Fuzzy-epo Optimization Technique for Optimised Resource Allocation and Minimum Energy Consumption With the Brownout Algorithm.

Jitendra Kumar Samariya
NIT Jalandhar: Dr BR Ambedkar National Institute of Technology
Rajeev Tiwari (errajeev.tiwari@gmail.com)
University of Petroleum and Energy Studies https://orcid.org/0000-0002-8245-4748
Mohammad S. Obaidat
University of Sharjah
Gaurav Bathla
UPES: University of Petroleum and Energy Studies

Research Article

Keywords: Cloud Computing, Quality of Service, Virtual Machine Allocation, Virtual Machine Migration, Fuzzy- EPO

Posted Date: July 1st, 2022

DOI: https://doi.org/10.21203/rs.3.rs-945687/v1

FUZZY-EPO OPTIMIZATION TECHNIQUE FOROPTIMISED RESOURCE ALLOCATION AND MINIMUM ENERGY CONSUMPTION WITH THE BROWNOUT ALGORITHM.

Jitendra Kumar Samriya[1], Rajeev Tiwari[2] *,Mohammad S. Obaidat[3], Gourav Bathla[4]

[1] Dr B R Ambedkar National Institute of Technology, Jalandhar,

[3] Fellow of IEEE and Fellow of SCS Founding Dean and Professor, College of Computing and Informatics
University of Sharjah, Sharjah 27272, UAE,
with King Abdullah II School of Information Technology,
University of Jordan,
Amman 11942, Jordan and with University of Science and Technology Beijing 100083, China msobaidat@gmail.com or m.s.obaidat@ieee.org ,

[2.4] University of Petroleum and Energy Studies, Uttarakhand

Correspondence

[2] **rajeev.tiwari@ddn.upes.ac.in*

Abstract

Cloud Computing is an eminent and reputable agenda which relies on large-scale distributed processing to provide access to their resources and services. In the cloud environment a rigorous management system is mandatory to collect all information regarding task processing levels and proving impartial resource provisioning through the levels of Quality of Service (QoS). These concerns can be settled by employing a meta-heuristic optimization-based resource management. Subsequently, this paper presents a Fuzzy Emperor Penguin Optimization (Fuzzy-EPO) algorithm-based resource provisioning framework for heterogeneous cloud environment. To deploy the optimal set of virtual machines (VM) to physical machines the VM allocation model is employed. The proposed Fuzzy-EPO algorithm does the VM consolidation mainly to reallocate overloaded VM to under-loaded PM to minimize the migration time and the brownout mechanism is adopted to reduce the rate of energy consumption. CloudSim simulation platform is used to implement the proposed system. The simulation results expose that the proposed Fuzzy -EPO based system is effective in restraining the proportion of SLA violation and increasing QoS requirements for providing proficient cloud service.

Keywords*:* Cloud Computing, Quality of Service, Virtual Machine Allocation, Virtual Machine Migration, Fuzzy- EPO.

1. INTRODUCTION

In rendering IT services the cloud computing is becoming a principal approach by minimizing the cost for the consumers. Cloud computing offers a secure computing paradigm for scientific application. Virtualization as a service offers On-demand facility of virtualized resources using cloud computing without any delay [1]. In cloud computing a vital task is regarding the resource allocation to user requests. To allocate resources in the cloud framework the virtualization is generally used. Each task is mapped to the virtual machines (VMs) using a tasks scheduler to minimize a given cost function. In cloud the, problem of tasks scheduling and resource allocation on existing VMs is a NP-hard problem [2,3]. For this problem the search space is vast such that if an approach successively inspects this space and discovers the optimal solution, it needs exponential time. Hence, this problem can be handled using heuristic and intelligent approaches. The resource deficiency is the vital issue in cloud computing. As a result, reducing the make span and enlarging resources utilization concurrently is an important goal [4,5]. Preventing resource overloading and balancing load on them can surge the resource utilization and thus increase cloud throughput.

In contrast, average response time is used by the service level agreement (SLA) for cloud environment. Hence, in a reliable scheduling approach, the makespan must be minimum with a determined load balancing [6-9]. The use of unsuitable scheduling algorithm could make the system perform worse. In cloud computing environments numerous algorithms and techniques for resource scheduling are accessible. In certain systems, for example, classical algorithms are employed [10-12]. The meta-heuristics approaches as per the literature survey are inspired by nature; these are considered to be the dominant methods which could handle the resource scheduling problems in an efficient way [13–15]. One of the types of a prevailing natural phenomena inspired approach is the swarm intelligence. These approaches are iterative, stochastic and population-based. Several real life concerns such as wireless sensor networks (WSNs) localization issues robot path planning, placement of drone, image processing, portfolio optimization, computer assisted diagnostic systems and lot more have been effectively addressed using swarm intelligence. Along with abundant applications, with the goal of accomplishing the optimal results the swarm intelligence algorithms have been frequently modified, parallelized and hybridized. Resource management as an objective is focused by the existing resource scheduling policies in view of workload management, cost reduction and performance. The present scheduling techniques due to the rising complication of applications domain are incapable of delivering and exploiting cloud resources. Therefore, an optimization technique is needed to deal with such uncertainty situation [16, 17]. Heuristic/ meta-heuristic approaches have been followed by the present resource scheduling techniques for scheduling and identifying the virtual machine (VM) [18, 19]. Optimization process has been endorsed by the Meta-heuristic techniques which can capably deal with distributed, dynamic, and erratic demand, and based on the situation capable to manage VMs. This research work motivation starts from conflicts to attain the most reliable resource search for an arrangement to task requirements [20-22]. Allocation of resources to task is done by an efficient resource allocation algorithm in a way that increases the data centers energy efficiency and reduces the makespan.

This study presents the Fuzzy-Emperor Penguin Optimizer (EPO) for optimizing the resource allocation in the cloud environment. Fuzzy logic is employed in this work to surge the accuracy of the EPO algorithm. The authors have adopted the brownout technique to decrease energy consumption at the components level, which moreover yields profits for the CSP. In a given viable solution range the proposed approach uses fewer running time and has robust global exploration ability. Moreover the proposed approach is well balanced and enhanced in the exploitation and exploration phases. The proposed approach performs well in resource allocation when compared with the existing models. The proposed methodology is validated and the simulation is conducted using the CloudSim toolkit. The outcomes reveal that the Fuzzy-EPO has massive ability as it provides substantial cost savings and feasible for resource allocation; moreover the use of brownout has improved the energy efficiency and can fulfill the customers the SLA requested.

The remaining of this paper is organized like this: After the introduction section in 1, sec 2 describes the works allied with resource allocation and scheduling. Sect. 3 presents the problem statement. Sec 4 discusses the system architecture. Sec 5 deals with the EPO optimizer. In Sect. 6, Resource allocation optimization models with the Brownout Enabled Algorithm for Energy Efficiency are described. The experiments and analysis are discussed in Sect. 7, the conclusion and future scope is presented in Sect. 8.

2. LITERATURE REVIEW

A wide range of methods in cloud computing have been researched for resource scheduling. For dissimilar objectives various resource scheduling approaches were presented. These methods improve the efficacy of task processing and resource usage to a certain range. Given below are some of the important recent contributions discussed in detail.

Only for specific domain the heuristic algorithms provide good solution and they are mostly problem dependent. High communication and computation cost incur during the scheduling of applications. The scheduling algorithms main objective is to minimize the time for execution deprived of the grid computing cost, however cloud has assorted resources and every resource has distinct prices (hourly/monthly) depending on the computational ability. At varying prices, the resources are provided to the user. For future applications in heterogeneous cloud, several scheduling plans may acquire the dissimilar values of makespan time, processing time and cost. To gratify both factors (ie time and cost), the scheduling problem is required in the cloud framework. To maintain the incoming requests factors and the user requirements the authors Tang et al. 2019) [23] suggested the edge computing. To achieve the things this model has to face several challenges. Matching and scheduling algorithms are present in the dynamic resource allocation algorithm which advances the QoS in cloud environment.

The prominence of energy efficiency is clarified by Zhang et al. (2019) [24], who considered heterogeneous cloud environments. For migration and resource allocation, Task deadlines are very important in VMs which devours more energy. Unbalanced resource utilization is caused due to this excess consumption of energy; considering this factor an energy-efficient resource ranking approach was presented by authors Mekala and Viswanathan (2019) [25] for utilizing the resource in VMs. The authors Wang et al. (2019)[26] minimized the cost in view of the completion time without affecting user experience. It has been accomplished using distributed algorithm. Moreover the queue delay is minimized. To minimize total power consumption authors in [27] proposed a network based optimization methods. The outcomes reveal that the power depletion has been decreased by 53.68%. When compared with other algorithms the proposed method reduced the consumption of power by 30.3% with least root mean square error. Using deep reinforcement learning, a resource allocation model has been presented by K Kartiban et al 2020 [28] in a green computing. An effective resource allocation is attained by this proposed approach.

TABLE1: Related Work Summary

References	Mechanism/Technique	Advantage	Limitations
Tang et al. (2019)[23]	Dynamic resource algorithm	Less bandwidth and latency	In this work the tabu search algorithm used has high number of iterations
Wang et al 2019[24]	Resource scheduling and Task offloading mechanism	Queue delay and Optimal task offloading rate	Task deployment policies is not considered
Mekala and Viswanathan (2019)[25]	Energy-efficient resource ranking	Reduces VM migrations and Surges the resource utilization rate and	In an effective way the resource balance could be made
Zhang et al. (2019)[26]	VM allocation using Effective evolutionary approach	attains better energy efficiency	For reserved cloud service requests, the energy efficiency could be increased
P Akki presented et al., 2020[27]	Neural network based optimization methods	minimum root mean square error based minimized power consumption	An enhanced should be made in the execution time
K Kartiban et al 2020[28]	Deep reinforcement learning model Based resource allocation	a better resource allocation model is achieved	overall efficiency of the resource allocation might be improved
J Pravenchande et al 2020[29]	Optimal power minimization and improved task scheduling approach	Efficient resource allocation	Task scheduling could have obtained much more better results
Z Peng et al 2020[30]	Cloud resource management scheduling using Deep Q-network	Generate an efficient and reasonable resource allocation and scheduling strategy	Lot of problems in the resource management should be addressed by the proposed model
V Ropa et al 2020[31]	Energy and Power VMs VM management using Dynamic Migration (EPADM) model in cloud framework	Reduced SLA violations ,energy-power efficiency under distinct workload cases	various load conditions are not considered

For an efficient dynamic resource allocation process an improved task scheduling has been proposed by Pravenchande et al 2020[29] with an optimal power minimization approach. Using the prediction mechanism and dynamic resource table updating algorithm the efficiency relating to response time and task completion is achieved. This outline reduces the power consumption and gets optimal power reduction in data centers. In the cloud computing services, a detailed research is done by Peng et al 2020[30] on resource scheduling. In this paper a scheduling framework is proposed using a Deep Q-network which deals with the resource scheduling concerns in complex cloud environment. A Deep reinforcement learning strategies is used next for resource optimization scheduling. Finally by fine-tuning the weights of different optimization objectives, a trade-off is made between energy consumption and makespan of this frame. Like this, an adjustment can be made dynamically to meet different requirements. Energy and Power Aware Dynamic Migration (EPADM) model was proposed in cloud paradigm by Ropa et al 2020 [31] for managing the VMs dynamically using reduced SLA-Violations with energy and power efficiency.This model focuses on efficient cloud resource management which has three phases i.e., VM relocation, VM placement and VM consolidation. A summary is given in Table 1. The existing methods in cloud framework reveal that

suitable resource allocation affects the performance factors directly and indirectly.

3 PROBLEM STATEMENTS

The most vital issues in cloud computing is scheduling of tasks on the VMs, which has been stated in the introduction section. An eminent scheduling in cloud represents minimized cost and prevents the resource overloading and to uplift the systems efficiency. The major requirement of the cloud user is to minimize the tasks execution time and make span. The proposed algorithm with the intension to fulfill the interest of both, Thus it should consider minimizing the tasks execution time and the average make span as well as boosting the resource utilization of VMs for better efficiency of cloud resources. In cloud, resource allocation and Task scheduling plays a major role in improving the efficiency of cloud. For task scheduling, EPO is employed to offer optimization. Some of the parameters such as Memory, CPU and other resources requirements are used to describe the requests of resources. A request is satisfied by the cloud provider by representing the virtual resources to physical ones. On demand basis the resources are allocated to tasks. At a time numerous tasks can be accomplish by a VM, but two VM’s at a time cannot process the same task. EPO is used to allocate VM’s to tasks and to improve the performance of EPO, the fuzzy logic is employed. In order to maximize fitness value and minimize the function value of masses, a combination of Mean Flow Time and Make span along with Load imbalance as cost function is used with EPO, which can offer better task allocations during task scheduling. Along with it brownout controllers used can offer activation and deactivation of load on VMs and thus can offer better energy efficiency on cloud servers.

4 SYSTEM ARCHITECTURE

In the cloud environment, the resources are assigned by the system architecture which is normally considered as provisioning plan in datacenters. The complexities in resource allocation can be solved using many approaches such as allocation of resource can be done manually, in a dynamic way or combination of both. The proposed system architecture is illustrated in Fig 1.

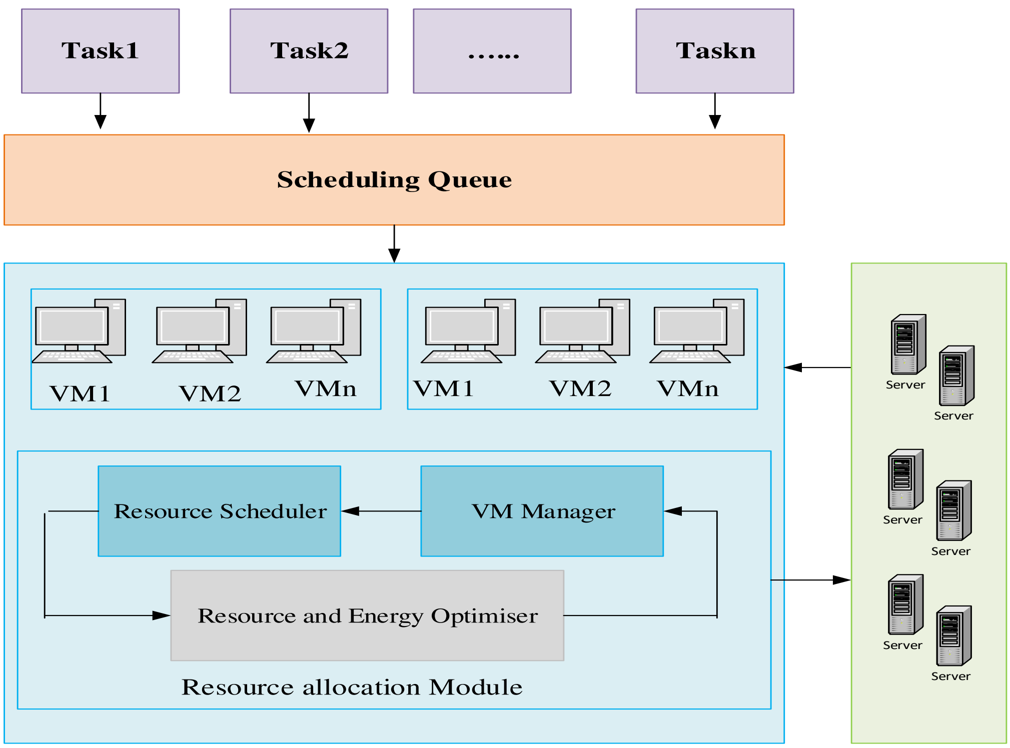


FIGURE 1. System architecture of the proposed model

This architecture represents how the user request is handled and the request is forwarded to the Cloud broker and finally to Cloud datacenter. Initially the Cloud provider accepts all the user requests, then the request is processed by the Cloud broker based on the user’s requirement and resource availability. The request is forwarded by the broker to the resource management of the datacenter which has the resource scheduler and VM manager. Moreover, the resource management modules mentioned above will validate whether the requirement such as availability and on demand service will be supported by the

VM. The Cloud resource manage determines the user request based on the systems readiness. The aforementioned issued are handled using the Fuzzy EPO which moreover deals with the resource utilization and reduces the consumption of energy

using Brownout approach. The subsequent sections discuss about the resource allocation models that are used in attaining the energy-efficiency in the datacenter.

5 EMPEROR PENGUIN OPTIMIZER (EPO)

5.1. Inspiration

Emperor penguins do several activities such as foraging in groups and hunting mainly for living and they are considered as social animals. During extreme winters these penguins perform huddling to survive. During huddling, each penguin gives a mutual sense of unity and ensemble in their social behavior.

Summarized below is the huddling behavior of the penguins:

- Determine and produce the huddling domain.
- Estimate the temperature.
- Analyze each penguins distance.
- Relocate the validated mover.

Modeling is done mainly to identify the actual mover. The huddle shape is denoted as L-shape plane of polygon. When the effective is discovered then the huddle boundary is computed once again.

5.2. Create and analyze the huddle boundary

The first thing needed for mapping the emperor penguins behavior is huddling. During huddling at least two penguins are approached. The wind flow speed and direction defines the huddling boundary. The huddling boundary can be mathematically expressed as shown below:

The gradient of η is denoted as χ and the wind velocity is denoted as η.

$$x = \nabla \eta \tag{1}$$

To attain a complex potential, the vector α is integrated with η.

$$G = \eta + i\alpha \tag{2}$$

The function of polygon plane is represented as G and the imaginary constant is represented as 'i'.

5.3. Temperature profile

To conserve their energy, the huddling action is done by the Emperor penguins which surges the temperature of huddle, T = 1 if X< 1, T = 0 if X> 1; the polygon radius is represented as X.

The temperature is computed, as:

$$T' = \left(T - \frac{Max_{itr}}{y - Max_{itr}} \right) \tag{3}$$

$$T = \begin{cases} 0, & if\ X > 1 \\ 1, & if\ X < 1 \end{cases}$$

The time needed to find the optimal solutions is represented as T , the maximum number of iterations is denoted as Max_{itr}, the current iteration is represented as y and X is the radius.

5.4 Calculate the distance

The emperor penguins distance is computed once the huddling boundary is discovered. The highest fitness value solution is considered as the best solution. With respect to the best optimal solution, the positions of the search agents are updated.

The mathematical depiction of position updating process is as follows:

$$\vec{N}_{ep} = Ebs\left(N(\vec{E}).\vec{Q}(y) - \vec{E}.Q_{ep}(y)\right) \tag{4}$$

Here, the distance is denoted as $\vec{N}_{ep}$, the collision among the penguins are avoided by $\vec{X}$ and $\vec{E}$. The optimal solution is represented as $\vec{Q}$, the running iteration is represented by y, and the emperor penguins position vector is denoted as $\vec{Q}ep$. The social forces are represented as N(), which aids to find best optimal solution. The Eqns below compute the vectors $\vec{Y}$ and $\vec{E}$.

$$\vec{Y} = \left(N \times (T' + R_{grid}(Acc)\right) \times Rand()) - T' \tag{5}$$

$$R_{grid}(Acc) = Abs\left(\vec{Q} - \vec{Q}_{ep}\right) \tag{6}$$

$$\vec{C} = Rand() \tag{7}$$

The movement parameter is defined as N, which sustains a gap for evading the collision among the search agents. The polygon grid

accuracy is denoted as $Pgrid\ (Acc)$, random number is denoted as Rand () which lies in the range of [0, 1]. The parameter N is set to $2.T'$

The function N () is computed as given below:

$$N(\vec{E}) = \left(\sqrt{f.e^{-x/l} - e^{-x}}\right)^2 \tag{8}$$

The expression function is denoted as e and the control parameters are denoted as $f$and l. In the range of [2, 3] and [1.5, 2], are the values of f and l lie.

5.5. Relocation

The emperor penguins position is updated by utilizing the mover. Considering the best solution in a search space the movement of other search is led. The subsequent equations are utilized to find the next position of emperor penguin:

$$\vec{Q}_{ep}(y+1) = \vec{Q}(y) - \vec{X}.\vec{N}_{ep} \tag{9}$$

Emperor penguin updated position is represented as $\vec{Q}\ (y+1)$.

6 RESOURCE ALLOCATION OPTIMIZATION MODELS

This discussion can be simplified by assuming a set of tasks is available and each task has many subtasks. In order to process each subtask any available resource is endorsed. A specified level of capacity is enclosed in the cloud resources (e.g., memory, CPU, storage, network). The application of Fuzzy- EPO for resource allocation process is stated below:

Inputs: Let the set of '*m'* available resources be R = (R_1, R_2, …, R_j, …, R_m) processing '*n'* independent tasks represented by the set T = (T_1, T_2, …, T_i, …, T_n), i = 1, 2, …, n, j = 1, 2, …, m. The resources are parallel and unrelated on the available subset of resources Rj∈ R in which the task Ti can be executed.

Outputs: It is an effectual and active resource allocation structure which encloses the tasks scheduling to suitable resources.

Limitations: Every task execution time is based on the actual state, and in an advance manner the value cannot be placed. Once started, without any interruption the task must be completed.

Objectives: To attain an energy-efficient scheduling through improving the data centers energy efficiency and minimizing the makespan. In this work a resource allocation optimization model is designed which could incorporate the factors of makespan and energy-efficient optimization.

6.1 Brownout Enabled Algorithm for Energy Efficiency

The brownout enabled energy efficient approach (EEBA) is based on the placement and consolidation (PCO) algorithm [32]. The brownout controller based on power consumption of the host effectively activates or deactivates the applications. There are 6 steps in the Algorithm 1, EEBA mainly consisting of: The PCO and overloaded power threshold algorithm initializes the VM placement (lines 1-2). In order to test whether a host is overloaded or not, the power threshold value is deemed.

Below are the nest steps:

1) Checking all the hosts in each time interval *t*; the total overloaded host can be counted as n_t (line 4);

2) Change dimmer value $\theta_t \sqrt{\frac{n_t}{m}}$ as per the host size M and several overloaded hosts n_t (line 6). To regulate the degree of power consumption, the dimmer value dimmer θ_t acts as a control knob at time t. All the hosts are overloaded at time t with the dimmer value θ_t it means that brownout controls components are on the hosts. The dimmer value exposes the dimmer adjustment approach.

3) Compute the estimated reduction for the utilization on the overloaded hosts (lines 8-10). The EEBA based on the host power model and the dimmer value calculates the expected power reduction of the host P_i^r (line 9) and utilization drop u_{hi}^r (line 10).

4) Then the estimated utilization reduction on VM is computed (lines 11-13). Initializing a bare and disabled component list dcl_{ijt} of VM_j on host h_i is to formulate the components that are deactivated (line 12). Based on VM utilization model the VM reduction $u_V^r M_{ij}$ is processed and the VM utilization multiplies u_{hi}^r (line 13).

5) To find and disable components list dcl_{ijt} the component selection policy CSP is applied (line 14). The component selection

policy as per the expected VM utilization reduction $u_V^r M_{ij}$, is liable for obtaining the components filling the utilization limit. Then deactivating these components and their connected ones, and updating total discount amount (line 15).

Algorithm 1 Brownout Algorithm for Energy Efficiency

Input: Host List Hl with size *M*, information of application modules, dimmer value θ_t at time t, overloaded power threshold TP, deactivated list of components dcl_{ijt} Of VMij on host hi, scheduling interval T, power model of host HPM, component selection policy CSP, and VM utilization model V UM.

Output: No of shutting down hosts, discount amount, total energy consumption

1: To initialize VMs placement, use PCO algorithm.

2: using the inputs like T P initialize the parameters

3: for $t \leftarrow 0 \text{ to T do}$

4: $n_t \leftarrow COHI(\text{hl})$

5: $if\ \mathrm{n_t} > 0\ then$

6: $\theta_t \leftarrow= \sqrt{\frac{nt}{M}}$

7: for all h_i in $\mathrm{h_1}$ do

8: $if\ (\mathrm{P_i^{server}} > P_i^{\max} \times TP) then$

9: $P_r^i \leftarrow \theta_t \times Ph_i$

10: $u_{hi}^r \leftarrow HPM\left(h_i, P_i^r\right)$

11: for all $VM_{ij}\ on\ \mathrm{h_i}$ do

12: $dcl_{ijt} \leftarrow NULL$

13: $u_v^r M_{ij} \leftarrow VUM(u_{hi}^r, VM_{i,j})$

14: $dcl_{ijt} \leftarrow CSP(u_v^r M_{ij})$

15: $D_i \leftarrow D_i + d\left(VM_{ij}\right)$

16: end for

17: end if

18: end for

19: else

20: stimulate components which are deactivated

21: end if

22: top to optimize VM placement use VM consolidation in PCO algorithm

23: end for

6.2 Resource allocation using the proposed EPO enhanced by Fuzzy

The deemed NP-hard problem in cloud is the scheduling of tasks on existing VMs. The number of VMs is represented as VM_{num} and the number of tasks as $Task_{num,}$ then the estimated allocations is $VM_{num}Task_{num}$. The scheduler finds a VM to allocate the task to ensure that the memory requirements and the computational of all tasks are fulfilled in a least expensive way. In several ways the cost could be designed.

Some of the vital costs are:

1. Mean (Flow-Time): Sum of all tasks execution time
2. Make span: Time to complete the final task
3. Load imbalance

These parameters are combined to represent the Cost. Due to the size of the search space resources allocation and scheduling is a NP-Hard problem in cloud. As the EPO algorithm has achieved enriched performance in nonlinear problems, this algorithm is employed for tasks scheduling. Initially a sequence of tasks is created to code problem responses which are allocated to each VM's and later the sequences created are combined and stored in an array. The number of tasks ($Task_{num}$) is equal to the length of this array; in each machine the allocated task numbers are stored. The allocation cost is computed using Eq. (10):

$$cost = \alpha \times \left(\beta \times Makespan + (1-\beta) \times mean\_flowtime\right) + (1-\alpha) \times Load\_balance \quad (10)$$

From Eq. (11) the allocation fitness is computed. A better allocation is the one which has more fitness.

$$fitness = \frac{1}{cost} \quad (11)$$

The EPO algorithm is used after planning the allocations to find an optimal allocation. The best and the worst allocation at the time t, is attained from Eqs. (12) and (13), where problem responses are represented as N.

$$best(t) = \max_{j \in \{1,\dots N\}} fitness_j(t) \quad (12)$$

$$worst(t) = \min_{j \in \{1,\dots N\}} fitness_j(t) \quad (13)$$

According to the optimal positions obtained the penguins are updated using Eqn 14.

$$\vec{P}_{ep}(x+1) = \vec{p}(x) - \vec{A}.\vec{D}_{ep} \quad (14)$$

The updated position of the penguin iteration process is represented as $\vec{P}_{ep}(x+1)$, the emperor penguin huddling behavior is recomputed during the iteration process. The fuzzy logic is used to adjust and control the parameter '*k*'. The value of *k* is increased as the optimal solution does not change considerably so that more search agents can affect each others. This possibly conducts a mutation process and avoids the algorithm to be trapped in the local optimal. The k value is minimized to enhance the convergence speed and prevents the wasting of time of the algorithm. A two inputs and one output fuzzy inference system is used.

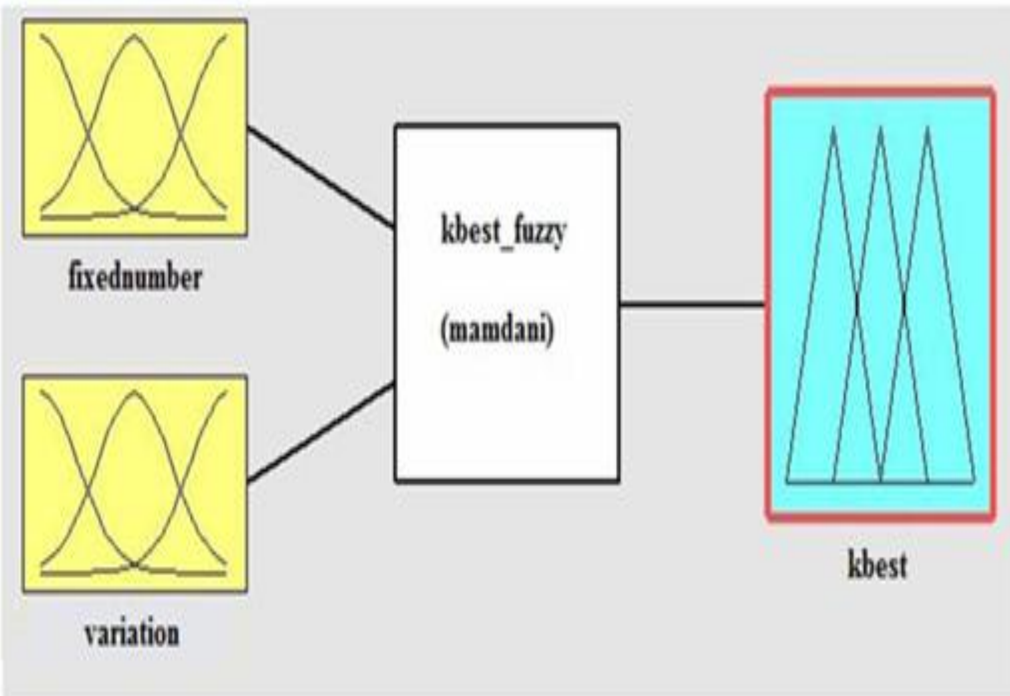


FIGURE 2 The fuzzy systems structure

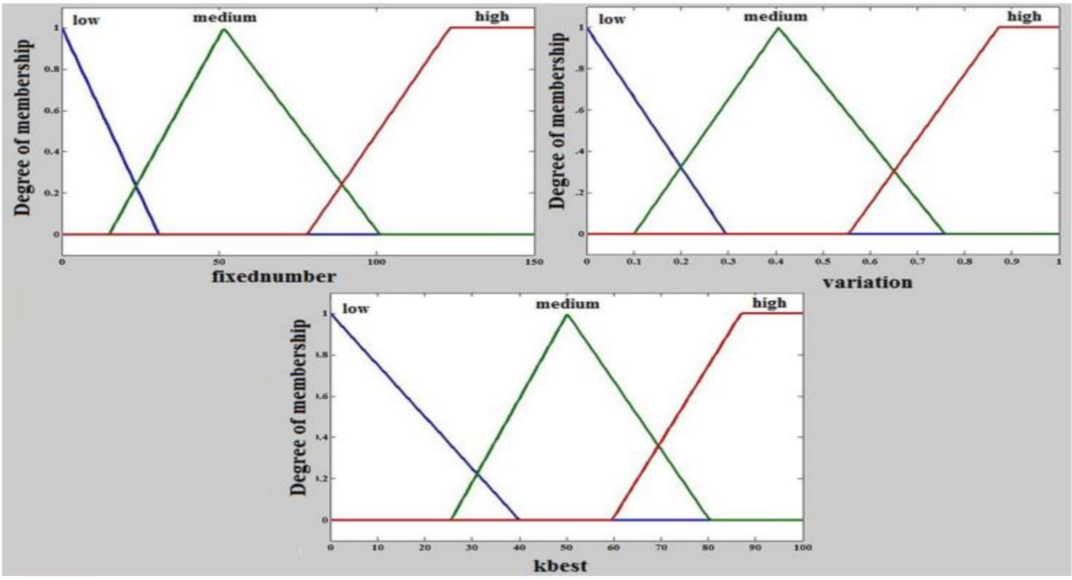


FIGURE 3 Membership functions of Fuzzy inputs and output

1.If (fixed number is low) and (variation is low) then K best is medium
2.If (fixed number is low) and (variation is medium) then K best is medium
3.If (fixed number is low) and (variation is high) then K best is low
4.If (fixed number is medium) and (variation is low) then K best is high
5.If (fixed number is medium) and (variation is medium) then K best is medium
6.If (fixed number is medium) and (variation is high) then K best is low
7.If (fixed number is high) and (variation is low) then K best is high
8.If (fixed number is high) and (variation is medium) then K best is medium
9.If (fixed number is high) and (variation is high) then K best is medium

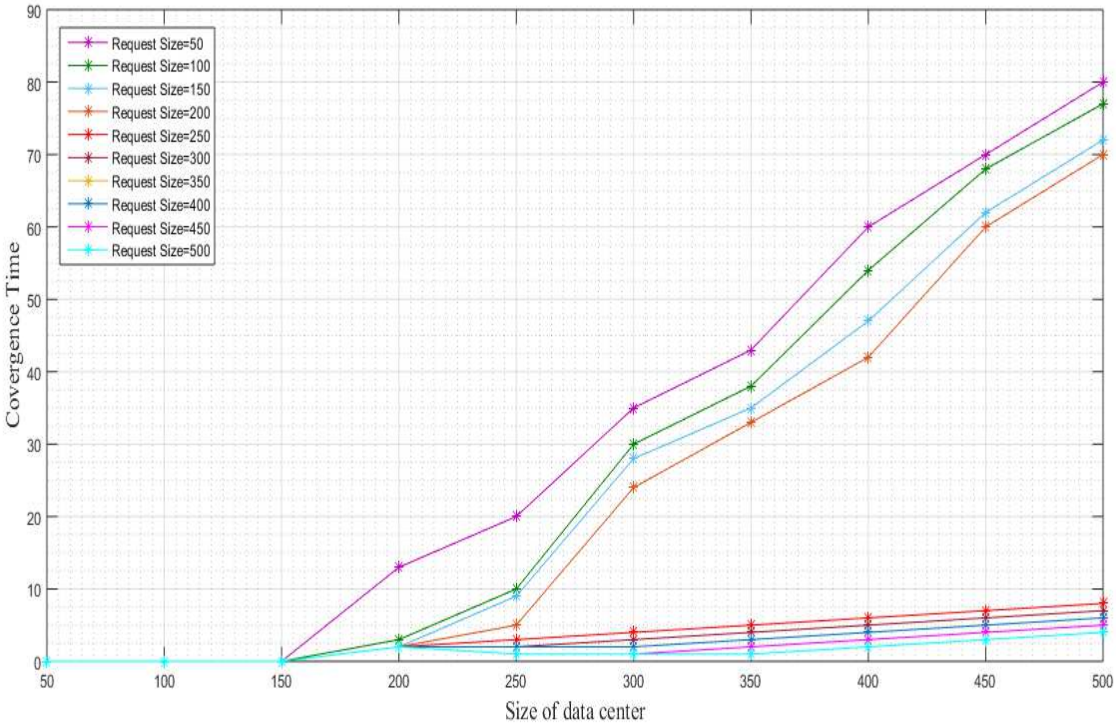
FIGURE 4 Fuzzy inference system rules

The proposed fuzzy system is illustrated in the figure 2. The inputs and output membership functions are depicted in Figure3. Figure 4 illustrates the fuzzy inference rules.

7 RESULT AND DISCUSSION

The simulation platform used to evaluate the proposed approach is CloudSim. The experimentation is conducted to validate resource allocation and energy consumptions. The complexity and scalability of the proposed model is provided by the numerical analysis relating to the arrival of the request, size of the data center and resource allocated metrics regarding the system load.

For experimentation in a data center, a total of 500 servers are modeled and the crucial performance indicators are collected along with the used server's ratio to attain the optimal solution. The performance of VM allocation is depicted in figure 5 with respect to the request load and data center size.



FIGIURE 5: The proposed methods execution time graph

Before the optimal placement the convergence time is considered as a function for data center, and the request ranges from 50 to 500. The proposed methods convergence time is better with the requests exceed 300 VMs.

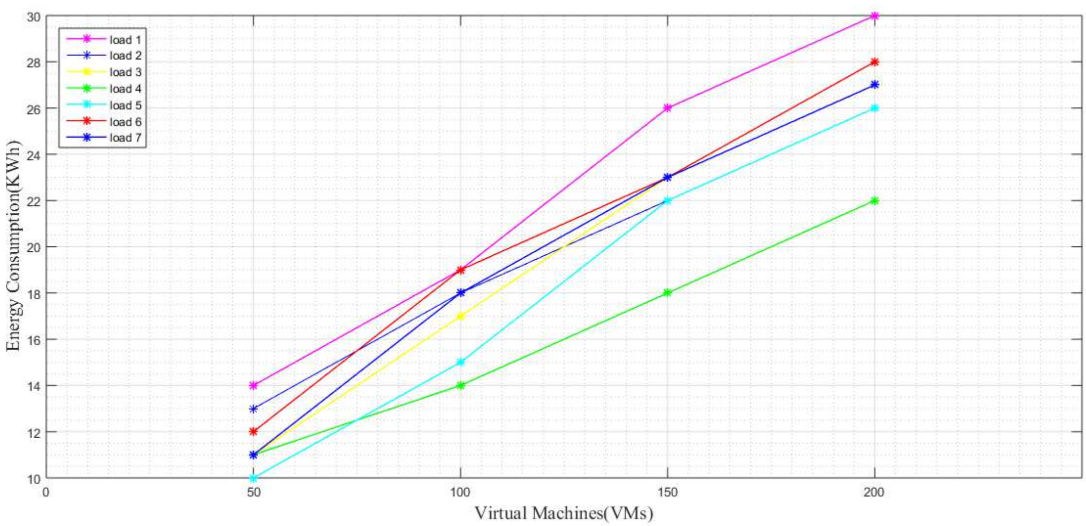


FIGURE 6: The proposed methods energy consumption graph

Fig.6 illustrates the consumption of energy from VMs for various loads. To validate the performance, seven different loads are used. When there is an increase in the number of VMs the consumption also increases. When the proposed approach is considered the consumption of energy is less than the usual consumption process. For some users the full access is not provided. The waiting time is considered here as a factor. For the resource allocation the waiting time may increase based on type of request.

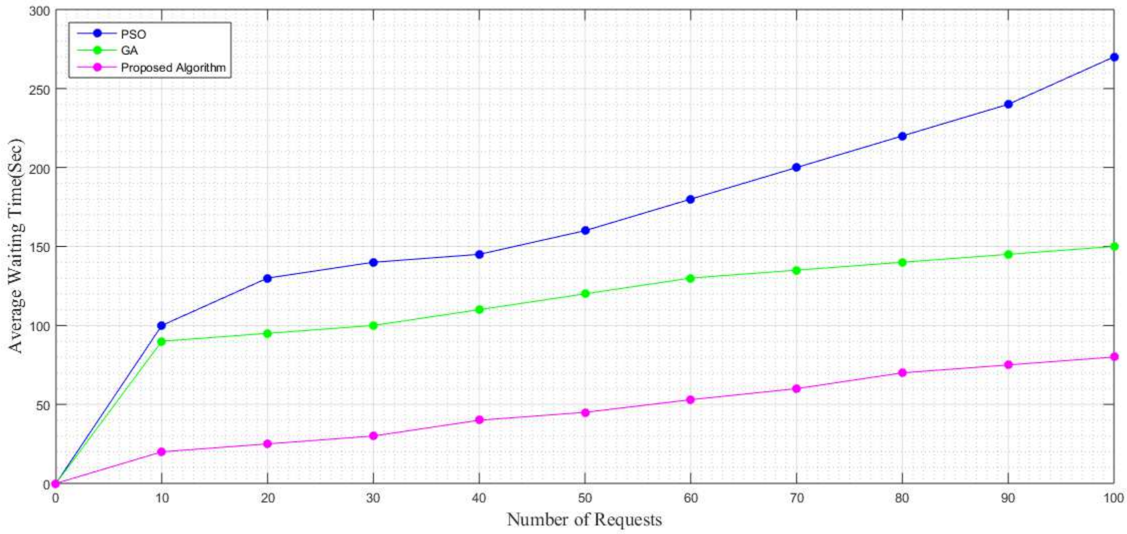


FIGURE 7 Comparison of average waiting time

The algorithm is designed to deal with the waiting time based on the policies. In Fig. 7 a comparison of average waiting time is made with the existing PSO and GA algorithms. The proposed method has upheld an average time response even with increasing time for all the requests. The waiting of GA reaches extreme of 145 s and more time is taken by the PSO takes than the other two.

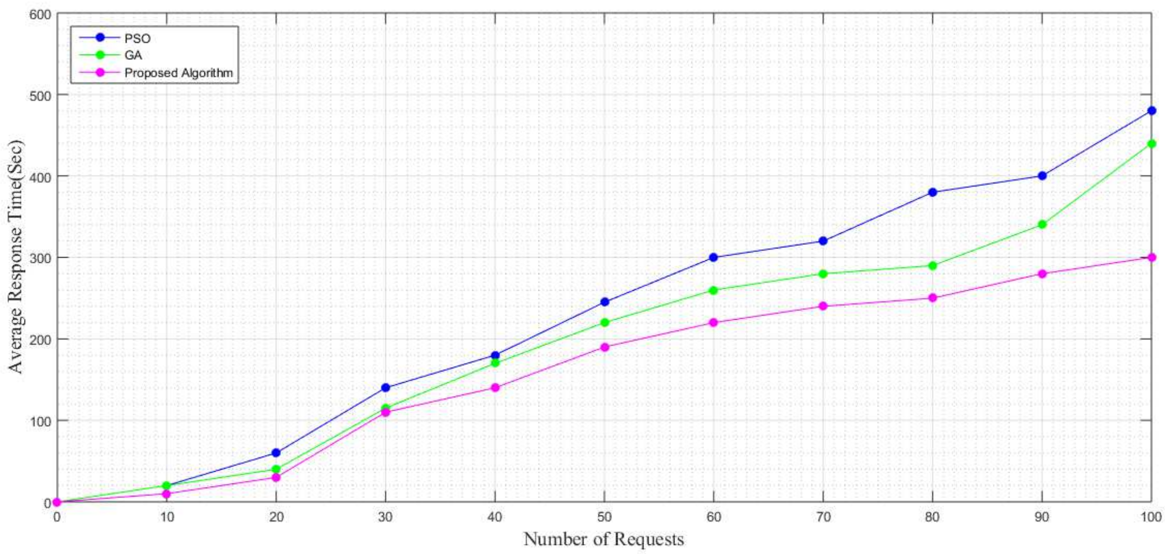


FIGURE 8 Comparisons of average response time

When there is a request to resource, the time needed to respond the user is the average response time. It is stated as the total service time's ratio to the entire tasks waiting time. A comparison for the response time is illustrated in figure 8 which reveals that the proposed model approach when compared to PSO and GA has the fast response time.

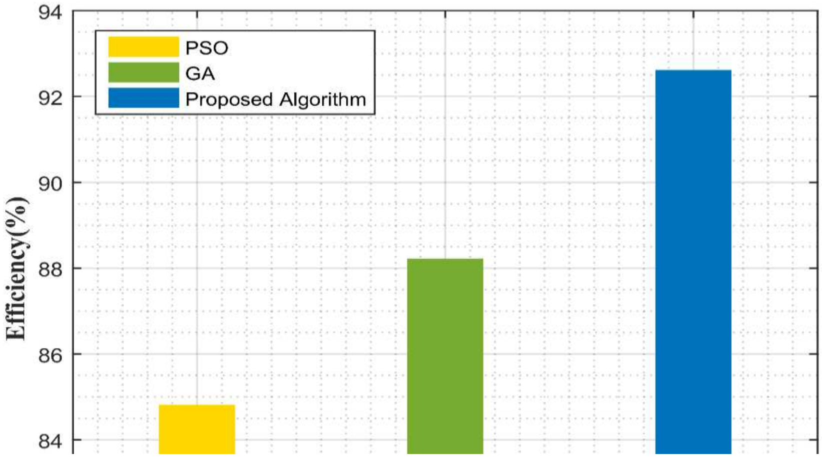


FIGURE 9 Comparison of efficiency

Fig.9 represents the overall efficiency of the proposed method compared with existing methods according to the average processing time ,average waiting time, number of responded requests, number of handled requests and energy consumption.

8 CONCLUSIONS

A resource management system based on Fuzzy EPO has been proposed in this paper for resource provisioning to the cloud customers. A brownout enabled algorithm has been adopted in this work to minimise the energy consumption based on PCO. This particular approach is presented considering component discount and utilization with a number of component selection policies. The main intention of the presented system is to fulfil the needs of the client through improving QoS parameters and minimising the energy consumption. In order to deploy an optimal set of VMs, the optimization is done at the process management level. The optimal set of VM is decided by the Fuzzy- EPO algorithm without triggering SLA violations based on different heterogeneous objectives. A rational resource provisioning options is provided by the proposed approach which selects the optimal set of VM Moreover cloud services can be accessed without SLA violation. As future work, the proposed algorithm can also be implemented in a real cloud system like Microsoft's Azure, Amazon's EC2 etc.

Funding: No funding support for this work
Conflicts of interest/Competing interests: We declare no any conflict of interest
***Availability of data and material: NA**
***Code availability : NA**

Reference

1. Lu, Y., & Sun, N. (2019). An effective task scheduling algorithm based on dynamic energy management and efficient resource utilization in green cloud computing environment. *Cluster Computing*, *22*(1), 513-520.
2. Samriya, J. K., & Kumar, N. (2020). A QoS Aware FTOPSIS-WOA based task scheduling algorithm with load balancing technique for the cloud computing environment. *Indian Journal of Science and Technology*, *13*(35), 3675-3684.
3. You, C., Zeng, Y., Zhang, R., & Huang, K. (2018). Asynchronous mobile-edge computation offloading: Energy-efficient resource management. *IEEE transactions on wireless communications*, *17*(11), 7590-7605.
4. Samriya, J. K., & Kumar, N. (2020). An optimal SLA based task scheduling aid of hybrid fuzzy TOPSIS-PSO algorithm in cloud environment. *Materials Today: Proceedings*.
5. Strumberger, I., Bacanin, N., Tuba, M., & Tuba, E. (2019). Resource scheduling in cloud computing based on a hybridized whale optimization algorithm. *Applied Sciences*, *9*(22), 4893.

6. Thein, T., Myo, M. M., Parvin, S., & Gawanmeh, A. (2020). Reinforcement learning based methodology for energy-efficient resource allocation in cloud data centers. *Journal of King Saud University-Computer and Information Sciences*, *32*(10), 1127-1139.
7. Shrimali, B., & Patel, H. (2020). Multi-objective optimization oriented policy for performance and energy efficient

resource allocation in Cloud environment. *Journal of King Saud University-Computer and Information Sciences*, *32*(7), 860-869.
8. Kumar, M., & Sharma, S. C. (2019). PSO-based novel resource scheduling technique to improve QoS parameters in cloud computing. *Neural Computing and Applications*, 1-24.
9. Zhong, Wei, Yi Zhuang, Jian Sun, and Jingjing Gu. "A load prediction model for cloud computing using PSO-based weighted wavelet support vector machine." *Applied Intelligence* 48, no. 11 (2018): 4072-4083.
10. Zhong, W., Zhuang, Y., Sun, J., & Gu, J. (2018). A load prediction model for cloud computing using PSO-based weighted wavelet support vector machine. *Applied Intelligence*, *48*(11), 4072-4083.
11. Samriya, J. K., & Kumar, N. (2020, October). Fuzzy Ant Bee Colony For Security And Resource Optimization In Cloud Computing. In *2020 5th International Conference on Computing, Communication and Security (ICCCS)* (pp. 1-5). IEEE.
12. Mustafa, S., Bilal, K., Malik, S. U. R., & Madani, S. A. (2018). SLA-aware energy efficient resource management for cloud environments. *IEEE Access*, *6*, 15004-15020.
13. Seethalakshmi, V., Govindasamy, V., & Akila, V. (2020). Hybrid gradient descent spider monkey optimization (HGDSMO) algorithm for efficient resource scheduling for big data processing in heterogenous environment. *Journal of Big Data*, *7*(1), 1-25.
14. Qie, X., Jin, S., & Yue, W. (2019). An energy-efficient strategy for virtual machine allocation over cloud data centers. *Journal of Network and Systems Management*, *27*(4), 860-882.
15. Devaraj, A. F. S., Elhoseny, M., Dhanasekaran, S., Lydia, E. L., & Shankar, K. (2020). Hybridization of firefly and improved multi-objective particle swarm optimization algorithm for energy efficient load balancing in cloud computing environments. *Journal of Parallel and Distributed Computing*, *142*, 36-45.
16. Saxena, D., & Singh, A. K. (2021). A proactive autoscaling and energy-efficient VM allocation framework using online multi-resource neural network for cloud data center. *Neurocomputing*, *426*, 248-264.
17. Harshitha, H. D., & Beena, B. M. (2017). Ant Colony Optimization for Efficient Resource Allocation in Cloud Computing. *International Journal on Recent and Innovation Trends in Computing and Communication*, *5*(6), 1232-1235.
18. Ebadifard, F., & Babamir, S. M. (2018). A PSO-based task scheduling algorithm improved using a load-balancing technique for the cloud computing environment. *Concurrency and Computation: Practice and Experience*, *30*(12), e4368
19. Reddy, V. D., Gangadharan, G. R., Rao, G. S. V. R. K., & Aiello, M. (2020). Energy-Efficient Resource Allocation in Data Centers Using a Hybrid Evolutionary Algorithm. In *Machine Learning for Intelligent Decision Science* (pp. 71-92). Springer, Singapore.
20. Al-Mahruqi, A. A. H., Morison, G., Stewart, B. G., & Athinarayanan, V. (2021). Hybrid heuristic algorithm for better energy optimization and resource utilization in cloud computing. *Wireless Personal Communications*, *118*(1), 43-73.
21. Miao, Z., Yong, P., Mei, Y., Quanjun, Y., & Xu, X. (2021). A discrete PSO-based static load balancing algorithm for distributed simulations in a cloud environment. *Future Generation Computer Systems*, *115*, 497-516.
22. Seethalakshmi, V., Govindasamy, V., & Akila, V. (2020). Hybrid gradient descent spider monkey optimization (HGDSMO) algorithm for efficient resource scheduling for big data processing in heterogenous environment. *Journal of Big Data*, *7*(1), 1-25.
23. Tang, H., Li, C., Bai, J., Tang, J., & Luo, Y. (2019). Dynamic resource allocation strategy for latency-critical and computation-intensive applications in cloud–edge environment. *Computer Communications*, *134*, 70-82.
24. Wang, Q., Guo, S., Liu, J., & Yang, Y. (2019). Energy-efficient computation offloading and resource allocation for delay-sensitive mobile edge computing. *Sustainable Computing: Informatics and Systems*, *21*, 154-164.
25. Mekala, M. S., & Viswanathan, P. (2019). Energy-efficient virtual machine selection based on resource ranking and utilization factor approach in cloud computing for IoT. *Computers & Electrical Engineering*, *73*, 227-244.
26. Zhang, X., Wu, T., Chen, M., Wei, T., Zhou, J., Hu, S., & Buyya, R. (2019). Energy-aware virtual machine allocation for cloud with resource reservation. *Journal of Systems and Software*, *147*, 147-161.
27. Akki, P., & Vijayarajan, V. (2020). Energy Efficient Resource Scheduling Using Optimization Based Neural Network in Mobile Cloud Computing. *Wireless Personal Communications*, *114*(2).
28. Karthiban, K., & Raj, J. S. (2020). An efficient green computing fair resource allocation in cloud computing using modified deep reinforcement learning algorithm. *Soft Computing*, 1-10.
29. Praveenchandar, J., & Tamilarasi, A. (2021). Dynamic resource allocation with optimized task scheduling and improved power management in cloud computing. *Journal of Ambient Intelligence and Humanized Computing*, *12*(3), 4147-4159.
30. Peng, Z., Lin, J., Cui, D., Li, Q., & He, J. (2020). A multi-objective trade-off framework for cloud resource scheduling based on the deep Q-network algorithm. *Cluster Computing*, 1-15.
31. Roopa, V., Malarvizhi, K., & Karthik, S. (2021). Efficient Resource Management on Cloud Using Energy and Power Aware Dynamic Migration (EPADM) of VMs. *Wireless Personal Communications*, *117*(4), 3327-3342.
32. Xu, M., Toosi, A. N., & Buyya, R. (2020). A Self-adaptive Approach for Managing Applications and Harnessing Renewable Energy for Sustainable Cloud Computing. *IEEE Transactions on Sustainable Computing*.