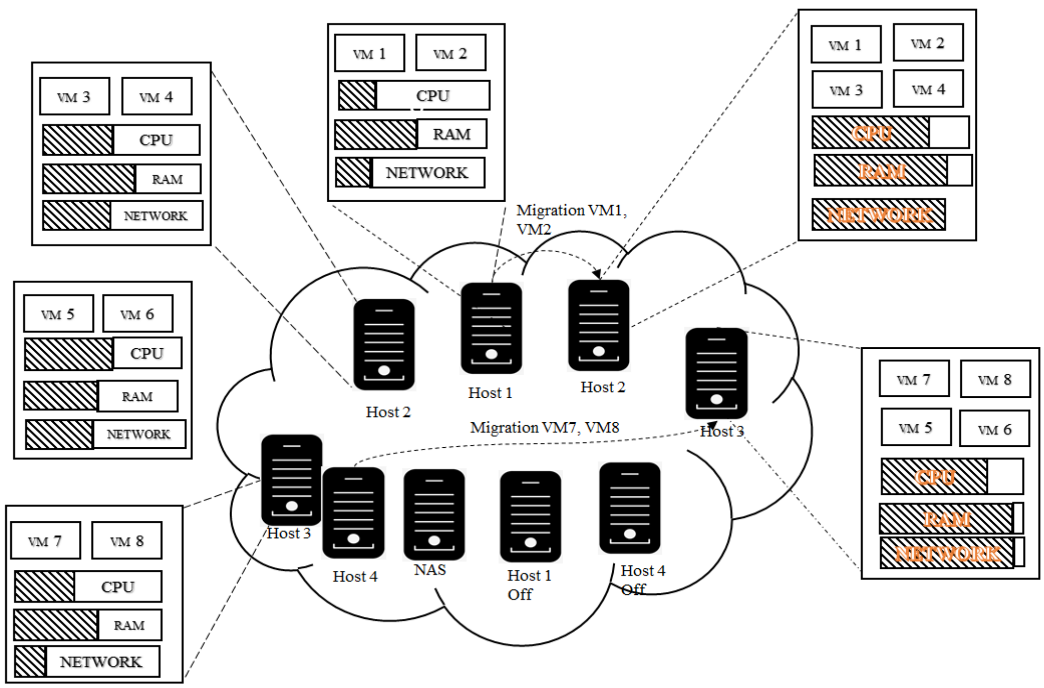


**Fig. 3 Overview of VM placement and migration as per CS algorithm**

### 4.2. The Proposed Solution

The proposed work is extended for the optimal selection of VM to initiate the migration process using the CS algorithm. The population based technique solves the optimization issue considering the minimum parameters. The proposed algorithm is formulated considering the birds behaviour as cuckoo lay egg at a time and nests are selected randomly for placement and then dump it as shown in Fig. 3. The next generation identified the best nest and select the highest quality egg. For the nests that are fixed, the probability to identify egg is [0,1]. In such a scenario, egg is either removed from the host nest or the bird which is hosting the nest leave the current nest and builds a new nest. The relevant solution is identified and incorporated for the placement of VM. To determine the best optimal solution for VM placement, the migration process is initiated by selecting the suitable VM considering the probability factor. The host PM either selects the migrated VM or transfer to the next VM based on selection probability.

In order to understand the proposed CS algorithm, the ordinal measures of CS architecture are illustrated as follows. The nest having egg indicated the valid solution and one egg which is laid by the cuckoo directly proportional to the one solution. Each egg in the nest indicates a viable solution, and one egg directs to one solution. Each nest and position of egg indicates a solution and the initial solution is considered randomly and the process to update the position is given as follows

$$o_i(j+1) = o_i(j) + \delta \otimes \varrho(\varphi) \quad (7)$$

Where, the position of the nest is $(j+1)^{th}$ generation which is represented as $o_i(j+1)$, the nest position having $j^{th}$ generation is represented as $o_i(j)$. $\delta$ is the step size and $\varrho(\varphi)$ considered as levy distribution which is used to produce random search vector. $\otimes$ represented the multiplication and the global search optimization is done and the levy flights role is important and represented as mathematical solution given as follows: -

$$\varrho(\varphi) \sim v = t^{-\varphi} (1 < \varphi < 3) \quad (8)$$

A random walk process is created considering the consecutive steps and the levy walk is used for the generation of solutions. The local search and global search is accelerated considering the solutions acquired by levy process. The local solution is randomization which is far enough from the current solution which is best and optimal. This process ensure that the VM placement problem overcome the issue of finding the local optimal solution. The levy flights consider for updated solutions for every simulation round and best solution is detected. Later on, the solutions that are retained eliminated on a random basis until the termination criteria have been met. The different steps for CS algorithm are given as follows: -

- In the initial step, the different parameters of the system are initialized such as the number of nests for birds, termination criteria of the algorithm, search space dimension, probability factor, and step size.
- The nest population is initialized randomly as $m^{(0)} = [o_1^{(o)}, o_2^{(o)}, o_3^{(o)}, \ldots, o_n^{(o)}]^t$. The fitness function is obtained for the initial population and the best net position is identified as $o_{Gb}^{(o)}$. Next, the position of the nest is updated considering the weight coefficient which is given as follows: -

$$wt = wt_{max} - (wt_{max} - wt_{min}) \times \frac{S_{current}}{S_{max}} \quad (9)$$

Where, the maximum and minimum weight coefficients are represented as $wt_{max}$ and $wt_{min}$ respectively. $S_{current}$ represents the current simulation round and $S_{max}$ are the maximum simulation rounds of the optimization technique. Considering the above equations, new positions are detected and the fitness function is computed for new updated positions.

- The current and the past fitness value is compared and the best nest is selected considering the

**Algorithm 1** PROPOSED cr-Cuckoo Algorithm

Input: $AllocationTable$

**Output**: VM list for migration($VMlist - m$)

$PMl$=$AllocationTable$.OverloadedPMs

**For**(P in $PMl$)

**QoS**=$AllocationTable$.P.Qos // Extract the QoS parameters , in proposed case it is power consumption and load

Initialize of number of nests or hives as by applying k-means for cuckoo eggs, termination criteria of the algorithm, search space dimension, probability factor, and step size as levy flight $lf$=5;

Calculate the fitness function for the initial population

[$Cindex$,$Ccentroid$]=kmeans(QoS,2) // Divide the entire population in 2 nests or hives

Identify the best optimal nest position as $o_G b^(o)$)VMl=AllocationTable.$Find$(P)$ashost$

**For** ( $VM$ as egg in $VMlist$ do) // For each VM in overloaded list

**Reward=**[]; // Generate a reward matrix based on co-relation defined in Figure 4

**For**( $k$=1:$lf$)

Update the position of nest considering the weight function as per Eqn (9)

$f1$=find(Reward==1); //Find positive rewards

$f0$=find(Reward==0); // Find no rewards

**If**($f1 \leq$f0)

$VMl - m$.append($VM$) // Append to migration list

**EndIf**

**EndForVM**

**EndFor**$_P$

**Return** $VMl - m$ **list** 1

population region and the new positions are given as follows: -

$$P_{j+1} = [o_1(j+1), o_2(j+1), o_3(j+1), \ldots \ldots . o_n(j+1)]^T \quad (10)$$

The nest process is elimination of randomized values and probability factor is discovered.

- The status of the nest is determined using the probability factor ($F_{probability}$) or abandoned state. In such conditions, if there is an egg identified by host then nest is changed and this is formulated as problem of l-dimensional vector given as

$$Q_l = [q_1, q_2, q_3, , \ldots \ldots q_k]$$

For $q_j$ in $Q_l$ ,Each vector follow uniform distribution [0,1] and if $q_j > F_{probability}$ and the positions of the nests are changed on a random basis .

- The best nest position is formulated as follows:-

$$B_{j+1} = [o_1(j+1), o_2(j+1), o_3(j+1), \ldots \ldots . o_n(j+1)]^T \quad (11)$$

Using the above equation, the best position of the nest is updated and the step is carried out until the termination criteria have been met. If the current solution satisfies the termination criteria then process is stopped and otherwise process is repeated from equation 9. The proposed work can be illustrated using the work flow diagram that is presented in Fig. 4.

As the proposed algorithm uses correlation factor in Cuckoo Search algorithm, it is named as cr-Cuckoo algorithm and the algorithmic description is as follows.

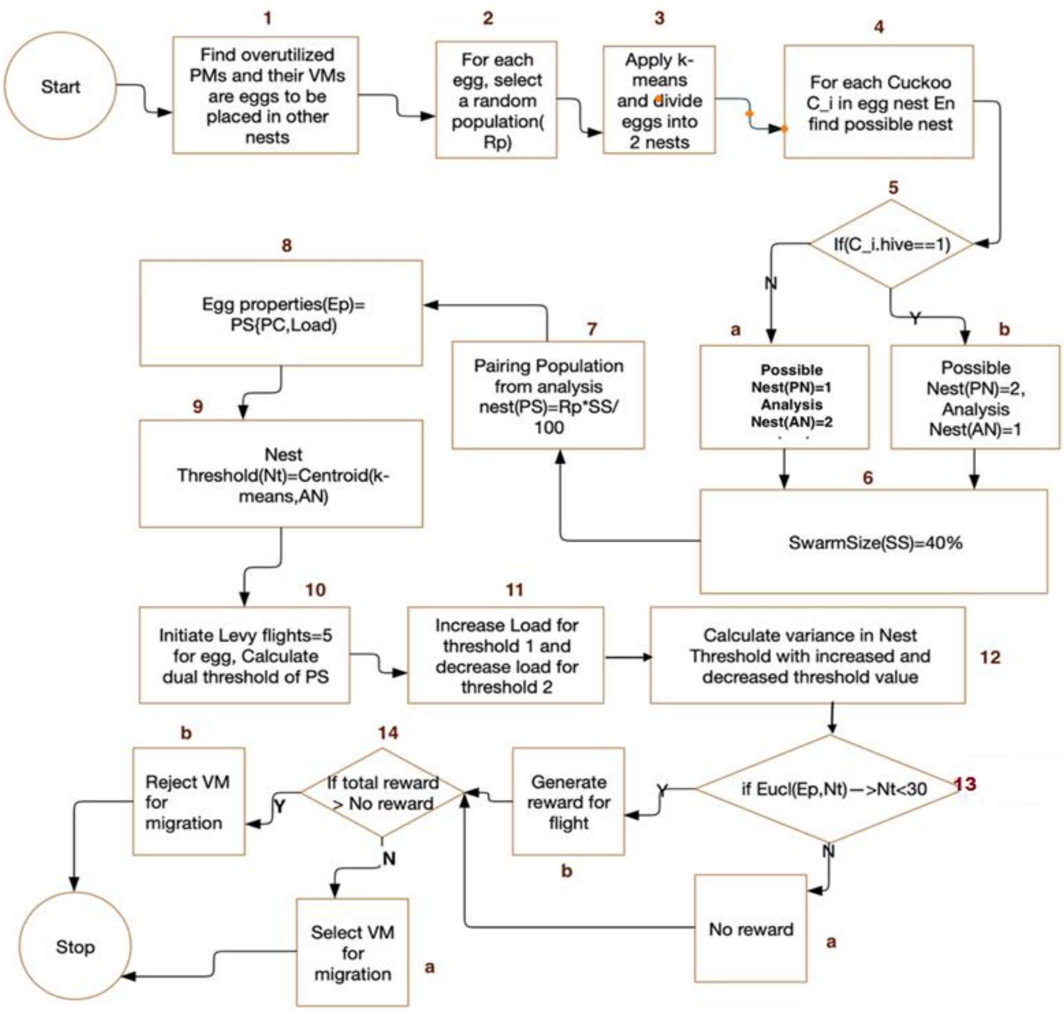


**Fig. 4 The Proposed Work**

The selected VMs are migrated to the PMs that holds the capacity to execute a VM with least power consumption. The proposed work has been evaluated on the base of computation parameters and are illustrated as in Fig. 4.

## 5. Results and Illustrations

The results have been evaluated concerning two major factors namely the overall power consumption and SLA-Violation(SLA-V). The SLA-V has been evaluated using power consumption itself. The overall consumed power in a given interval of time is referred to as energy consumption in proposed work.

$$\text{SLA-V}=\frac{T_b^{energy_{extra}}}{Tb^{Threshold}} \tag{14}$$

Where $TbTh\,rsh\,old$ is the average threshold of energy consumption and $Tbenergyextra$ is the extra energy that is consumed more than threshold. In case of proposed work or any other work that has been simulated, $Tbenergyextra$ do not exceeds beyond $TbTh\,resh\,old$.

Table 1 illustrates the results based on the power consumption in kw that has been computed against low to high VM count with increasing number of resources.

**Table 1. Power Consumption**

| 'Total Number of PMs' | 'Total Number of VMs' | 'Power consumption in kw Proposed' | 'Power consumption in kw Venkata Subramanian[8]' | 'Power consumption in kw Liu[3]' |
|---|---|---|---|---|
| 20 | 100 | 8.44782093 | 9.19742138 | 9.62858337 |
| 40 | 200 | 8.83970183 | 8.94770715 | 10.0638791 |
| 60 | 300 | 9.34222702 | 9.42365967 | 9.62886119 |
| 80 | 400 | 9.68890012 | 11.1052058 | 11.2767915 |
| 100 | 500 | 10.2237205 | 10.9573737 | 11.0655346 |
| 120 | 600 | 10.4568195 | 10.9187601 | 10.6395649 |
| 140 | 700 | 11.1100212 | 11.384044 | 12.4194865 |
| 160 | 800 | 11.2506805 | 12.0635518 | 12.3104625 |
| 180 | 900 | 11.8792794 | 12.5805631 | 12.3689037 |
| 200 | 1000 | 12.4547683 | 12.481835 | 14.0808038 |
| 220 | 1100 | 12.7562842 | 13.1747567 | 13.4767398 |
| 240 | 1200 | 12.996069 | 13.9254438 | 14.2047308 |
| 260 | 1300 | 13.585771 | 15.8880175 | 14.9312173 |
| 280 | 1400 | 13.9676818 | 14.2848142 | 14.241001 |
| 300 | 1500 | 14.5680099 | 16.6473118 | 16.3314745 |
| 320 | 1600 | 14.6442454 | 17.1323777 | 17.1251136 |
| 340 | 1700 | 14.9491171 | 17.2520724 | 16.1118204 |
| 360 | 1800 | 15.2052744 | 15.4607403 | 17.6519946 |
| 380 | 1900 | 15.8521353 | 16.2415503 | 17.7007659 |
| 400 | 2000 | 16.1885752 | 17.4696717 | 17.6672227 |
| 420 | 2100 | 16.576362 | 19.4366141 | 19.4905957 |
| 440 | 2200 | 16.8836436 | 17.7619807 | 19.4031282 |
| 460 | 2300 | 17.7268091 | 18.1166366 | 20.3007874 |
| 480 | 2400 | 18.6907278 | 21.6912878 | 19.0834571 |
| 500 | 2500 | 18.3892184 | 19.0911225 | 20.8097805 |

As it is clear from table 1 that total experimental VMs are 2500 for maximum utilization and minimum VM count is 100. The PMs are considered in a ratio of 1/5 in the proposed approach. For maximum VMs, the proposed work uses 18.39 kw of power whereas other compared state of art technique consumes way more than 2 kw power approximately. The percentage improvement in terms of power can be illustrated using Fig. 5 as follows.

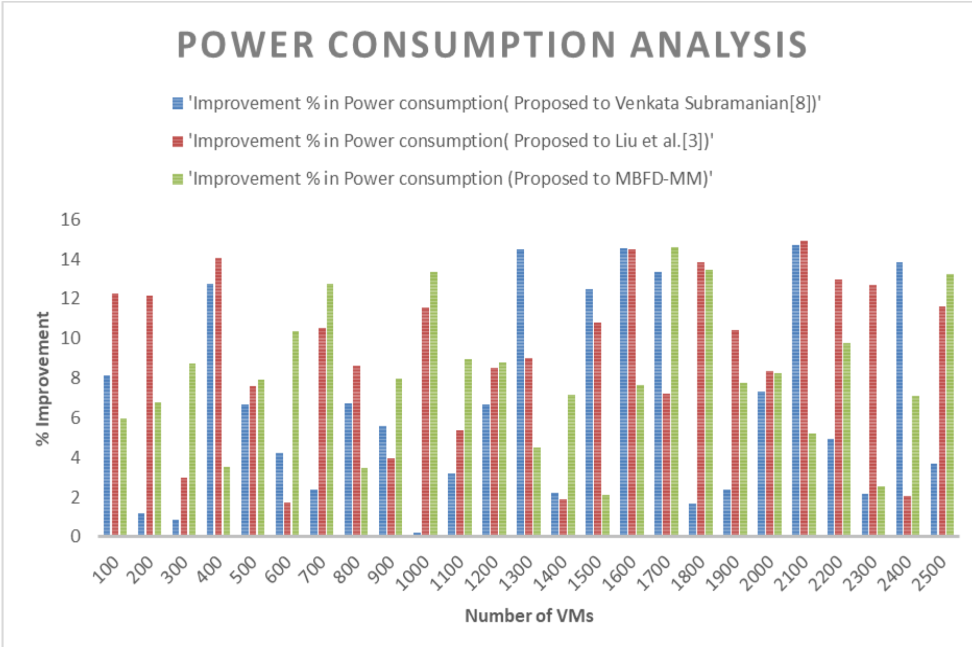


**Fig. 5 Power Consumption**

As shown in Fig. 5, a maximum improvement of 13.33 % is observed when the proposed work is compared to Liu et al[3] whereas a maximum of 8% has been observed when it comes to comparing the proposed work with Subramanian[8]. On comparison with traditional MBFD-MM algorithm, the improvement in power consumption is observed to be 15.83%.The improvement has been noticed due to the improved and adaptive behaviour of proposed CS algorithm that also incorporates load as multi-objective fitness behaviour. As SLA-V is computed based on the power consumption itself, a similar tradition is observed with SLA-V as well. The data presented in table 2 shown the evidences of the claimed analysis.

**Table 2. SLA Value**

| 'Total Number of PMs' | 'Total Number of VMs' | 'SLA-V Venkata Subramanian [8]' | 'SLA-V Liu et al. [3]' | 'SLA-V MBFD-MM.' |
|---|---|---|---|---|
| 20 | 100 | 0.09408778 | 0.09332424 | 0.09559801 |
| 40 | 200 | 0.09107927 | 0.09484226 | 0.08676893 |
| 60 | 300 | 0.08808653 | 0.09771751 | 0.09512323 |
| 80 | 400 | 0.09984496 | 0.08937885 | 0.09060758 |
| 100 | 500 | 0.09791387 | 0.09021476 | 0.09181373 |
| 120 | 600 | 0.09747992 | 0.09547406 | 0.09617521 |
| 140 | 700 | 0.09232334 | 0.09022594 | 0.09634488 |
| 160 | 800 | 0.09758512 | 0.09222196 | 0.09875113 |
| 180 | 900 | 0.0984218 | 0.09051424 | 0.09117096 |
| 200 | 1000 | 0.09585505 | 0.09434619 | 0.08756511 |
| 220 | 1100 | 0.08735006 | 0.09750268 | 0.09901093 |
| 240 | 1200 | 0.10587141 | 0.10284053 | 0.09364148 |
| 260 | 1300 | 0.09687204 | 0.10033815 | 0.09906627 |
| 280 | 1400 | 0.10040121 | 0.10136834 | 0.09725679 |
| 300 | 1500 | 0.09536787 | 0.09305409 | 0.10181426 |
| 320 | 1600 | 0.09733652 | 0.09908423 | 0.1009659 |
| 340 | 1700 | 0.09929936 | 0.103028 | 0.10394464 |
| 360 | 1800 | 0.09275544 | 0.09132323 | 0.09140966 |
| 380 | 1900 | 0.10102023 | 0.09368512 | 0.09329804 |
| 400 | 2000 | 0.10529643 | 0.09039102 | 0.10622235 |
| 420 | 2100 | 0.09353589 | 0.09802261 | 0.09992057 |
| 440 | 2200 | 0.08951739 | 0.0931745 | 0.09331479 |

| | | | | |
|---|---|---|---|---|
| 460 | 2300 | 0.09768612 | 0.09880585 | 0.09739336 |
| 480 | 2400 | 0.09751561 | 0.09540592 | 0.10399083 |
| 500 | 2500 | 0.10110989 | 0.09722213 | 0.0938995 |

As SLA-V is a ratio parameter and the values have been computed on the base of PC, the range of SLA-V does not exceed 0-1 ratio. The improvement in terms of SLA-V is illustrated in Fig. 6 as follows.

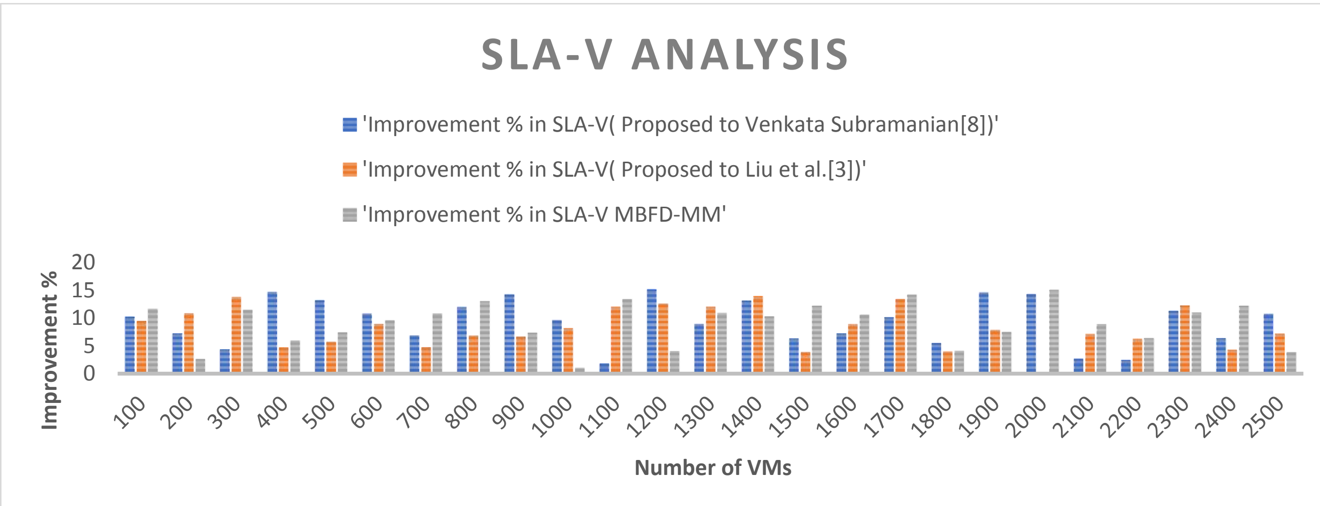


**Fig. 6 SLA-V Improvement Analysis**

The SLA-V in case of proposed work and other compared state of art techniques, justifies a value of 2% to just less than 16% in all the cases. It is observed that, a maximum of 15.32 % violation has occurred in case of increasing load and VM by MBFD-MM whereas proposed work demonstrates 8% of improvement in that comparison. In the similar context, Liu demonstrated 12% of violation whereas for Venkat Subramanian, it is observed to be 11%.

In addition to this, the proposed work also computes the VM migration analysis. The number of VMs that has to be migrated represents the structure of stability of the allocation policy. It is obvious that total number of migrations will increase with the increase in total number of VMs still more migrations represents that the load in the network is not balanced. Fig. 7 presents the migration analysis of proposed algorithm and other state of art algorithms.

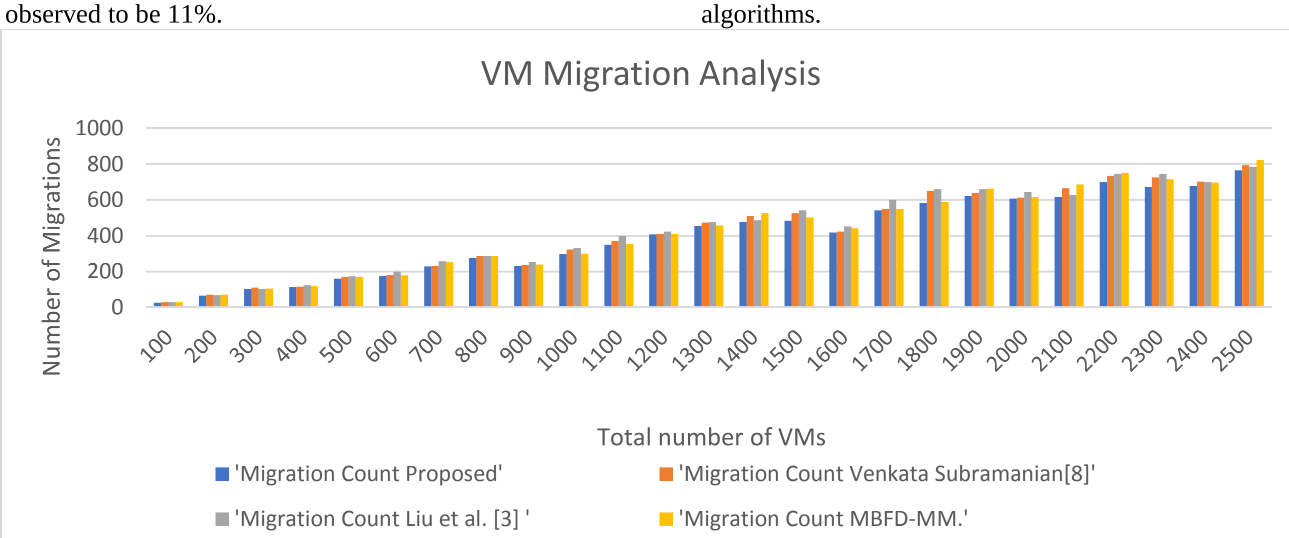


**Fig. 7 VM Migration Analysis**

Due to the appropriate VM selection policy, the proposed work is more power efficient as compared to other techniques that are listed in the reference of comparisons with less number of migrations. For maximum number of VMs viz. 2500, the proposed work migrates 764 VMs to balance the load whereas the migration count for Venkat et al. is 793, for Liu et al. it is 783 whereas MBFD-MM migrates 821 VMs. The average number of migrations in the same sequence is 401,421,430 and 420 VMs

## 6. Conclusion

The paper illustrated the problem of VM selection from the over-utilized PM as the primary concern. A novel behaviour of VM selection is presented using optimized CS algorithm that incorporates levy flights and a reward mechanism is presented. The proposed work has been evaluated with varying load in terms of VMs and the proposed work is primarily focused on selection of the VMs from the over-utilized hotspots or PM as the underutilized PM loses all its concerned VMs. The proposed work has been evaluated based on the QoS parameters in terms of power consumption and SLA-V. A maximum of 2500 VMs have been incorporated in the simulation against 1/5 count of PMs. When it comes to state of art comparison, the proposed work has been compared with two state of art techniques and have shown improvement in terms of power consumption with a maximum value of 15.83% whereas in case of SLA violation, the improvement is noted to be 15.32%. The current state has a lot of future aspects and abilities. Incorporation of learning mechanism may also improve further results.

## References

[1] Z-G Chen, K-J Du, Z-H Zhan, J Zhang, “Deadline constrained cloud computing resources scheduling for cost optimization based on dynamic objective genetic algorithm". *congress on evolutionary computation (CEC), IEEE*, pp 708–714, 2015.

[2] H-H Li, Y-W Fu, Z-H Zhan, J-J Li “Renumber strategy enhanced particle swarm optimization for cloud computing resource scheduling”. *congress on evolutionary computation (CEC), IEEE*, pp 870–876, 2015.

[3] F. Liu, Z. Ma, B. Wang, & W. Lin, “A virtual machine consolidation algorithm based on ant colony system and extreme learning machine for cloud data center”. *IEEE Access*, vol 8, pp. 53-67, 2019.

[4] B. M. Moura, G. B. Schneider, A. C. Yamin, H. Santos, R. H. Reiser, & B. Bedregal, “Interval-valued fuzzy logic approach for overloaded hosts in consolidation of virtual machines in cloud computing”. *Fuzzy Sets and Systems*, vol. 446, pp. 144-166, 2022.

[5] B. Shukla, “Modified novel algorithm based on learning automata to determine the underutilized physical machine (PM) in a data center” (Doctoral dissertation, Dublin, National College of Ireland), 2019.

[6] N.V. Donyagard, M.A.Ghobaei, & A. Souri, “Multiobjective virtual machine placement mechanisms using nature- inspired metaheuristic algorithms in cloud environments”: *A comprehensive review*. *International Journal of Communication Systems*, vol. 32, no. 14, p.e4068, 2019.

[7] S. Azizi, M.H. Zandsalimi, & D. Li, “An energy-efficient algorithm for virtual machine placement optimization in cloud data centers. Cluster Computing”, vol. 23, no. 4, pp. 3421-3434, 2020.

[8] N.S. Venkata, & V.S,S. Shankar, “An Effective Secured Dynamic Network-Aware Multi-Objective Cuckoo Search Optimization for Live VM Migration in Sustainable Data Centers Sustainability”, vol. 14, no. 20, p. 13670, 2022.

[9] F.N.A. Wesabi, M. Obayya, M.A. Hamza, J.S. Alzahrani, D. Gupta, S. Kumar, “Energy Aware Resource Optimization Using Unified Metaheuristic Optimization Algorithm Allocation for Cloud Computing Environment. Sustain”. *Comput. Inform. Syst.*, vol. 35, p. 10068, 2022.

[10] N. Ai, B. Wu, B. Li, Z. Zhao, “5G Heterogeneous Network Selection and Resource Allocation Optimization Based on Cuckoo Search Algorithm”. *Comput. Commun*., vol. 168, pp. 170–177, 2021.

[11] M. Mareli, and B. Twala, “An Adaptive Cuckoo Search Algorithm for Optimisation”. *Appl. Comput. Inform.*, vol. 14, pp. 107–115, 2018.

[12] H.M. Naeen, E. Zeinali, A.T. Haghighat, “A stochastic process-based server consolidation approach for dynamic workloads in cloud data centers”. *J Supercomputing*, 2018.

[13] M.H. Sayadnavard, A.T. Haghighat, A.M. Rahmani, “A reliable energy-aware approach for dynamic virtual machine consolidation in cloud data centers”. *J Supercomputing.*, vol. 75,pp. 2126–2147, 2019.

[14] Y. Gao, H. Guan, Z. Qi, B. Wang, “An ant colony system algorithm for the problem of server consolidation in virtualized data centers”. *J Comput Inf Syst.*, vol. 8, no. 16, pp. 6631–6640, 2012.

[15] T. Deepika, and P. Prakash, “Power consumption prediction in cloud data center using machine learning”. *Int J Electr Comput Eng (IJECE).*, vol. 10, no. 2, pp. 1524–1532, 2020.

[16] E. Barlaskar, Y.J. Singh, & B. Issac, “Enhanced cuckoo search algorithm for virtual machine placement in cloud data centres”. *International Journal of Grid and Utility Computing.*, vol. 9, no. 1, pp. 1-17, 2018.

[17] H.O. Salami, A. Bala, S.M. Sait & I. Ismail, “An energy-efficient cuckoo search algorithm for virtual machine placement in cloud computing data centers”. *The Journal of Supercomputing*., vol. 77, no. 11, pp. 13330-13357, 2021.

[18] S. Mangalampalli, K.S. Pokkuluri, G.N. Satish, & S.K. Swain, “Effective VM Placement Mechanism in Cloud Computing using Cuckoo Search Optimization”. *International Conference on Computing, Communication and Power Technology (IC3P) IEEE.*, pp. 238-241, 2022 January.

[19] J.P. Gabhane, S. Pathak, & N.M. Thakare, “Metaheuristics Algorithms for Virtual Machine Placement in Cloud Computing Environments—A Review”. *Computer Networks, Big Data and IoT*, pp. 329-349, 2021.

[20] M. Ghetas, “A multi-objective Monarch Butterfly Algorithm for virtual machine placement in cloud computing”. *Neural*

*Computing and Applications.*, vol. 33, no. 17, pp. 11011-11025, 2021.

[21] M.S.A. Khan, & R. Santhosh, "Hybrid Optimization Algorithm for VM Migration in Cloud Computing. Computers and Electrical Engineering", vol. 102, pp. 108152, 2022.

[22] S. Rahmani, V. Khajehvand, & M. Torabian, "Burstiness-aware virtual machine placement in cloud computing systems". *The Journal of Supercomputing.*, vol. 76, no. 1, pp. 362-387, 2020.