A MULTI-OBJECTIVE OPTIMIZATION APPROACH TO LOAD BALANCING AND TASK SCHEDULING IN IOT-FOG-CLOUD NETWORKS

M.Parveen Taj[1] and N.Muthumani[2]

[1]Research scholar, Department of Computer Science, PPG College of Arts and Science, Coimbatore- 641035, Tamil Nadu, India
[2]Principal, PPG College of Arts and Science, Coimbatore- 641035, Tamil Nadu, India

ABSTRACT

Although distributed systems can exist across multiple data centers, they require fog and cloud computing paradigms for data management in an Internet of Things (IoT) era. The integrated IoT, fog, and cloud (IoT-fog-cloud) method enables the processing of large amounts of IoT data in real-time. Alternatively, a lack of Load Balancing (LB) and improper handling of network resources can reduce the Quality of Service (QoS) under such circumstances. In real-time applications, increasing traffic to fog nodes causes delays and increases energy consumption. This problem was resolved by an effective LB algorithm. However, good resource utilization can be achieved whenever an effective LB is incorporated with Task Scheduling (TS). Hence, this article proposes a Multi-Objective Weight Optimized Task Scheduling and Load Balancing (MOWOTSLB) for IoT-fog-cloud systems. This research aims to effectively schedule workloads in a balanced manner, which conserves energy, enhances QoS, and reduces task execution time. This scheme employs a Horse Herd Optimization Algorithm (HOA) for TS. This HOA optimizes the scheduling of users' task requests to suitable computing resources according to the fitness function calculated using makespan, execution cost, and energy utilization. Though it improves resource utilization, some Physical Machines (PMs) are overloaded, and others are underloaded during uncertain fluctuations in workloads. This causes high energy and resource wastage in the data center. To solve this problem, the HOA is also adopted for Virtual Machine (VM) migration in this article. By assessing the fitness function of PMs such as load, migration cost, energy, and bandwidth use, the HOA chooses the finest VMs to drift to the appropriate PMs. This successfully strikes a balance between load distribution among PMs and energy usage. The simulation findings show that the MOWOTSLB outperforms current LBTS schemes in 0.95 throughput, 30 ms delay,100 ms response time, 175 kj energy consumption, 12.5 mb memory usage, and 900s lifetime.

KEYWORDS

IoT-fog-cloud systems, Load balancing, Task scheduling, Horse herd optimization & VM migration

1. INTRODUCTION

Since the emergence of IoT technology, several applications, like power grids, healthcare, automation, etc., have produced enormous amounts of data. This data from various systems needs to be effectively handled to satisfy QoS criteria such as low latency, high reliability, real-time responsiveness, and scalability [1-2]. Preserving the consistency of performance throughout the IoT infrastructure necessitates optimizing essential attributes such as energy usage, execution time, latency, execution cost, throughput, and resource usage [3]. Because it offers substantial memory and processing power to handle the enormous amount of data generated by IoT components, cloud computing has emerged as a key technology [4-5]. It is the perfect option for

business applications and services due to its adaptability, scalability, affordability, and user-friendliness. Users can use any computing capacity, storage, or services they require over the Internet at no additional cost while leveraging all available resources. However, latency problems with the conventional centralized cloud computing architecture might make it difficult to carry out time-sensitive tasks effectively [6-7].

Fog computing can help resolve some of the challenges faced by IoT applications functioning over the cloud. It links cloud servers with edge devices through a virtualized platform to enhance management and reduce communication lag. Distance reduction improves analytical speed for data processing locally and transmitting it to the cloud [8-9]. It minimizes network latency compared to cloud usage; however, it has limited storage, network, and computation resources [10]. A sustainable model for IoT applications may be developed to utilize fog and cloud resources by combining fog and cloud computing models [11]. Efficient allocation of workloads in IoT-fog-cloud environments has gained increased importance due to the rising number of IoT applications [12]. The fog or cloud layer assigns IoT programs to available computing resources using a well-designed workload scheduling strategy [13]. Many factors concerning QoS, like latency, power usage, processing time, resource demand, and throughput, need to be considered when assigning tasks to resources in a heterogeneous system [14]. Balancing the workload distributed among resources is critical to the effectiveness of IoT applications. Such management can be accomplished by closely monitoring each resource.

Nonetheless, inefficient allocation of workloads and system resources can lead to higher energy consumption and a lowered QoS. Due to the traffic congestion on fog gateways, there are high delays in delivering real-time services that are difficult to tolerate [15]. A system needs to be able to effectively manage the challenges that result from increased load, power consumption, and latencies produced by heavy data transfer and handling workloads as the proliferation of IoT devices continues. Many researchers studied about various effective LB algorithms. However, good resource utilization can be achieved whenever an effective LB is combined with TS in IoT-fog-cloud systems. Therefore, this study combines LB with the TS algorithm to effectively schedule workloads in a balanced manner, which saves energy, increases efficiency, and decreases the time needed to complete all tasks.

This manuscript proposes the MOWOTSLB scheme utilizing the HOA in IoT-fog-cloud computing. It includes an enhanced TS algorithm for allocating the task of the user to different computing resources. It can also reduce the cost and the execution time, as well as improve the resource utilization of the TS problem using the HOA. This HOA assists in introducing a multi-objective optimization for efficiently handling TS issues. The HOA optimizes the users' TS requests to suitable computing resources according to the fitness function calculated using makespan, execution cost, and energy utilization. Though it improves resource utilization, some PMs are overloaded and others are underloaded during uncertain fluctuations in workloads. This causes high energy and resource depletion in the information center. As a result, the VM migration is applied with the TS to effectively balance energy utilization and load fairness among PMs in each fog node. The HOA can also be applied to choose the VMs to drift to the appropriate PMs. Optimizing the migration of VMs involves taking into account the fitness function of every possible PMs; this finds the optimum mapping relationship between the chosen VMs and the suitable PM. Accordingly, the contribution of this study is as follows:

- To adopt and implement a novel optimization scheme called HOA for TS to assign the user's tasks to multiple computing resources.
- To select the optimal VMs to drift to the appropriate PMs using the HOA, resulting in reducing both energy and resource wastage.

- To reduce the cost and execution period while enhancing the resource utilization in IoT-fog-cloud setups efficiently.
- To demonstrate the efficiency of MOWOTSLB against conventional algorithms in terms of different metrics such as make span, energy utilization, resource usage, etc.

The below portions are arranged as follows: Section 2 discusses relevant works. Section 3 describes the MOWOTSLB algorithm. Section 4 assesses the simulation results. Section 5 settles the research and recommends further upgrades.

2. LITERATURE SURVEY

2.1. Task Scheduling

Safi et al. [16] presented a Multi-objective Grey Wolf Optimizer (MGWO) model to minimize the delay and energy utilization in the fog broker during TS. However, the main disadvantage was that it had to account for resource heterogeneity, which might affect resource use and load imbalance. Ghafari & Mansouri [17] designed Enhanced African Vultures Optimization Algorithm-based TS (E-AVOA-TS) for fog-cloud systems. The tasks were initially prioritized to manage the sensitivity of task latencies by the best-worst technique according to the number of tasks, deadline, and file volume. However, load imbalance can impact the resource usage. Liu et al. [18] suggested an optimal scheduling of IoT requests in an IoT-fog-cloud setting utilizing a mixture of Aquila Optimizer and AVOA (AO-AVOA). However, system throughput was low due to the uneven use of resources among nodes.

A Dynamic Multi-Criteria Scheduling (DMCS) technique was created by Bhakhar and Chhillar [19] to improve TS in fog-cloud computing. It ensured that time-sensitive jobs were completed quickly on fog nodes and resource-based tasks were controlled by cloud information centres by dynamically allocating tasks based on parameters including computational difficulty, urgency, and task dimension. Nevertheless, it has problems with scalability and processing overhead in bigger networks. By considering the location of the data storage, Khezri et al. [20] created a Data-Locality-aware Job Scheduling for IoT Fog-cloud (DLJSF) systems. Only the makespan criterion was taken for scheduling, whereas other factors like energy, cost, etc., were required to schedule the activities effectively.

Salehnia et al. [21] developed the Multi-Objective Moth-Flame Optimization (MOMFO) algorithm for TS, which minimizes task request completion time and increases throughput in fog-cloud-based IoT services. However, energy consumption has remained high. For the TS in heterogeneous cloud systems, Behera & Sobhanayak [22] created a hybrid Genetic Algorithm and GWO (GA-GWO). The crossover and mutation were used to improve local search and preserve wolf variety. By taking into account several goals at once, such as lowering costs, energy use, and making span, a new fitness function was established. However, the best solution was not always found, particularly when working with large and intricate networks.

To categorize requests and identify processing targeted levels in fog-cloud structures, Srichandan et al. [23] created the Adaptive Neuro-Fuzzy Inference System (ANFIS). At the target layer, the Chaotic Honey Badger (CHB) technique was used to schedule such requests. A chaotic mapping function was integrated with an Opposition-based Learning (OBL) strategy to enhance HBA's convergence. Though it was difficult to predict the loads of each compute node to schedule requests on available nodes.

Mahapatra et al. [24] presented a Dynamic Energy-and-Latency-Aware Task (DELTa) scheduling for fog-cloud networks. A multi-level queue technique was applied to prioritize tasks and find the suitable node for offloading. Then, the DELTAmethod was used to effectively schedule tasks onto the chosen nodes for execution. However, it may struggle in large-scale, dynamic, and heterogeneous settings. Rateb et al. [25] investigated the TS issue of IoT devices in a cloud-fog setting. First, Aquila and Salp Swarm Algorithms (ASSA) was utilized to pick ideal VMs for executing workflows. Then, Reducing MakeSpan Time (RMST) method minimized the MST of task on chosen VMs. Moreover, utilizing VM inclusion and the Dynamic Voltage Frequency Scaling (DVFS) approach on result from RMST, the static and dynamic energy utilization were minimized. However, bandwidth and memory utilization were impacted due to imbalanced load conditions.

2.2. VM Migration

Kaur et al. [26] used the Smart Elastic Scheduling mechanism (SESA) to create a low energy VM distribution and relocation mechanism. They used cosine similarity and bandwidth usage to improve QoS performance. However, energy usage has remained high. It is also difficult to implement on large-scale networks. Singh & Singh [27] presented a metaheuristic VM relocation utilizing Re-initialization and Decomposition-based Whale Optimization Algorithm (RD-WOA), which integrates the WOA with NSGA-II to find Pareto-optimality solutions. They enhanced work distribution to the VM, reducing VM migrations, costs, and energy usage. However, they did not consider optimal scheduling of tasks, which may impact the makespan and resource usage.

Swarnakar et al. [28] developed a multi-agent-based VM migration for dynamic LB in cloud computing. However, it has high response time and processing time. Yao et al. [29] proposed a low-power LB technique using VM merging in cloud networks. To prevent needless VM migrations, a load state categorization method was first used for PMs with load abnormalities that took into account present and prospective loads. Then, a resource-weighted assortment technique for driftable VMs was created, which picks proper VMs to move using multidimensional resource use while reducing resource fragmentation. Furthermore, to arrange VMs in the best PMs, a resource fitness and load correlation approach was applied. However, the amount of VM migrations and energy usage were high.

Radi et al. [30] created a Modified Genetic-based VM Consolidation (MGVMC) technique for replacing VMs online while accounting for energy usage, SLA breaches, and the amount of VM migrations. They used the GA to transfer VMs to the appropriate PM in such a manner that the number of overutilized and underutilized PMs was kept to a minimum. However, they did not consider the influence on dependability and scalability while expanding the number of jobs. Wu et al. [31] developed Predicted Mixed Integer Linear Programming (MILP) Robust Solver (PMRS) using Γ-robustness theory for VM migration. Initially, VMs' future actions were predicted, and the problem was formulated as a Γ-robust knapsack problem (Γ-RKP), which is solved using a new MILP technique. However, it is computationally demanding and lacks resilience.

Alsadie and Alsulami [32] introduced a Modified Feeding Birds Algorithm (ModAFBA) for VM association to increase resource distribution and effectiveness in cloud data centers. It implemented adaptive position updating rules and tactics to reduce VM migrations. However, it requires extra data, such as application requirements and workload patterns, to improve the decision-making process. Liu et al. [33] proposed a unique technique for LB and VM migration that uses modeled agents and IoT devices to maximize resource usage and task allocation. They presented a unique Optimized VM Migration Scheme (OVMMS) that migrates VMs using the

Squirrel Search Algorithm (SSA) during migration and search. However, energy consumption was not taken into consideration, affecting network efficiency.

Siruvoru and Aparna [34] created a Harmonic Migration Algorithm (HMA) by combining the migration algorithm with harmonic analysis to migrate a VM from an overloaded to an underloaded PM and enable or disable the VM using switching techniques in cloud computing. The tasks were given to the relevant VM in a round-robin fashion, and the VM's load was forecasted using the gated recurrent unit. However, it relocated the VM when the expected load exceeded the threshold, which is not suitable in real-time cloud systems.

Archana and Kumar [35] successfully introduced an altered BAT method for VM migration in a cloud setting. Spider monkey optimization's fitness function calculation approach was incorporated into the regular bat algorithm to optimize every bat's fitness computation and give further possibilities to select the best option rapidly. It allows VMs to join with the neighbouring PMs and finish the migration quickly, rather than being stalled. However, the network's energy usage throughout the migration method was not taken into account, which influenced network efficiency.

From this literature, it is observed that most of the earlier TS and LB schemes consider single objectives, such as delay or energy, to enhance resource use. They have scalability issues for large-scale networks due to the lack of additional objectives like makespan, throughput, execution cost, etc. Moreover, earlier VM migration strategies often fail to integrate TS, leading to high energy consumption and unwanted VM migrations. As a result, this study proposes the MOWOTSLB scheme using the optimization technique for enhancing resource usage in IoT-fog-cloud networks.

3. PROPOSED METHODOLOGY

In this section, the proposed MOWOTSLB scheme is described in detail. First, the system model, followed by the task and resource model. Then, problem formulation and HOA for TS and VM migration optimization problems are described. A graphic version of this manuscript is presented in Figure 1.

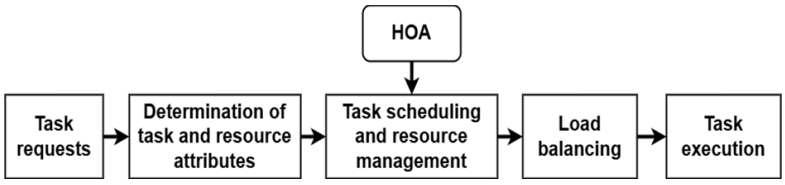

Figure 1: Graphical depiction of this research

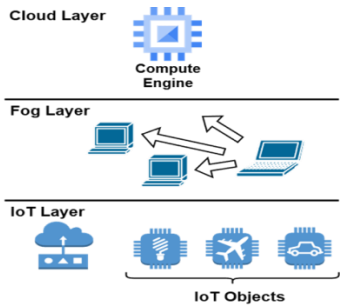

Figure 2: Three-Layer IoT-Fog-Cloud Infrastructure

3.1. System Model

Figure 2 shows the edge, fog, and cloud nodes of the IoT-Fog-Cloud network. All edges request several tasks. The traffic patterns include M/M/1, M/M/c, and M/M/∞ delays at each level. Wireless channels connect edge and fog nodes. The fog nodes can handle inquiries if the highest approval rate exceeds the inquiry rate. Additionally, fog nodes can send communications to the cloud for computation.

3.2. Task and Resource Model

Assume a list of independent tasks $T = \{T_1, T_2, \dots, T_n\}$ created by the edge devices to be implemented. These are transmitted to the relevant resources to be handled in the fog-cloud computing scenario. Each task T_i contains parameters such as task length (in million instructions), deadline, memory requirements, and amount of input and output files. Consider a group of computation nodes $M = M_f \cup M_c$ in fog-cloud network, in which $M_f = \{M_1, \dots, M_{m_f}\}$ denotes the group of fog nodes and $M_c = \{M_1, \dots, M_{m_c}\}$ denotes the set of cloud nodes. So, the sum amount of computing nodes is signified as $m = m_f + m_c$. Every node M_j includes features such as CPU processing rate (measured in millions of instructions per second (MIPS)), memory size, bandwidth, energy utilization, communication delay, resource usage cost.

3.3. Problem Formulation

HOA-based TS aims to distribute tasks among compute nodes as efficiently as possible to diminish makespan, execution cost, and energy utilization although meeting task deadline and resource constraints. The binary allocation matrix Q of dimension $n \times m$ is defined as follows:

$$Q_{ij} = \begin{cases} 1, & if\ T_i\ is\ allocated\ to\ M_j \\ 0, & or\ else \end{cases} \quad (1)$$

Constraints

- Task allocation constraint: Every task should be allocated to precisely to one computation node as defined in Eq. (2).

$$\sum_{j=1}^{m} Q_{ij} = 1, \forall i = 1, \dots, n \quad (2)$$

Deadline constraint: Completion Time (CT) of every task never surpasses its limit, as represented in Eq. (3).

$$CT_i \leq Deadline_i, \forall i = 1, \dots, n \quad (3)$$

In Eq. (3), $Deadline_i$ is the deadline for task T_i to be executed.

- Resource size constraint: The entire resource requirement in every node never surpasses its size, as in Eq. (4).

$$\sum_{t=1}^{n} Mem_i \times Q_{ij} \leq MemCap_j, \forall j = 1, \dots, m \quad (4)$$

In Eq. (4), Mem_i is the memory requirement of task T_i and $MemCap_j$ is the complete memory size of M_j in MB.

3.3.1. Fitness Function for Task Scheduling

Makespan: For n tasks and m computing nodes, the Execution Time (ET) matrix with dimension $n \times m$ is used to determine the ET for task requests on nodes. The task scheduler uses the ET

matrix to make TS choices on the fog-cloud platform. The ET of i^{th} task on the j^{th} computing node (ET_{ij}) is determined as:

$$\mathrm{ET_{ij}} = \frac{\mathrm{Task\_length_i}}{\mathrm{CPU_j}} \tag{5}$$

In Eq. (5), $\mathrm{Task\_length_i}$ denotes the length of $\mathrm{i^{th}}$ task and $\mathrm{CPU_j}$ is the handling ability of $\mathrm{j^{th}}$ node in MIPS. The primary goal of the TS issue is to determine the optimal fog-cloud system schedule that minimizes the makespan, or task execution time. In this situation, no task will certainly take a considerable amount of time to perform and finish. So, the aim is to reduce the makespan (MK), which is represented in Eq. (6).

$$MK = \max_{j\in 1,\ldots,m} \left(\sum_{i=1}^{n} ET_{ij} \times Q_{ij}\right) \tag{6}$$

Total Execution Cost: The total execution cost (C_{tot}) comprises computational cost (C_{comp}), communication cost (C_{comm}), and deadline violation cost (C_{dv}), which is defined in Eq. (7).

$$\mathrm{C_{tot}} = \mathrm{C_{comp}} + \mathrm{C_{comm}} + \mathrm{C_{dv}} \tag{7}$$

$$\text{Where } \mathrm{C_{comp}} = \sum_{i=1}^{n}\sum_{j=1}^{m}\left(\mathrm{C_{cpu,j}} \times \mathrm{ET_{ij}} + \mathrm{C_{mem,j}} \times \mathrm{Mem_i}\right) \times \mathrm{Q_{ij}} \tag{8}$$

$$\mathrm{C_{comm}} = \sum_{i=1}^{n}\sum_{j=1}^{m} \mathrm{C_{bw,j}} \times \mathrm{DataSize_i} \times \mathrm{Q_{ij}} \tag{9}$$

$$\mathrm{C_{dv}} = \sum_{i=1}^{n} \mathrm{Penalty_i} \times \max\left(0, \frac{\mathrm{CT_i - Deadline_i}}{\mathrm{Deadline_i}}\right) \tag{10}$$

In Eqns. (8)-(10), $\mathrm{C_{cpu,j}}, \mathrm{C_{mem,j}}$, and $\mathrm{C_{bw,j}}$ are cost per unit CPU, memory, and bandwidth utilizations on $\mathrm{M_j}$, respectively, $\mathrm{DataSize_i}$ is the $\mathrm{T_i}$ size in MB, and $\mathrm{Penalty_i}$ denotes penalty cost for delay.

Total Energy Utilization: The goal is to reduce the energy utilization E_{tot}, which is represented in Eq. (11).

$$\mathrm{E_{tot}} = \sum_{j=1}^{m}\left(\mathrm{ActiveTime_j} \times \alpha_j + \left(\mathrm{MK - ActiveTime_j}\right) \times \beta_j\right) \tag{11}$$

$$\text{Where } \mathrm{ActiveTime_j} = \sum_{i=1}^{n} \mathrm{ET_{ij}} \times \mathrm{Q_{ij}} \tag{12}$$

In Eq. (11), α_j and β_j are the energy utilization rates when node M_j is active and idle, respectively.

The HOA is a metaheuristic algorithm that schedules tasks based on a fitness function to discover the most appropriate computer resources. In this study, the fitness function selects the solution depending upon the minimum MK, C_{tot}, and E_{tot}. Thus, the fitness function for the above multi-objective TS issue (f_{TS}), which uses the weighted sum technique to balance the objectives, is defined as follows:

$$\mathrm{f_{TS}} = (\omega_1 \times \mathrm{MK}) + (\omega_2 \times \mathrm{C_{tot}}) + (\omega_3 \times \mathrm{E_{tot}}) \tag{13}$$

In Eq. (13), $\omega_1, \omega_2, \omega_3$ are the weight parameters such that $\omega_1 + \omega_2 + \omega_3 = 1$.

3.3.2. Fitness Function for VM Migration

In fog-cloud network, consider q number of PMs $Q = \{Q_1, Q_2, \dots, Q_q\}$ and v number of VMs are available in each PM, denoted as $V = \{V_1, V_2, \dots, V_v\}$. The edge user can schedule each task request to each VM using HOA, represented by $T_v = \{T_{v_1}, T_{v_2}, \dots, T_{v_n}\}$, where v_n denotes number of tasks scheduled to each VM. Besides, each VM has various parameters such as CPU usage, memory, bandwidth, MIPS, delay, and the number of processing elements. So, the v^{th} VM in q^{th} PM is defined as:

$$V_v^q = \{H_v^q, D_v^q, A_v^q, B_v^q, G_v^q, E_v^q\} \quad (14)$$

In Eq. (14), H_v^q is the total amount of processing elements of v^{th} VM in q^{th} PM, D_v^q is the total amount of CPUs utilized by v^{th} VM in q^{th} PM, A_v^q and B_v^q are the bandwidth and memory of v^{th} VM in q^{th} PM, respectively. Also, G_v^q is the total number of MIPS used by v^{th} VM in q^{th} PM, and E_v^q is the delay used by v^{th} VM in q^{th} PM.

Estimation of VM and PM Load:Depending on the VM task processing resources that are retrieved from the edge users, the load of VM is assessed. The VM load (L) is calculated as:

$$L_v = \frac{RU}{t} \quad (15)$$

In Eq. (15), t refers to the interval and RU represents the resource use, which is computed as:

$$RU = \frac{1}{G}\sum_{i=1}^{v}\left(\frac{H^G}{\max H^G} + \frac{D^G}{\max D^G} + \frac{A^G}{\max A^G} + \frac{B^G}{\max B^G} + \frac{P^G}{\max P^G} + \frac{Z^G}{\max Z^G}\right) \quad (16)$$

In Eq. (16), G is the normalization factor, and $\max(\cdot)$ is the maximum value of each parameter. So, the load of PM is determined as:

$$L_q = \frac{1}{V}\sum_{i=1}^{q} L_v \quad (17)$$

Migration Cost: It is the proportion of the total amount of migrations completed to the number of migrations attempted. It is computed as:

$$MC = \frac{1}{Q}\sum_{j=1}^{q}\left(\frac{\tau}{\epsilon \times V}\right) \quad (18)$$

In Eqns. (17)-(18), Q is the total amount of PMs, V denotes the total VMs, τ denotes total migrations of it, and ϵ is a migration constant.

Energy Utilization: The energy usage is solely determined by how much power is used by the many resources that are accessible in the VM. It is defined as:

$$EC = \frac{1}{T}\sum_{t=1}^{T} U \quad (19)$$

In Eq. (19), T is the total interval and U is the power consumed, which is determined as:

$$U = s \times U_{max} + (1 - s) \times U_{max} \times RU \quad (20)$$

In Eq. (20), s is a scaling factor, U_{max} is the maximum power used and RU is the resource utilization.

Bandwidth Factor: It is calculated by

$$A = \sum_{i=1}^{q} \sum_{j=1}^{v} (B_{ij} + B_j) \quad (21)$$

In Eq. (21), B_{ij} is the bandwidth of i^{th} VM in j^{th} PM, and B_j is the bandwidth used for migration.

Thus, the fitness function for the above multi-objective VM migration issue (f_{VM}), which uses the weighted sum technique to balance the objectives, is defined as follows:

$$f_{VM} = (\omega_1 \times L_q) + (\omega_2 \times MC) + (\omega_3 \times EC) + (\omega_4 \times A) \quad (22)$$

In Eq. (22), $\omega_1, \omega_2, \omega_3, \omega_4$ are the weight parameters such that $\omega_1 + \omega_2 + \omega_3 + \omega_4 = 1$.

3.4. Horse Herd Optimization Algorithm

The HOA is encouraged by horse herding activities, such as mimicking the social activities of horses of different ages through grazing, hierarchy, sociability, mimicry, defense strategy, and roaming characteristics. This HOA outperforms the GA, GWO, MFO, and others in addressing high complications, owing to the vast amount of regulation factors depending on horse activities at various ages. The basic terminologies of HOA are described based on the TS and VM migration optimization dilemmas in Table 1.
Eq. (23) serves as the basis for the movement that applies to horses with each repetition.

$$X_m^{itr,Age} = \vec{V}_m^{itr,Age} + X_m^{(itr-1),Age}, Age = \alpha, \beta, \gamma, \delta \quad (23)$$

In Eq. (23), $X_m^{itr,Age}$ is the location of m^{th} horse, itr indicates the present iteration, and $\vec{V}_m^{itr,Age}$ defines the velocity of horse.

Table 1. Basic Terminologies in HOA for TS and VM Migration Optimization Problems

Terminologies	**Descriptions for optimization problems in this study**	
	TS	**VM migration**
Horse	A possible assignment of tasks to nodes.	A possible VM to PM migration.
Horse location	A specific schedule of tasks on nodes.	A specific configuration of VMs on PMs.
Herd	Populations of candidate TS schedules.	Population of VM migrations.
Velocity	Directional variation in task to node allocation.	Directional variation in VM to PM migration.
Fitness function	A weighted combination of makespan, execution cost, and energy use, as defined in Eq. (13).	A weighted combination of PM load, migration cost, energy, and bandwidth, as defined in Eq. (22).
Minimum and maximum margins	Lowest and highest values of task attributes considered in TS problem.	Lowest and highest values of resource parameters considered in VM migration problem.

In separate years, horses have distinct characteristics. A horse may live for a maximum of 25 to 30 years. Here, δ denotes 0-5 years, γ between 5-10, β between 10-15, and α above 15. To choose

the horse ages, a complete array of replies should be created for each iteration. The maximum 10% of the organized array is selected as α horses, as the matrix can be sorted based on preferred replies.

The β group includes the following 20%. In the remaining, 30% are γ horses and 40% are δ horses. The velocity is detected by analytically implementing the processes to replicate the six activities of the horses. Regarding the behaviour patterns, Eq. (24) represents the velocity of horses at different ages through every process.

$$\vec{V}_m^{itr,\alpha} = \vec{G}_m^{itr,\alpha} + \vec{D}_m^{itr,\alpha} \quad (24a)$$
$$\vec{V}_m^{itr,\beta} = \vec{G}_m^{itr,\beta} + \vec{H}_m^{itr,\beta} + \vec{S}_m^{itr,\beta} + \vec{D}_m^{itr,\beta} \quad (24b)$$
$$\vec{V}_m^{itr,\gamma} = \vec{G}_m^{itr,\gamma} + \vec{H}_m^{itr,\gamma} + \vec{S}_m^{itr,\gamma} + \vec{I}_m^{itr,\gamma} + \vec{D}_m^{itr,\gamma} + \vec{R}_m^{itr,\gamma} \quad (24c)$$
$$\vec{V}_m^{itr,\delta} = \vec{G}_m^{itr,\delta} + \vec{I}_m^{itr,\delta} + \vec{R}_m^{itr,\delta} \quad (24d)$$

The primary stages of societal and individualistic intellect in horses are deliberated in the following sections.

Grazing:It is the local search around current task schedules or VM migration mappings. In this stage, as grazing animals, horses consume grasses, plants, forage, and other vegetation. They graze 16–20 hours a day, and they don't get much sleep. This persistent grazing is named constant eating. HOA represents graze zone using factor g which surrounds each horse, meaning that each horse grazes on specific locations. Horses graze throughout their lives, regardless of age. Grazing is represented mathematically by

$$\vec{\mathrm{G}}_{\mathrm{m}}^{\mathrm{itr,Age}} = \mathrm{g}_{\mathrm{itr}}(\breve{\mathrm{u}} + \mathrm{p}\breve{\mathrm{l}})\left[\mathrm{X}_{\mathrm{m}}^{(\mathrm{itr}-1)}\right], \mathrm{Age} = \alpha, \beta, \gamma, \delta \quad (25)$$
$$\mathrm{g}_{\mathrm{m}}^{\mathrm{itr,Age}} = \mathrm{g}_{\mathrm{m}}^{(\mathrm{itr}-1),\mathrm{Age}} \times \omega_{\mathrm{g}} \quad (26)$$

In Eqns. (25)-(26), $\vec{\mathrm{G}}_{\mathrm{m}}^{\mathrm{itr,Age}}$ represents the movement variable of i^{th} horse, signifying its grazing habit. It minimizes the linearity with ω_{g} per repetition. ĭ and ŭ denote the least and extreme margins of graze area, correspondingly, and p denotes the arbitrary value from 0 to 1. ĭ and ŭ denote fixed as 0.95 and 1.05, correspondingly, and factor g is set to 1.5 for each age range.

Hierarchy:It yields the ideal TS or VM migration solutions. In this stage, horses cannot be left alone. As humans frequently do, they conduct their lives in obedience to a leader. According to the rule of hierarchy, a mare or a mature stallion oversees leadership for a group of wild horses. Here, the group of horses to trail behind the most capable and seasoned steed is represented by factor h in HOA. According to studies, when the horses are in the ages of 5 – 15, or middle agesβ and γ, they adhere to the law of hierarchy. Eqs. (27 – 28) states:

$$\vec{H}_m^{itr,Age} = h_m^{itr,Age}\left[X_*^{(itr-1)} - X_m^{(itr-1)}\right], Age = \alpha, \beta, \gamma \quad (27)$$
$$h_m^{itr,Age} = h_m^{(itr-1),Age} \times \omega_h \quad (28)$$

In Eqns. (27)-(28), $\vec{H}_m^{itr,Age}$ is the impacts of its ideal position on its velocity variable and $X_*^{(itr-1)}$ denote its ideal position.

Sociability: It moves towards the mean of better TS or VM migration solutions. In this stage, horses sometimes live with other animals and need a social existence. As a result of herd life, the horses have been protected from predators. Pluralism facilitates escape and improves survival chances. As a result of their sociable nature, horses are often seen engaged in fights, and their

uniqueness contributes to their irritation. Some of the horses seem to take pleasure in being with sheep and cattle, but they seldom ever enjoy being by themselves. It is exhibited by factor s which represent a shift near the normal place of another horses. Eqns. (29) and (30) demonstrates the obvious interest of horses in the ages of 5 -15 have in the herd:

$$\vec{S}_m^{itr,Age} = s_m^{itr,Age}\left[\left(\frac{1}{N}\sum_{j=1}^{N} X_j^{(itr-1)}\right) - X_m^{(itr-1)}\right], Age = \beta, \gamma \quad (29)$$

$$s_m^{itr,Age} = s_m^{(itr-1),Age} \times \omega_s \quad (30)$$

In Eqns. (29)-(30), $\vec{S}_m^{itr,Age}$ represents societal activity vector of i^{th} horse and $s_m^{itr,Age}$ is the horse's herd orientation in itr^{th} iteration. $s_m^{itr,Age}$ decreases in every cycle with ω_s feature.

Imitation: It mimics the optimal TS or VM migration characteristics. In this stage, horses copy each other's good and bad behavior, like choosing a pasture. In this approach, horses' mimicking behavior is also regarded as factor i. Eqns. (31) and (32), which explain how young horses attempt to imitate others, do not go away as they grow older.

$$\vec{I}_m^{itr,Age} = i_m^{itr,Age}\left[\left(\frac{1}{pN}\sum_{j=1}^{pN} \hat{X}_j^{(itr-1)}\right) - X^{(itr-1)}\right], Age = \gamma \quad (31)$$

$$i_m^{itr,Age} = i_m^{(itr-1),Age} \times \omega_i \quad (32)$$

In Eqns. (31)-(32), $\vec{I}_m^{itr,Age}$ stands for the i^{th} horse's movement vector towards the mean of ideal ones. With $\hat{X}$ positions, pN indicates horses with ideal positions. Here, p is equivalent to 10% of the total.

Defense Strategy: It prevents poor TS or VM migration decisions. In this stage, the response of horses reflects their experience as prey to predators. They utilize the emergency reaction to protect them. Their initial reaction is to flee. They also kick when someone traps them. Horses naturally compete for foodstuff and water to expel opponents and remain away from unsafe regions with the presence of dangers like wolves. The HOA algorithm's defensive mechanism for horses is to flee from them when they exhibit unsuitable or subpar reactions. The factor d describes their defensive mechanism. As previously said, horses must either escape or conflict with their enemies. When feasible, a horse's defensive system is present for the whole of its life, whether it is young or adult. In Eqns. (33) and (34),The negative coefficient symbolizes the horse's defense mechanism, which prevents it from being exposed to inappropriate surroundings.

$$\vec{D}_m^{itr,Age} = -d_m^{itr,Age}\left[\left(\frac{1}{qN}\sum_{j=1}^{qN} \breve{X}_j^{(itr-1)}\right) - X^{(itr-1)}\right], Age = \alpha, \beta, \gamma \quad (33)$$

$$d_m^{itr,Age} = d_m^{(itr-1),Age} \times \omega_d \quad (34)$$

In Eqns. (33)-(34), $\vec{D}_m^{itr,Age}$ denotes the escaping vector of i^{th} horse from mean of approximate ones with poorest places, depicted by $\breve{X}$, qN represents horses in the poorest places. Here, q is equivalent to 20% of the total, and ω_d is the decline coefficient for each cycle of d_{itr} as early observed.

Roaming: It explores new TS or VM migrations to prevent local optima. In this stage, horses wander and graze in wildlife, moving in quest of nourishment. Many horses are maintained in stables, although this preserves the aforementioned features. A horse can quickly move to a new spot to graze. They are incredibly curious, and they frequently explore different pastures and learn about their surroundings. The side walls are created so that they may grasp each of them

and satisfy their interest in an acceptable stable. A component denotes this tendency, that is imitated as casual effort. It roams nearly exclusively at a young age and progressively depart as they mature. It is specified in Eqns. (35) and (36).

$$\vec{R}_m^{itr,Age} = r_m^{itr,Age} pX^{(itr-1)}, Age = \gamma, \delta \tag{35}$$

$$r_m^{itr,Age} = r_m^{(itr-1),Age} \times \omega_r \tag{36}$$

In Eqns. (35)-(36), denotes the arbitrary velocity vector of i^{th} horse for a local search and an escape from local minima, and ω_r signifies the reduction factor of $r_m^{itr,Age}$ per cycle.

The random velocity of the i^{th} horse for local searches and escapes from local minima is denoted by $\vec{R}_m^{itr,Age}$ and the reduction factor of $r_m^{itr,Age}$ per cycle is written as ω_r.

The general velocity vector is calculated by inserting Eqns. (25) to (36) into Eq. (24). Velocity of δ horses is:

$$\vec{V}_m^{itr,\delta} = \left[g_m^{(itr-1),\delta} \times \omega_g(\check{u} + p\check{l})\left[X_m^{(itr-1)}\right]\right] + \left[i_m^{(itr-1),\delta}\omega_i\left[\left(\frac{1}{pN}\sum_{j=1}^{pN}\hat{X}_j^{(itr-1)}\right) - X^{(itr-1)}\right]\right] + \left[r_m^{(itr-1),\delta}\omega_r pX^{(itr-1)}\right] \tag{37}$$

Velocity of γ horses is:

$$\vec{V}_m^{itr,\gamma} = \left[g_m^{(itr-1),\gamma} \times \omega_g(\check{u} + p\check{l})\left[X_m^{(itr-1)}\right]\right] + \left[h_m^{(itr-1),\gamma}\omega_h\left[\left[X_*^{(itr-1)} - X_m^{(itr-1)}\right]\right]\right] + \left[s_m^{(itr-1),\gamma}\omega_s\left[\left(\frac{1}{N}\sum_{j=1}^{N}X_j^{(itr-1)}\right) - X_m^{(itr-1)}\right]\right] + \left[i_m^{(itr-1),\gamma}\omega_i\left[\left(\frac{1}{pN}\sum_{j=1}^{pN}\hat{X}_j^{(itr-1)}\right) - X^{(itr-1)}\right]\right] - \left[d_m^{(itr-1),\gamma}\omega_d\left[\left(\frac{1}{qN}\sum_{j=1}^{qN}\breve{X}_j^{(itr-1)}\right) - X^{(itr-1)}\right]\right] + \left[r_m^{(itr-1),\gamma}\omega_r pX^{(itr-1)}\right] \tag{38}$$

Velocity of β horses is:

$$\vec{V}_m^{itr,\beta} = \left[g_m^{(itr-1),\beta} \times \omega_g(\check{u} + p\check{l})\left[X_m^{(itr-1)}\right]\right] + \left[h_m^{(itr-1),\beta}\omega_h\left[\left[X_*^{(itr-1)} - X_m^{(itr-1)}\right]\right]\right] + \left[s_m^{(itr-1),\beta}\omega_s\left[\left(\frac{1}{N}\sum_{j=1}^{N}X_j^{(itr-1)}\right) - X_m^{(itr-1)}\right]\right] - \left[d_m^{(itr-1),\beta}\omega_d\left[\left(\frac{1}{qN}\sum_{j=1}^{qN}\breve{X}_j^{(itr-1)}\right) - X^{(itr-1)}\right]\right] \tag{39}$$

Velocity of α horses is:

$$\vec{V}_m^{itr,\alpha} = \left[g_m^{(itr-1),\alpha} \times \omega_g(\check{u} + p\check{l})\left[X_m^{(itr-1)}\right]\right] - \left[d_m^{(itr-1),\alpha}\omega_d\left[\left(\frac{1}{qN}\sum_{j=1}^{qN}\breve{X}_j^{(itr-1)}\right) - X^{(itr-1)}\right]\right] \tag{40}$$

Table 2 presents the parameters used in the HOA, and the pseudocode of the HOA for TS and VM migration tasks is summarized in Algorithm 1. Also, Figure 3 illustrates the entire schematic representation of the proposed MOWOTSLB scheme in an IoT-fog-cloud network.

Table 2: HOA Parameters

Parameters	**Values**
No. of horses	80
No. of iterations	500
Grazing coefficient g	1.5
h^{β}, h^{γ}	0.9, 0.5
s^{β}, s^{γ}	0.2, 0.1
i^{γ}	0.3
$d^{\alpha}, d^{\beta}, d^{\gamma}$	0.5, 0.2, 0.1
r^{δ}, r^{γ}	0.1, 0.05

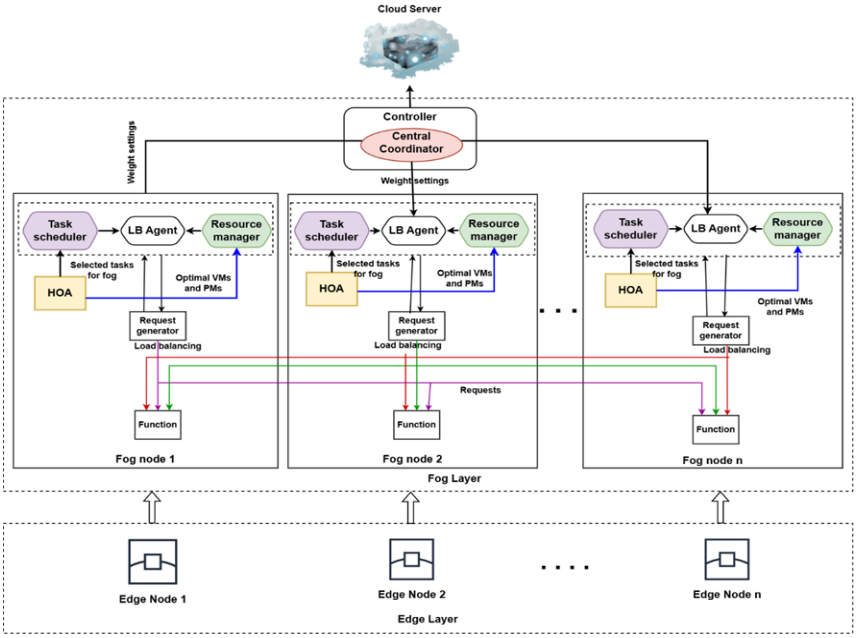

Figure 3. Schematic Representation of Proposed MOWOTSLB Scheme

Algorithm 1: HOA for TS and VM Migration

Input: Horse population, maximum iteration, T set of tasks, M set of computing nodes, Q set of PMs, and V set of VMs

Output: Group of ideal task schedules and optimal VMs to be migrated to the optimal PMs

1. Begin
2. Generate the initial population of horses in the search space arbitrarily;
3. Estimate the fitness of an individual horse using Eqns. (13) and (22), i.e.,
4. Update the ideal horse's position according to fitness value;
5. while(itr $\leq$ itr$_{max}$)
6. for(i = 1: total horses)
7. Calculate the center of each herd from the maximum iterations.
8. Assess the rank of each horse in rising order based on their fitness value;
9. Compute the velocity of each horse.
10. Modify the location of each horse;
11. Evaluate the fitness for the updated location ofthe individual horse;

12. if (updated fitness <previous fitness)
13. $new\ location = Optimal\ location$;
14. $new\ fitness = Optimal\ fitness$;
15. end if
16. end for
17. end while
18. Return optimal location and optimal fitness value (i.e., best task schedules and VM migration mappings);
19. End

4. SIMULATION RESULTS

This part compares the efficacy of the MOWOTSLB with current schemes, such as DLJSF [20], ANFIS-CHBA-OBL [22], DELTA[24], and SESA [26]. Throughput, delay, response time, network lifespan, energy, memory, and bandwidth utilization are among the performance parameters evaluated.

4.1. Simulation Setup

Windows operating systems running on Intel CPUs with 16 GB of RAM were used for the testing. To account for variations in load and identify idle fog gates and overload situations, the virtual Core i5-4210 CPU lifetime is set at 16.66 minutes. Twenty nodes were used in this investigation, each of which produced information with ten tasks per second, for a total of 200 tasks per second. Additionally, the data transfer rate for all nodes was 40 MB/s, with each node transferring data at a rate of 2 MB/s. To guarantee accurate and useful outcomes, the simulation runs 1200 instances with different settings. The simulation parameters and the values that correspond to them are shown in Table 3.

Table 3. Simulation Constraints

Parameter	**Values**
Tasks	200 to 1000 tasks/s
Cloud nodes	5
CPU rate (Cloud & Fog)	100×109 cycles/s
CPU rate (IoT)	500×106 cycles/s
End nodes and fog nodes	50-100, 10-100
Gateway devices	2
Highest channel bandwidth	30 MHz
Processing power utilization	0.5 J
Transmit power (IoT)	0.2 mW

4.2. Convergence Analysis for HOA with Other Metaheuristic Algorithms

In this section, the HOA is compared with the other metaheuristic algorithms to understand its efficiency in TS and VM migration optimization problems. The compared algorithms include GWO [16], AO-AVOA [18], MFO [21], ASSA [25], and SSA [33]. Figure 4 depicts the convergence curves for HOA and other optimization techniques using the same number of populations (80) and iterations (500). The HOA-based TS and VM migration methods produce a more optimum solution and are significantly faster than the SSA, MFO, GWO, ASSA, and AO-AVOA algorithms for the identical system design and operating circumstances.

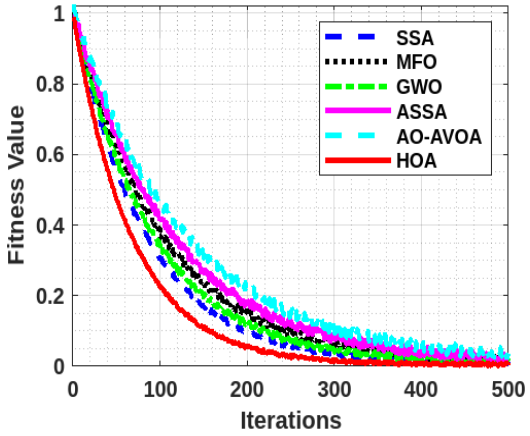

Figure 4. Convergence Curve for HOA and Other Metaheuristic Algorithms

4.3. Throughput

It specifies the requests or tasks count handled on the system for each interval. This is determined as:

$$\text{Throughput} = \frac{\text{Number of tasks}}{\text{MK}} \tag{41}$$

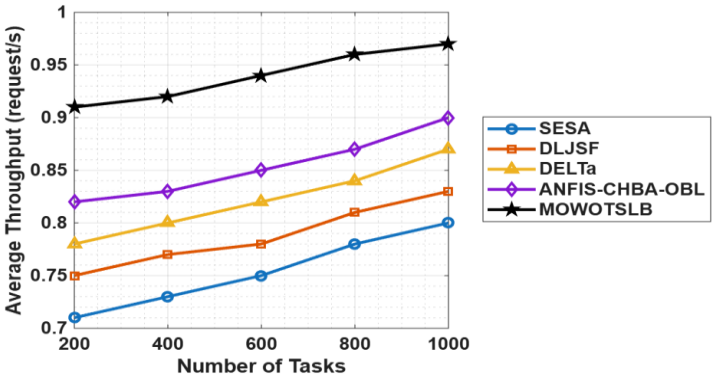

Figure 5. Throughput against Number of Tasks

The average TP for many LBTS schemes with various task counts is displayed in Figure 5. For every number of tasks, the MOWOTSLB performs better in terms of TP efficiency than SESA, DLJSF, DELTA, and ANFIS-CHBA-OBL. For example, compared to these current schemes, the MOWOTSLB improves TP by 21.25%, 16.87%, 11.49%, 7.78%, and 1.04% for 1000 tasks in the network, respectively. This is a result of allocating tasks to the right resources and moving VMs to the right PMs if an overloaded or underloaded scenario arises while a task is being executed.

4.4. Delay

Both transmission and processing delays contribute to the IoT-fog-cloud operating latency. It is computed in the following manner:

$$\text{Delay} = \frac{\text{Communication delay+Computation delay}}{\text{Total number of devices}} \tag{42}$$

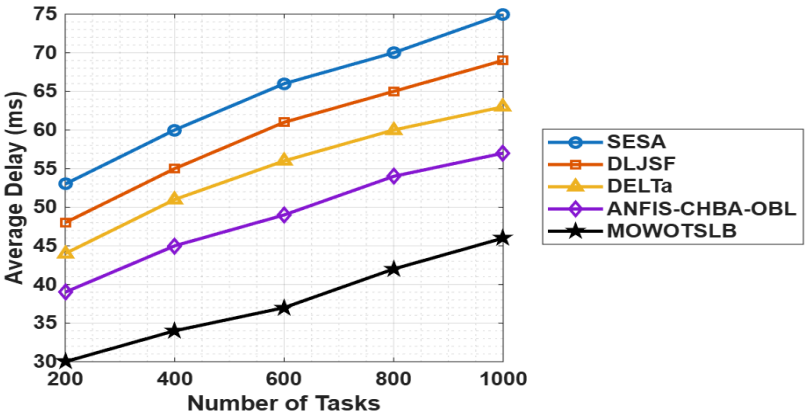

Figure 6. Delay vs. No. of Tasks

A comparison of the average latency for several LBTS schemes with different task counts is shown in Figure 6. For every number of tasks, the MOWOTSLB lowers the latency in comparison to the SESA, DLJSF, DELTA, and ANFIS-CHBA-OBL schemes. For instance, compared to the current schemes, the MOWOTSLB reduces the average latency by 38.67%, 33.33%, 26.98%, 19.3%, and 9.8%, respectively, when 1000 tasks are used. This is accomplished by allocating tasks to the right computing nodes while balancing loads in the best possible way.

4.5. Energy Utilization

It indicates the IoT-fog-cloud system's overall power consumption.

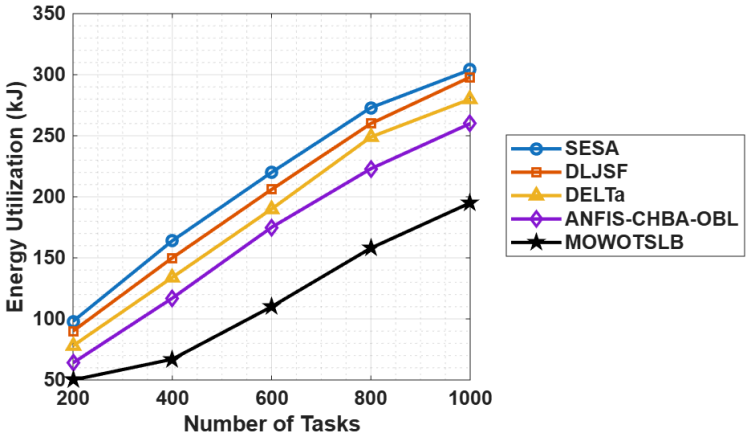

Figure 7. Energy Utilization against number of Tasks

Performance of energy utilization for several LBTS schemes with varied task counts is displayed in Figure 7. For every number of tasks, the MOWOTSLB uses less energy than the SESA, DLJSF, DELTA, and ANFIS-CHBA-OBL schemes. For instance, compared to other schemes, the MOWOTSLB reduces energy consumption by 35.86%, 34.56%, 30.36%, 25%, and 22%, respectively, when 1000 tasks are used. To prevent excessive energy usage, this is accomplished by optimizing TS and VM migration with LB based on energy use as one of the objectives.

4.6. Response Time

Each job's response time is the amount of time needed to reply to it within the allotted time. Response times for several LBTS schemes with varied task counts are compared in Figure 8.

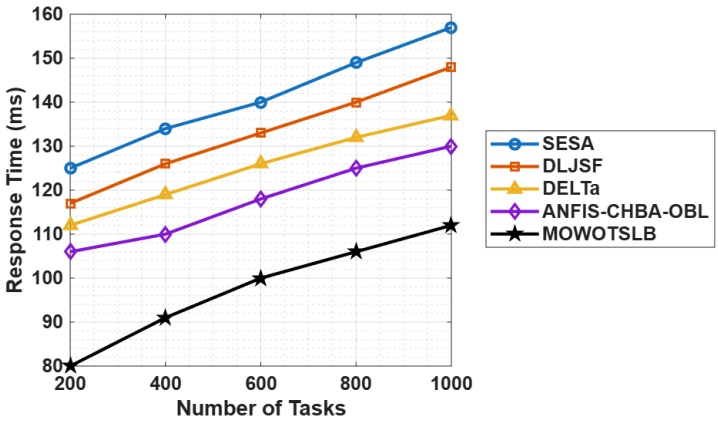

Figure 8. Response Time vs. No. of Tasks

For every task quantity, the MOWOTSLB outperforms the SESA, DLJSF, DELTA, and ANFIS-CHBA-OBL in terms of response time. For instance, compared to the current schemes, MOWOTSLB 's response time for 1000 tasks is reduced by 28.66%, 24.32%, 18.25%, 13.85%, and 8.2%, respectively. This is accomplished by allocating tasks to the best resources so they may be completed within the constraints of their specifications.

4.7. Memory Utilization

This represents full amount of RAM that every VM needs to execute a job.

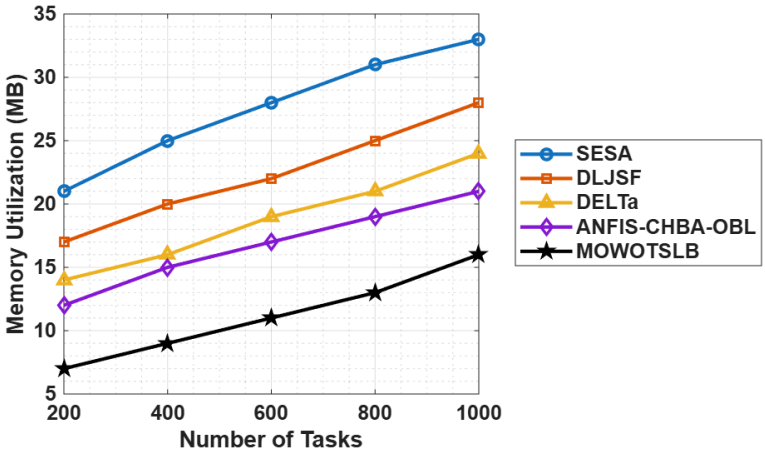

Figure 9. Memory Utilization against number of Tasks

A comparison of memory use for several LBTS schemes with varied task counts is shown in Figure 9. For every number of tasks, the MOWOTSLB method uses less memory than the SESA, DLJSF, DELTA, and ANFIS-CHBA-OBL schemes. For instance, MOWOTSLB uses 51.52%, 42.86%, 33.33%, 23.81%, and 15.79% less memory than the other schemes when 1000 tasks are used. This is because of the LB approach, TS optimization, and VM migration to the appropriate PMs.

4.8. Bandwidth Utilization

It is each VM's maximum bandwidth needed to complete a task. The bandwidth use of many LBTS schemes with various task counts is contrasted in Figure 10.

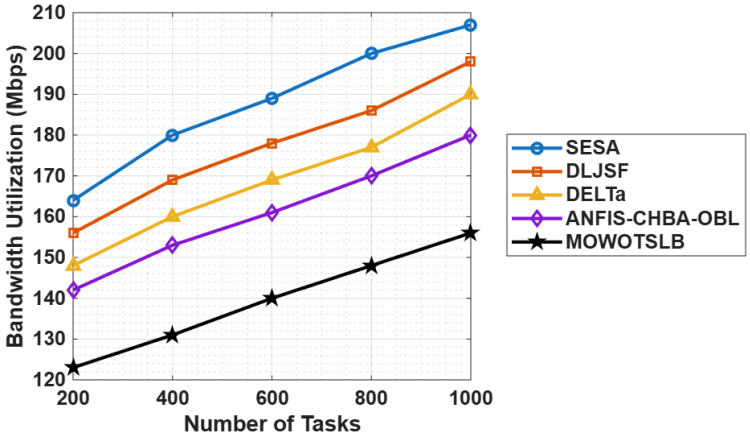

Figure 10. Bandwidth Utilization vs. No. of Tasks

For every number of tasks, the MOWOTSLB reduces bandwidth usage in comparison to the SESA, DLJSF, DELTA, and ANFIS-CHBA-OBL schemes. For instance, MOWOTSLB uses 24.64%, 21.21%, 17.89%, 13.33%, and 8.77% less bandwidth than the other schemes when 1000 tasks are used. This is accomplished by effectively optimizing TS and VM migration based on several parameters, which reduces duplicate data transfers and boosts bandwidth efficiency.

4.9. Network Lifetime

It calculates the lifespan of the system as:

$$Lifetime = \frac{Residual\ energy\ of\ nodes}{Drain\ rate} \tag{44}$$

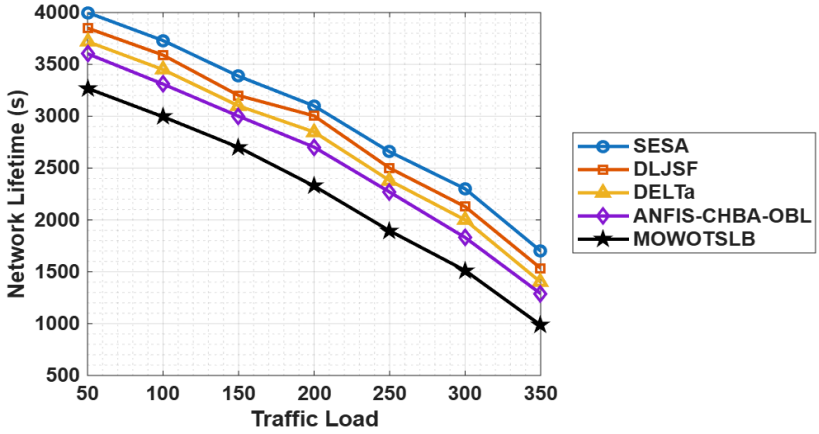

Figure 11. Network Lifetime vs. No. of Tasks

The system lifespan of several LBTS schemes with various traffic volumes is assessed in Figure 11. The SESA, DLJSF, DELTA, and ANFIS-CHBA-OBL schemes are all outperformed by the MOWOTSLB. With a load of 350, the network lifespan for MOWOTSLB is enhanced with

42.06%, 35.79%, 29.64%, 23.58%, and 15.81% compared to the other schemes, respectively. This is accomplished by efficiently allocating the tasks to the proper computing nodes and migrating VMs to the best PMs according to the various criteria, which lowers energy usage and lengthens the network lifespan.

5. CONCLUSION

This article introduces the MOWOTSLB method for IoT-fog-cloud systems, which schedules tasks and migrates VMs to the appropriate PM in a balanced way. In this design, HOA was used for TS and VM migration. This HOA optimizes user TS requests to correct computer resources depending upon numerous objectives. It also identified the finest VMs to migrate to the most appropriate PMs by evaluating their fitness function. This properly balances load distribution among PMs and energy usage. Finally, the results showed that the MOWOTSLB outperformed current schemes in terms of network performance, with a mean throughput of 0.97 requests/sec, a mean delay of 46ms, energy utilization of 195 kJ, response time of 112ms, memory utilization of 16 MB, and bandwidth usage of 156 Mbps for 1000 network tasks.

CONFLICTION OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] Chataut, R., Phoummalayvane, A., & Akl, R., (2023)"Unleashing the power of IoT: a comprehensive review of IoT applications and future prospects in healthcare, agriculture, smart homes, smart cities, and industry 4.0",*Sensors*, Vol. 23, No. 16, p. 7194.

[2] Ficili, I., Giacobbe, M., Tricomi, G., &Puliafito, A., (2025)"From sensors to data intelligence: leveraging IoT, cloud, and edge computing with AI",*Sensors*, Vol. 25, No. 6, p. 1763.

[3] Santoso, A., & Surya, Y., (2024)"Maximizing decision efficiency with edge-based AI systems: advanced strategies for real-time processing, scalability, and autonomous intelligence in distributed environments",*Quarterly Journal of Emerging Technologies and Innovations*, Vol. 9, No. 2, pp. 104-132.

[4] Al-Jumaili, A. H. A., Muniyandi, R. C., Hasan, M. K., Paw, J. K. S., & Singh, M. J., (2023) "Big data analytics using cloud computing based frameworks for power management systems: status, constraints, and future recommendations",*Sensors*, Vol. 23, No. 6, p. 2952.

[5] Rajagopalan, A., Swaminathan, D., Bajaj, M., Damaj, I., Rathore, R. S., Singh, A. R., ... & Prokop, L., (2024)"Empowering power distribution: unleashing the synergy of IoT and cloud computing for sustainable and efficient energy systems", *Results in Engineering*, Vol. 21, p. 101949.

[6] Abd Alnabe, N., &Zeebaree, S. R., (2024)"Distributed systems for real-time computing in cloud environment: a review of low-latency and time sensitive applications",*The Indonesian Journal of Computer Science*, Vol. 13, No. 2.

[7] Avan, A., Azim, A., & Mahmoud, Q. H., (2023) "A state-of-the-art review of task scheduling for edge computing: a delay-sensitive application perspective",*Electronics*, Vol. 12, No. 12, p. 2599.

[8] Hazra, A., Rana, P., Adhikari, M., &Amgoth, T., (2023) "Fog computing for next-generation internet of things: fundamental, state-of-the-art and research challenges",*Computer Science Review*, Vol. 48, p. 100549.

[9] Shwe, T., &Aritsugi, M., (2024)"Optimizing data processing: a comparative study of big data platforms in edge, fog, and cloud layers",*Applied Sciences*, Vol. 14, No. 1, p. 452.

[10] Das, R., & Inuwa, M. M., (2023) "A review on fog computing: issues, characteristics, challenges, and potential applications",*Telematics and Informatics Reports*, Vol. 10, p. 100049.

[11] Gad-Elrab, A. A., Alsharkawy, A. S., Embabi, M. E., Sobhi, A., & Emara, F. A. (2024) "Adaptive multi-criteria-based load balancing technique for resource allocation in fog-cloud environments," *International Journal of Computer Networks & Communications (IJCNC)*, Vol. 16, No. 1.

[12] Alsadie, D., (2024) "Advancements in heuristic task scheduling for IoT applications in fog-cloud computing: challenges and prospects",*PeerJ Computer Science*, Vol. 10, p. e2128.
[13] Bhatt, R., Mehra, R., & Upreti, K. (2026) "An adaptive hybrid scheduling approach for sustainable and reliable cloud services", *International Journal of Computer Networks & Communications (IJCNC)*, Vol. 18, No. 2.
[14] Goel, G., & Tiwari, R., (2023) "Resource scheduling techniques for optimal quality of service in fog computing environment: a review",*Wireless Personal Communications*, Vol. 131, No. 1, pp. 141-164.
[15] Buyya, R., Ilager, S., & Arroba, P., (2024) "Energy-efficiency and sustainability in new generation cloud computing: a vision and directions for integrated management of data centre resources and workloads",*Software: Practice and Experience*, Vol. 54, No. 1, pp. 24-38.
[16] Saif, F. A., Latip, R., Hanapi, Z. M., &Shafinah, K., (2023) "Multi-objective grey wolf optimizer algorithm for task scheduling in cloud-fog computing",*IEEE Access*, Vol. 11, pp. 20635-20646.
[17] Ghafari, R., & Mansouri, N., (2023) "E-AVOA-TS: enhanced African vultures optimization algorithm-based task scheduling strategy for fog–cloud computing",*Sustainable Computing: Informatics and Systems*, Vol. 40, p. 100918.
[18] Liu, Q., Kosarirad, H., Meisami, S., Alnowibet, K. A., & Hoshyar, A. N., (2023) "An optimal scheduling method in IoT-fog-cloud network using combination of aquila optimizer and African vultures optimization",*Processes*, Vol. 11, No. 4, p. 1162.
[19] Bhakhar, R., &Chhillar, R. S., (2024) "Dynamic multi-criteria scheduling algorithm for smart home tasks in fog-cloud IoT systems",*Scientific Reports*, Vol. 14, No. 1, pp. 1-37.
[20] Khezri, E., Yahya, R. O., Hassanzadeh, H., Mohaidat, M., Ahmadi, S., & Trik, M., (2024) "DLJSF: data-locality aware job scheduling IoT tasks in fog-cloud computing environments",*Results in Engineering*, Vol. 21, p. 101780.
[21] Salehnia, T., Seyfollahi, A., Raziani, S., Noori, A., Ghaffari, A., Alsoud, A. R., &Abualigah, L., (2024) "An optimal task scheduling method in IoT-fog-cloud network using multi-objective moth-flame algorithm", *Multimedia Tools and Applications*, Vol. 83, No. 12, pp. 34351-34372.
[22] Behera, I., &Sobhanayak, S., (2024)"Task scheduling optimization in heterogeneous cloud computing environments: a hybrid GA-GWO approach",*Journal of Parallel and Distributed Computing*, Vol. 183, p. 104766.
[23] Srichandan, S. K., Majhi, S. K., Jena, S., Mishra, K., & Bhat, R., (2024)"A secure and distributed placement for quality of service-aware IoT requests in fog-cloud of things: a novel joint algorithmic approach",*IEEE Access*, Vol. 12, pp. 56730-56748.
[24] Mahapatra, A., Pradhan, R., Majhi, S. K., & Mishra, K., (2025)"DELTa: dynamic energy-and-latency-aware task scheduling for fog-cloud paradigm",*IEEE Access*, Vol. 13, pp. 74617-74633.
[25] Rateb, R., Hadi, A. A., Tamanampudi, V. M., Abualigah, L., Ezugwu, A. E., Alzahrani, A. I., ... & Jia, H., (2025) "An optimal workflow scheduling in IoT-fog-cloud system for minimizing time and energy",*Scientific Reports*, Vol. 15, No. 1, p. 3607.
[26] Kaur, A., Kumar, S., Gupta, D., Hamid, Y., Hamdi, M., Ksibi, A., ... & Saini, S., (2023) "Algorithmic approach to virtual machine migration in cloud computing with updated SESA algorithm",*Sensors*, Vol. 23, No. 13, p. 6117.
[27] Singh, S., & Singh, D., (2023)"A bio-inspired VM migration using re-initialization and decomposition based-whale optimization",*ICT Express*, Vol. 9, No. 1, pp. 92-99.
[28] Swarnakar, S., Banerjee, C., Basu, J., & Saha, D., (2023)"A multi-agent-based VM migration for dynamic load balancing in cloud computing cloud environment",*International Journal of Cloud Applications and Computing*, Vol. 13, No. 1, pp. 1-14.
[29] Yao, W., Wang, Z., Hou, Y., Zhu, X., Li, X., & Xia, Y., (2023) "An energy-efficient load balance strategy based on virtual machine consolidation in cloud environment", *Future Generation Computer Systems*, Vol. 146, pp. 222-233.
[30] Radi, M., Alwan, A. A., & Gulzar, Y., (2023) "Genetic-based virtual machines consolidation strategy with efficient energy consumption in cloud environment",*IEEE Access*, Vol. 11, pp. 48022-48032.
[31] Wu, J., Yang, W., Han, X., Qiu, Y., Gudkov, A., & Song, J., (2024)"Hotspot resolution in cloud computing: a Γ-robust knapsack approach for virtual machine migration",*Journal of Parallel and Distributed Computing*, Vol. 186, p. 104817.
[32] Alsadie, D., &Alsulami, M., (2024) "Efficient resource management in cloud environments: a modified feeding birds algorithm for VM consolidation",*Mathematics*, Vol. 12, No. 12, p. 1845.

[33] Liu, C., Ma, L., Zhang, M., & Long, H., (2025)"Optimizing cloud resource management with an IoT-enabled optimized virtual machine migration scheme for improved efficiency",*Journal of Network and Computer Applications*, Vol. 237, p. 104137.
[34] Siruvoru, V., & Aparna, S. (2025)"Harmonic migration algorithm for virtual machine migration and switching strategy in cloud computing",*Concurrency and Computation: Practice and Experience*, Vol. 37, No. 1, p. e8320.
[35] Archana, & Kumar, N., (2025) "A modified bat mechanism for virtual machine migration in a cloud environment",*SN Computer Science*, Vol. 6, No. 1, p. 74.

AUTHORS

Parveen Taj M is currently working as an Assistant Professor in the Department of Computer Science at Dr. G.R.D College of Science, Coimbatore. She previously served as an Assistant Professor in the Department of Computer Science at Sri Jayendra Saraswathy Maha Vidyalaya College of Arts & Science, Coimbatore, from 2005 to 2024. She is pursuing a Ph.D. in Computer Science (part-time) at PPG College of Arts & Science and has completed an M.Phil. in Computer Science with a specialization in Advanced Networking. She has guided 15 M.Phil. research scholars in the areas of Data Warehousing and Mining, and Advanced Networking.

Dr. Muthumani N received her Ph.D. in Computer Science from Mother Teresa Women's University, Kodaikanal, and has qualified For the UGC-NET in Computer Science, demonstrating her strong commitment to academic excellence and research. She currently serves as the Principal of PPG College of Arts and Science, Coimbatore, Tamil Nadu, a role she has held since November 2020. Previously, she served as Professor and Head at Sri Ramakrishna College of Arts and Science (2008–2020), Assistant Professor at KG College of Arts and Science (2005–2008), and Lecturer at RVS College of Arts and Science (2000–2005). Her areas of expertise include academic administration, accreditation processes, curriculum design, and faculty development. Dr. Muthumani has published more than 25 research articles in reputed journals, authored 8 books, and contributed 5 chapters to edited volumes. She has presented over 20 research papers at national and international conferences and has served as a resource person in more than 100 faculty development programs, inspiring and mentoring educators across the academic spectrum.