An Improved Dingo Optimization for Resource Aware Scheduling in Cloud Fog Computing Environment

M. Santhosh Kumar[1], Ganesh Reddy Karri[1]
1- VIT-AP University, Amaravathi, A.P., India, 522 237.
Email: santhosh.21phd7113@vitap.ac.in (Corresponding author)
Email: guncity11@gmail.com

Received: 11 May 2023 Revised: 17 June 2023 Accepted: 5 July 2023

ABSTRACT:
Task scheduling in Cloud-Fog computing environments is a critical aspect of optimizing resource allocation and enhancing performance. This study presents an improved version of the Dingo Optimization Algorithm (IDOA) specifically designed for task scheduling in Cloud-Fog computing. The enhanced IDOA incorporates novel modifications to address the limitations of the original algorithm and improve the efficiency and effectiveness of task allocation. The algorithm incorporates modifications to the fitness evaluation function, a dynamic update mechanism, and a neighborhood search technique to enhance task allocation efficiency. Extensive simulations and comparisons with existing algorithms are conducted to evaluate the performance of the IDOA. The results demonstrate its superiority in terms of task makespan time, VM failure rate, and degree of imbalance. Overall, the improved Dingo Optimization Algorithm offers a promising solution for efficient task scheduling in Cloud-Fog computing environments. The algorithm effectively balances exploration and exploitation, facilitating efficient task scheduling in Cloud-Fog computing environments and optimizing cloud-based applications and services.

KEYWORDS: Task scheduling, Cloud-Fog Computing, Makespan, Degree of Imbalance, Dingo Optimization Algorithm.

1. INTRODUCTION

Cloud-Fog computing has emerged as a promising paradigm that integrates cloud computing and edge computing to meet the demands of modern applications and services. Task scheduling plays a crucial role in optimizing resource allocation and enhancing the performance of such complex and dynamic computing environments [1,2]. The IDOA, inspired by the hunting behavior of dingoes, is a population-based metaheuristic algorithm that mimics the collaboration and coordination observed in a pack of dingoes. Its application to task scheduling aims to optimize resource allocation and enhance the overall system performance. Efficient task scheduling in Cloud-Fog computing requires intelligent algorithms that can adapt to the dynamic nature of resources and effectively allocate tasks to appropriate computing nodes [3].

In this research paper, we propose an improved version of the Dingo Optimization Algorithm (IDOA) specifically tailored for task scheduling in Cloud-Fog computing environments. The IDOA, inspired by the collaborative hunting behavior of dingoes, is a metaheuristic algorithm that has shown effectiveness in solving various optimization problems. By enhancing the IDOA, we aim to address the unique challenges of task scheduling in Cloud-Fog computing and provide an efficient solution to optimize resource utilization and task allocation.

The improved DOA incorporates several enhancements to overcome the limitations of the original algorithm and improve its performance in the context of task scheduling. These enhancements include a modified fitness evaluation function, a dynamic update mechanism, and a neighbourhood search technique. The modified fitness evaluation function takes into account factors such as task deadlines, resource availability, and communication costs between Fog nodes and Cloud data centres [4]. This comprehensive evaluation enables the algorithm to make informed decisions when assigning tasks to suitable computing resources.

The dynamic update mechanism in the improved DOA strikes a balance between exploration and exploitation by adapting the search strategies based on the current state of the system. This adaptability allows the algorithm to effectively explore new solution spaces while also exploiting the best solutions discovered so far [5]. Additionally, the incorporation of a neighbourhood search technique enhances the local search capabilities

Paper type: Research paper
DOI: 10.30486/mjee.2023.1989335.1165
How to cite this paper: M. Santhosh Kumar, G. Reddy Karri, **"An Improved Dingo Optimization for Resource Aware Scheduling in Cloud Fog Computing Environment"**, *Majlesi Journal of Electrical Engineering*, Vol. 17, No. 3, pp. 31-41, 2023.

of the algorithm, enabling it to refine promising solutions within the vicinity of the current solution [6].
To evaluate the performance of the improved DOA, extensive simulations and comparisons with existing task scheduling algorithms are conducted. The results demonstrate the superiority of the proposed algorithm in terms of task completion time, resource utilization, and energy consumption. By leveraging the collaborative hunting behavior of dingoes and incorporating these enhancements, the improved DOA offers a promising solution for efficient and effective task scheduling in Cloud-Fog computing environments.

In summary, this research paper presents an improved Dingo Optimization Algorithm for task scheduling in Cloud-Fog computing. By addressing the unique challenges of this computing paradigm and leveraging the algorithm's inherent capabilities, we aim to optimize resource allocation, improve task scheduling efficiency, and ultimately enhance the performance of cloud-based applications and services in Cloud-Fog computing environments.
The essential contributions of this manuscript are as follows,

- We present an improved evolutionary method called IDOA for scheduling tasks well in cloud and fog scenarios. This algorithm is the basic inspiration of DOA adaptive strategy to improve the speed of convergence and the number of searches.
- Makespan time, degree of imbalance, and VM failure rate are three competing factors that must be considered when making an efficient model for Internet of Things (IoT) tasks that are sent to a cloud-fog framework to be processed.
- Comparing the makespan time, degree of imbalance, and VM failure rate for the suggested scheduling method with those of other strategies used on real workloads.

The remaining portion of the manuscript is organized as follows: Section 2 provides a comprehensive review of related literature, Section 3 describes the research methodologies used, Section 4 describes the experiments conducted, and also provides a brief discussion about limitations. Finally, the paper is concluded in section 5.

2. LITERATURE SURVEY

Task scheduling in Cloud-Fog computing environments is a critical research area due to the dynamic nature of resources and the need for efficient task allocation. One promising approach to address this challenge is the application of metaheuristic algorithms, such as the Improved Dingo Optimization Algorithm (IDOA), which has gained attention in recent years. In this literature study, we explore the existing research on the utilization of the IDOA in task scheduling within the context of Cloud-Fog computing. The literature reveals several studies that have extended and improved the DOA specifically for task scheduling in Cloud-Fog computing environments.

In this literature study, we come up with a detailed analysis of scheduling parameters and techniques used in previous studies shown in Table 1, we categorize makespan, energy consumption, and SLA-based trust parameters among the many parameters and techniques used by previous investigations. We developed the IDOA, which is the improved technique of dingo optimization method to account for makespan time, VM failure rate, and degree of imbalance to simplify the scheduling of efficient operations in a cloud fog environment. Latency-sensitive applications, such as those involving vehicular networks, and real-time applications, such as smart cities, will benefit significantly from the suggested strategy.

Table 1. Detailed analysis of scheduling parameters and techniques used in cloud fog environment.

Authors	Year	Technique used	Parameters	Limitations
Aravind, Kalavagunta, et al. [7]	2022	SDO-BM	Security, distance, Quality of Service, delay, overhead, trust, energy usage	In this study, author was unable to extend the work with Social IoT and Multiple IoT scenarios
Pari, Deepanramkumar, et al. [8]	2022	6GCRN–IoCV	Throughput and packet delivery ratio	The author used hybrid beams which did not give the accurate result
Yin, Zhenyu, et al. [9]	2022	HMA	Task completion rate, energy consumption	The author did not use hybrid technique
Ahmed, Omed Hassan, et al. [10]	2021	DMFO-DE	Total VM’s, makespan, energy consumption	In this study author was unable to address multi objective with hybrid optimization algorithm
Badri, Sahar, et al. [11]	2023	CNN-MBO	Resource utilization, response time, energy consumption	The author did not work on real-time environment to determine the effectiveness of the system

Harika, Sonti, and B. Chaitanya Krishna. [12]	2022	HM-DA	Cost, credibility score, concurrency, task time computation	In this study proposed approach had a good result in terms of cost only. Other metrics need to improve in the future
Kumar, M. Santhosh, et al. [13]	2023	EEOA	Cost, makespan, energy consumption	In this study, the author was unable to measure his approach in edge-cloud environment
Gomathi, B., et al. [14]	2021	MOIMBO	Makespan, maximize reliability, resource utilization	The author did not extend the problem to minimize energy consumption.
Najafizadeh, Abbas, et al. [15]	2022	MOSA	Service delay time, service cost, access level control, deadline constraint	The author did not give much importance to protecting users' personal information, taking the scheduling of workflows into account, or creating scheduling algorithms with parameters for variables in this work.
Nguyen, Binh Minh, et al. [16]	2019	TCaS	Makespan, fitness value	The author here focused on just one statistic, rather than a wide range of parameters like runtime, data transfer rates, hardware requirements, and energy usage.
Huang, Xingwang, et al. [17]	2020	PSO	Makespan	Workflow applications operating in cloud and fog environments have various optimization objectives, such as load balancing and usage of energy, that the author does not consider
Movahedi, Zahra, et al. [18]	2021	OppoCWOA	Time and fog energy consumption.	The author did not investigate data privacy in the future using the blockchain-based FogBus framework
Singh, Gyan, and Amit K. Chaturvedi. [19]	2023	MPSO	Makespan, cost, energy consumption	This study was unable to cover the several data centres for expanding in various scenarios
Saravanan, T., and S. Saravanakumar. [20]	2022	SCCSO	Response time, scheduling time, load balancing rate, delay time	The author was unable to address resource management in fog-IoT environments
Jakwa, Ali Garba, et al. [21]	2023	MPSO-SPT	Energy consumption, resource utilization and response time,	This work is not implemented in iFogsim and also did not consider execution time, and execution cost these are the important metrics in this study
Liu, Weimin, et al. [22]	2023	PGABC	Energy consumption and delay	This work was unable to address multiple metrics
Abd Elaziz, Mohamed, et al. [23]	2019	MSDE	Makespan, degree of imbalance, CPU time	Multi-objective version of workflow scheduling is missing in this study
Mangalampalli, Sudheer, et al. [24]	2023	TAFFA	Makespan, SLA-based trust parameters.	The author was unable to use an artificial intelligence technique
Awad, A. I., et al. [25]	2015	LBMPSO	Load balancing, transmission delay, transmission cost, make-span, and execution time	In this study the technique used foe scheduling is unable to reschedule the failure tasks

Moon, YoungJu, et al. [26]	2017	SACO	Makespan, processing time,	In this work, author did not consider heterogeneous clusters because increasing cost leads to more expensive
Almezeini, Nora, and Alaaeldin Hafez. [27]	2017	LOA	Makespan, cost, average utilization, degree of imbalance	In this work, the amount of cost is to be increased because different cost models are not used
Singhal, Shweta, et al. [28]	2023	ABO	Size, runtime and overall execution time	The author did not test the bigger industrial programs with different programming languages
Supreeth, S., et al. [29]	2022	ESSOA	Cost, degree of imbalance, makespan time, resource utilization.	In this study, author has not used the advanced optimization technique, it leads to security limitations in scheduling and task allocation

3. SYSTEM MODEL AND PROBLEM FORMULATION

3.1. System model

With the rise of real-time applications and the need for low-latency processing, the integration of cloud computing with fog and edge computing has become increasingly important. To address the challenges of task scheduling in such complex environments, we propose a system model that encompasses the Cloud-Fog-Edge architecture shown in Fig. 1. This model provides a hierarchical structure that consists of the cloud layer, fog layer, and edge layer, enabling efficient task scheduling and resource allocation. The Improved Dingo Optimization Algorithm (IDOA), inspired by the hunting behavior of dingoes, is integrated into the task scheduling process to optimize resource utilization and enhance overall system performance.

Cloud Layer: The cloud layer serves as the central computing and storage infrastructure in the system model. It consists of powerful data centres with abundant resources. At the cloud layer, the DOA is employed to handle the scheduling of computationally intensive tasks that require substantial computing power and storage capabilities. The IDOA considers various factors such as task priorities, resource availability, and load balancing, aiming to minimize task completion time and maximize resource utilization within the cloud layer. Fog Layer: The fog layer acts as an intermediate tier between the cloud and edge layers, located closer to the end-users or devices. It comprises fog nodes or gateways that offer computing and storage capabilities. In the system model, the IDOA operates at the fog layer to handle task scheduling for latency-sensitive and bandwidth-intensive tasks. The IDOA takes into account factors such as communication costs, network congestion, and data locality to allocate tasks efficiently within the fog layer. This helps reduce network latency, improve response times, and offload computation from the cloud layer.

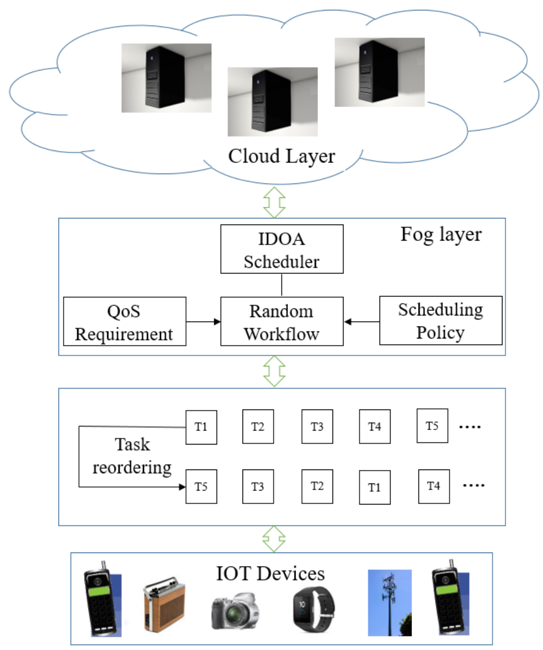

Fig. 1. System architecture.

Edge Layer: The edge layer represents the devices and sensors situated at the network's edge, such as IoT devices and mobile devices. These devices have limited computing resources and are responsible for generating data and performing initial data processing. In the system model, the IDOA operates at the edge layer to schedule lightweight and time-sensitive tasks. The IDOA considers factors such as device capabilities, energy constraints, and task dependencies to allocate tasks effectively within the edge layer. This enables real-time processing, reduces communication overhead, and supports edge computing capabilities.

$$T = T_1, T_2, \ldots, T_n \tag{1}$$

The proposed system model, incorporating the Dingo Optimization Algorithm, offers an effective approach for task scheduling in Cloud-Fog computing environments. The hierarchical structure of the cloud, fog, and edge layers, along with the IDOA, optimizes resource allocation and task scheduling. The integration of the IDOA enables intelligent decision-making in task allocation based on various factors, including task requirements, resource availability, and communication costs. This research contributes to efficient resource utilization, improved task scheduling, and enhanced system performance in Cloud-Fog computing environments.

$$T_{nodes} = C_{nodes} + F_{nodes} \tag{2}$$

Where T denotes the total nodes which are equivalent to the cloud nodes and fog nodes which denotes the C and F.

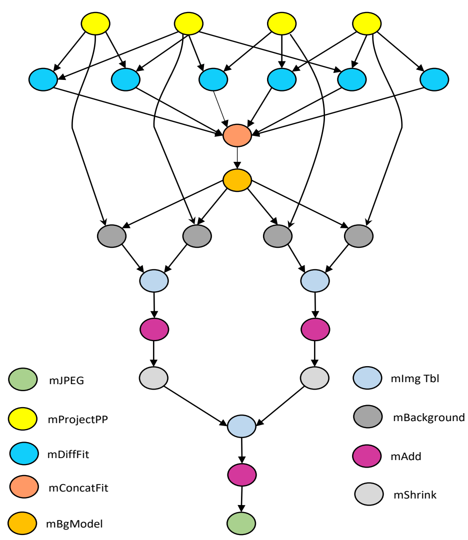

Fig. 2. Random workflow scheduling.

In task scheduling within cloud fog computing, the distribution of tasks is managed by a workflow system that facilitates the efficient allocation of tasks to suitable resources. The workflow system follows a two-step process to distribute tasks effectively.

Firstly, the system performs task modelling and scheduling shown in Fig. 2. It identifies the tasks that need to be executed and creates a workflow graph that represents the dependencies between tasks. The system considers factors such as task requirements, resource availability, and optimization objectives to schedule the tasks. This involves employing various scheduling algorithms, such as heuristic-based algorithms or optimization algorithms like the Dingo Optimization Algorithm, to assign tasks to appropriate resources. The scheduling algorithm ensures that constraints, such as task dependencies and resource limitations, are taken into account to optimize task execution.

Secondly, the system executes and monitors the assigned tasks. It initiates the execution of tasks on the allocated resources and monitors their progress and status. The workflow system manages data transfer and communication between tasks and resources, considering data locality, network latency, and communication costs. It also dynamically adapts to changes in the cloud fog computing environment, such as resource failures or workload variations, by reallocating tasks or adjusting the schedule to maintain optimal resource utilization and meet performance requirements.

By following the abovementioned two-step process, the workflow system in cloud fog computing efficiently distributes tasks, ensuring that they are assigned to appropriate resources while considering dependencies, optimizing resource utilization, and adapting to dynamic changes. This allows for the effective execution of tasks and enhances the overall performance of task scheduling in cloud fog computing environments.

3.2. Problem Formulation

Table 2 depicts the mathematical modelling notations used in the problem formulation.

Table 2. Mathematical modelling notations.

Notation used	Meaning
T	Total number of tasks
T_1	First task
T_n	Nth task
C_{nodes}	Cloud nodes
F_{nodes}	Fog nodes
MKS	Makespan
Cl	Cloudlet
VM	Virtual machine
DOI	Degree of imbalance
T_{max}	Maximum time of task execution
T_{min}	Minimum time of task execution
T_{avg}	Average time of task execution
V_i^n	Resource utilization

3.2.1. Makespan

In task scheduling, the makespan represents the total time required to complete the execution of all tasks in a schedule. The makespan calculation in the context of the Improved Dingo Optimization Algorithm (IDOA) involves considering the start and finish times of each task and determining the maximum completion time among all tasks.

$$MKS = \max(Cl_{VM1}, Cl_{VM2}, \ldots, Cl_{VMn}) \tag{3}$$

The above equation demonstrates that the MKS denotes the total amount of makespan time and Cl denotes the cloud lets and VM denotes the virtual machines, calculate the makespan for a given task schedule generated by the Improved DOA, follow these steps:

1. Initialize the makespan variable to 0.
2. For each task in the schedule:
   a. a. Determine whenever to start the task by examining the time its dependencies are due to end. The beginning time is equal to the earliest completion time of any dependent tasks.
   b. Add the execution time to the start time to get the total time required to complete the operation.
   c. If the current task's completion time is going to be later than the makespan, the makespan should be set to the completion time of the current task.
3. After processing all tasks in the schedule, the makespan variable will hold the maximum completion time among all tasks, representing the makespan of the schedule.

The makespan calculation considers task dependencies, ensuring that dependent tasks are scheduled after their prerequisite tasks have completed execution. It also considers the execution times of the tasks, as well as any delays caused by resource availability or communication costs.

By calculating the makespan, the efficiency of the task schedule generated by the Improved Dingo Optimization Algorithm can be evaluated. The goal is to minimize the makespan, indicating efficient task allocation and resource utilization in Cloud-Fog computing environments.

3.2.2. Degree of imbalance

The degree of imbalance in task scheduling refers to the extent of load imbalance among the resources in the Cloud-Fog computing environment. In the context of the Improved Dingo Optimization Algorithm (IDOA), the degree of imbalance can be evaluated by measuring the difference in the total workload assigned to each resource.

To calculate the degree of imbalance for a given task schedule generated by the Improved DOA, you can follow these steps:

$$DOI = \frac{T_{\max} - T_{\min}}{T_{avg}} \tag{4}$$

The above equation demonstrates that the DOI denotes the degree of imbalance and here T_{max} represents the maximum time of the execution time and T_{min} denotes the minimum time of the execution time and T_{avg} represents the average time of the execution time

1. Calculate the total workload assigned to each resource in the schedule by summing up the execution times of the tasks assigned to each resource.
2. Calculate the average workload across all resources by dividing the total workload by the number of resources.
3. Calculate the degree of imbalance as the maximum absolute deviation of the workload of any resource from the average workload.

- For each resource, calculate the absolute difference between its workload and the average workload.
- Identify the resource with the maximum absolute difference.
- This maximum absolute difference represents the degree of imbalance in the task schedule.

A higher degree of imbalance indicates a greater disparity in the workload assigned to different resources. It suggests that some resources may be heavily loaded while others are underutilized. Conversely, a lower degree of imbalance indicates a more balanced distribution of workload among the resources.

Evaluating the degree of imbalance helps assess the effectiveness of the Improved Dingo Optimization Algorithm in achieving load balancing in task scheduling. The objective is to minimize the degree of imbalance, aiming for a more even distribution of workload among the resources in the Cloud-Fog computing environment. This can enhance system performance, improve resource utilization, and mitigate the potential for bottlenecks or resource underutilization.

3.2.3. VM failure rate

The VM failure rate is not typically the primary objective of the Improved Dingo Optimization Algorithm in task scheduling. However, it can still be considered as a secondary objective or a constraint in the problem formulation. The VM failure rate represents the rate at which virtual machines (VMs) experience failures during the execution of tasks in the Cloud-Fog computing environment.

$$\sum_{i=1}^{n} V_i^n \times w_{in} \le s_k^n \tag{5}$$

In Eq. (5), where V_i^n signifies resource utilization of the i^{th} VM and R specifies resources like CPU denotes C and memory shortly notated as M, and bandwidth denotes as BW, respectively, for placement of the i^{th} VM denotes Vi at the k^{th} server S_k, we see the important limitations which need to be met prior every virtual machine (VM) installation and migrating. To incorporate the VM failure rate as a secondary objective or a constraint in the Improved Dingo Optimization Algorithm, the objective function can be modified as follows:

Objective Function of the VM failure rate is to minimize Makespan, subject to VM Failure Rate Constraints. Mathematically, the objective function with the VM failure rate constraint can be represented as:
Minimize: makespan Subject to: VM Failure Rate ≤ Maximum Allowable Failure Rate

In this formulation, the primary objective remains to minimize the makespan, while the VM failure rate constraint ensures that the algorithm considers the failure rate of VMs during task scheduling. The constraint sets an upper limit on the VM failure rate to maintain the desired level of system reliability.

By incorporating the VM failure rate constraint, the Improved Dingo Optimization Algorithm can take into account the trade-off between makespan minimization and maintaining a low failure rate of VMs. This helps in achieving a balanced task schedule that optimizes resource utilization, minimizes makespan, and maintains an acceptable level of VM reliability in Cloud-Fog computing environments.

3.2.4. Proposed IDOA Algorithm

The step-by-step procedure of Improved Dingo Optimization Algorithm in Task Scheduling for Cloud-Fog Computing:

Step 1. Initialize the population of dingo solutions with random task schedules.

Step 2. Set the maximum number of iterations and termination criteria (e.g., reaching a certain fitness threshold or a maximum number of iterations).

Step 3. Initialize the best solution and set its fitness to a high value.

Step 4. Initialize the fitness of each solution in the population.

Step 5. Repeat until the termination criteria are met or the maximum number of iterations is reached:

a. Perform the neighborhood search operation on each solution in the population.
b. Evaluate the fitness of each solution based on the makespan and constraints.
c. Update the best solution if a new best solution is found.
d. Update the pheromone levels based on the fitness of each solution.
e. Update the dynamic parameters based on the current iteration.
f. Repeat for each solution in the population.
i. Perform the local search operation on the solution.
ii. Perform the global search operation by updating the solution based on pheromone levels and exploration/exploitation factors.
iii. Evaluate the fitness of the updated solution.
iv. Update the solution if the updated solution has better fitness.

Step 6. Return the best solution obtained.

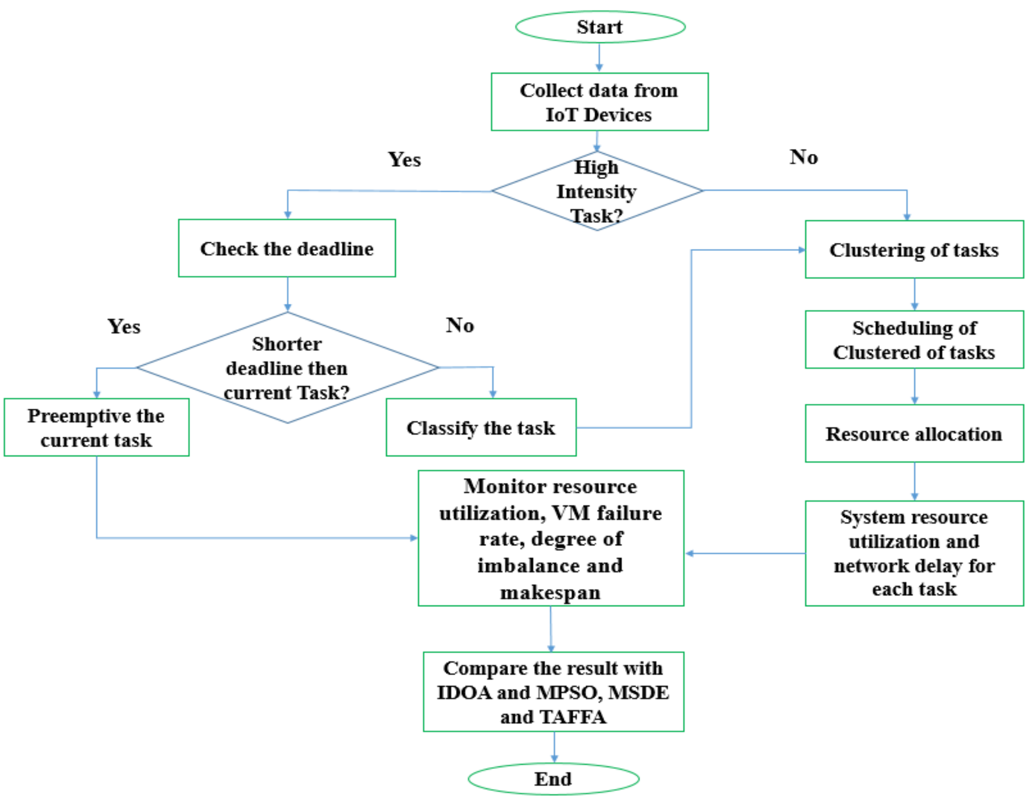

Fig. 3. Proposed IDOA flowchart.

4. RESULTS AND DISCUSSION

4.1. Results

We implemented the proposed method to work by using the cloud environment simulation software CloudSim [30]. The use of simulation has allowed for the execution of several experiments. To demonstrate the efficacy of the IDOA approach, we give comparative evaluations with the widely used MPSO, MSDE, and TAFFA algorithms.

An Intel Core i5-3373U processor at 1.8 GHz and 6 GB of RAM were used in our simulations for IDOA's results. In order to evaluate the IDOA algorithm's effectiveness in terms of both cost and makespan, we implemented it in CloudSim Toolkit and examined the results. Tasks that depended on the outcomes of other tasks were excluded from the study. Transmission speeds over the lines were predicted to fall into a normal distribution, offering speeds between 40 Mbps and 10,000 Mbps. In Table 3, we can see the simulation analysis parameters. The table outlines the required number of tasks, virtual machines (VMs), and data centres to run the simulation.

Table 3. Cloudsim toolkit parameters.

Parameter	Cloud	Fog
Total VMs	[15,20,25]	[20,25,30]
Computing power (MIPS)	[2000:4000]	[1000:2000]
RAM (MB)	[10000:20000]	[500:5000]
Bandwidth (Mbps)	[1024:4096]	[128:1024]

4.1.1. Makespan

As shown in Fig. 4, which indicates the best makespan results for real-time workloads with varying task sizes, the presented IDOA approach efficiently achieves the highest possible makespan when compared to all different existence methods such as MPSO, MSDE, and TAFFA algorithms. IDOA is found to be an excellent method of addressing crucial problems in cloud fog task allocation scheduling.

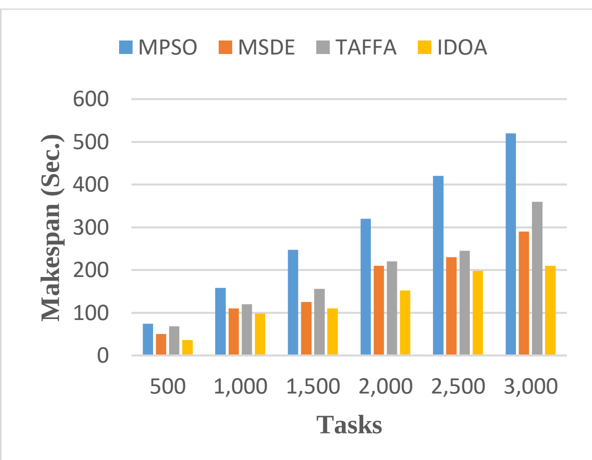

Fig. 4. Proposed approach makespan calculation.

4.1.2. Degree of imbalance

The comparative results depicted in Fig. 5 highlight the superiority of the presented IDOA approach in addressing the workload distribution challenges in cloud fog task allocation scheduling. Compared to existing methods such as MPSO, MSDE, and TAFFA algorithms, IDOA consistently achieves a significantly lower degree of imbalance among allocated tasks across resources. The graph clearly illustrates that IDOA outperforms the other methods by effectively minimizing the workload disparities and ensuring a more balanced distribution of tasks. These findings emphasize the effectiveness of IDOA in optimizing resource utilization and mitigating performance bottlenecks caused by uneven task allocation. Consequently, IDOA emerges as a robust solution for enhancing the degree of balance in cloud fog task scheduling and improving the overall system performance in terms of workload distribution.

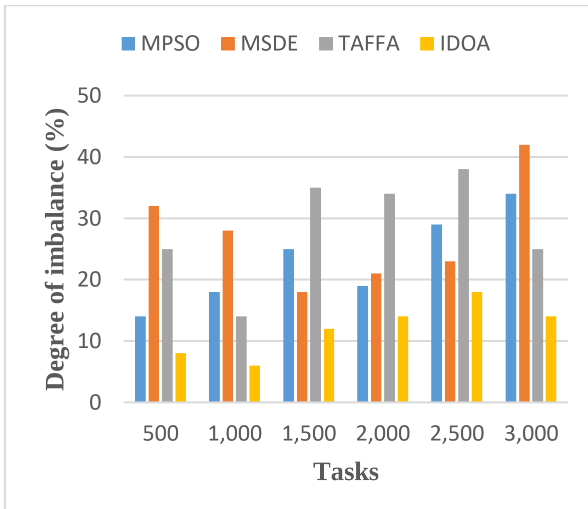

Fig. 5. Proposed approach degree of imbalance calculation.

4.1.3. VM failure rate

The comparative results presented in Fig. 6 demonstrate the superior performance of the IDOA approach in mitigating VM failures in cloud fog task allocation scheduling. When compared to existing methods such as MPSO, MSDE, and TAFFA algorithms, IDOA consistently exhibits a significantly lower VM failure rate. The graph clearly illustrates that IDOA outperforms the other methods by effectively reducing the occurrence of VM failures and improving the reliability of the system. These findings highlight the robustness and effectiveness of IDOA in optimizing task allocation, resource utilization, and fault tolerance mechanisms. By minimizing VM failures, IDOA ensures improved system stability and availability, thereby enhancing the overall performance of cloud fog task scheduling in terms of VM failure rate.

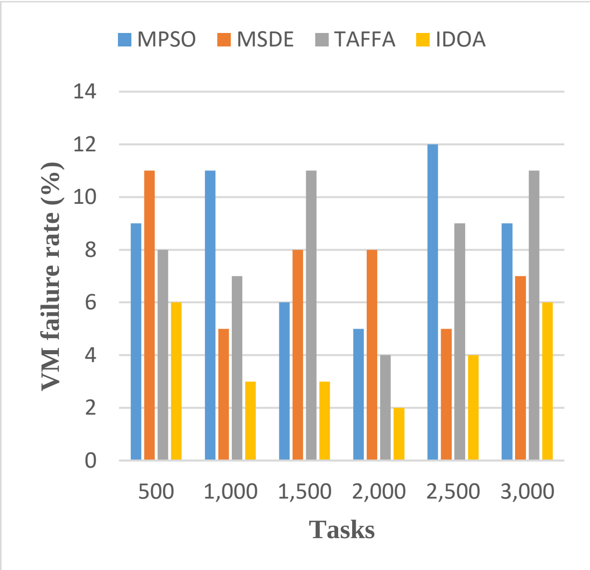

Fig. 6. Proposed approach VM failure rate.

4.2. Discussion

In task scheduling in cloud fog computing using the Improved Dingo Optimization Algorithm (IDOA), the consideration of makespan time, VM failure rate, and degree of imbalance is crucial. The IDOA aims to minimize the makespan time by optimizing task allocation and resource utilization, leading to improved task completion times and overall system performance. Additionally, by balancing the distribution of tasks across resources, the IDOA helps mitigate the degree of imbalance, ensuring efficient resource utilization and reducing performance bottlenecks. While the IDOA primarily focuses on task scheduling, its optimization capabilities indirectly impact the VM failure rate by considering resource availability and fault tolerance mechanisms. By minimizing makespan time, reducing the degree of imbalance, and potentially mitigating VM failures, the IDOA contributes to enhanced task scheduling in cloud fog computing environments. Further research is needed to assess the specific impact of the IDOA on these factors and to evaluate its effectiveness in different cloud fog computing scenarios.

4.2.1. Limitations

While the Improved Dingo Optimization Algorithm (IDOA) can contribute to improving makespan time, VM failure rate, and degree of imbalance in task scheduling for cloud fog computing, it is important to consider some potential limitations:

1. Makespan Time

- The effectiveness of IDOA in reducing makespan time depends on the problem complexity, task dependencies, and resource availability. In highly dynamic and complex environments, achieving optimal makespan may still pose challenges.
- The IDOA's performance heavily relies on the efficiency of the objective function and fitness evaluation. Inaccurate estimations or limitations in the fitness function may affect the algorithm's ability to effectively minimize makespan time.

2. VM Failure Rate

- The IDOA, as a task scheduling algorithm, primarily focuses on optimizing resource allocation and task scheduling. It may not directly address all aspects related to VM failure rates. Additional mechanisms or techniques specific to fault tolerance and VM management are required to further mitigate VM failures.
- The effectiveness of IDOA in reducing VM failure rate also depends on the fault tolerance capabilities of the cloud fog computing infrastructure and the resiliency measures implemented at the VM level.

3. Degree of Imbalance

- While IDOA aims to achieve load balancing and reduce the degree of imbalance, there might still be inherent limitations in completely eliminating imbalance due to the dynamic nature of workload and resource availability.
- The performance of IDOA in load balancing heavily relies on the accuracy of workload estimation, task allocation mechanisms, and the ability to handle unexpected variations in resource demands.

It is crucial to consider these limitations when applying the IDOA to task scheduling in cloud fog computing. While the algorithm can contribute to improving makespan time, reducing VM failure rate, and mitigating the degree of imbalance, additional considerations and complementary approaches may be required to fully address these challenges in real-world cloud fog computing environments.

5. CONCLUSIONS AND FUTURE WORK

In conclusion, the application of the Improved Dingo Optimization Algorithm (IDOA) in task scheduling for cloud fog computing shows promise in improving system performance and resource utilization. By minimizing makespan time, optimizing task allocation, and mitigating the degree of imbalance, the IDOA contributes to efficient task execution and load balancing. However, it is important to consider the limitations of the IDOA, such as its dependence on accurate estimations, fitness function quality, and the need for additional fault tolerance mechanisms to address VM failure rates. Further research and experimentation are necessary to evaluate the IDOA's performance in different cloud fog computing scenarios and to explore complementary approaches that can address the challenges associated with makespan time, VM failure rate, and degree of imbalance more comprehensively. Overall, the IDOA offers valuable insights and serves as a foundation for enhancing task scheduling in cloud fog computing, but further advancements are required to fully harness its potential in real-world environments.

6. ACKNOWLEDGMENT

I would like to express my sincere gratitude to my Ph.D. research supervisor Dr. Ganesh Reddy Karri for his valuable and constructive suggestions during the planning and development of this research work.

Financially, this work does not support any organization/funding agency.

REFERENCES

[1] Arunarani, A. R., Dhanabalachandran Manjula, and Vijayan Sugumaran. **"Task scheduling techniques in cloud computing: A literature survey."** *Future Generation Computer Systems* 91 (2019): 407-415.

[2] Houssein, Essam H., et al**. "Task scheduling in cloud computing based on meta-heuristics: review, taxonomy, open challenges, and future trends."** *Swarm and Evolutionary Computation* 62 (2021): 100841.

[3] Soltani, Nasim, Behzad Soleimani, and Behrang Barekatain. **"Heuristic algorithms for task scheduling in cloud computing: a survey."** *International Journal of Computer Network and Information Security* 11.8 (2017): 16.

[4] Singh, Sukhpal, and Inderveer Chana. **"A survey on resource scheduling in cloud computing: Issues and challenges."** *Journal of grid computing* 14 (2016): 217-264.

[5] Menaka, M., and KS Sendhil Kumar. **"Workflow scheduling in cloud environment–Challenges, tools, limitations & methodologies: A review."** *Measurement: Sensors* (2022): 100436.

[6] Jamil, Bushra, et al. **"Resource allocation and task scheduling in fog computing and internet of everything environments: A taxonomy, review, and future directions."** *ACM Computing Surveys (CSUR)* 54.11s (2022): 1-38.

[7] Aravind, Kalavagunta, and Praveen Kumar Reddy Maddikunta. **"Dingo Optimization Based Cluster Based Routing in Internet of Things."** *Sensors* 22.20 (2022): 8064.

[8] Pari, Deepanramkumar, and Jaisankar Natarajan. **"Secure Spectrum Access, Routing, and Hybrid Beamforming in an Edge-Enabled mmWave Massive MIMO CRN-Based Internet of Connected Vehicle (IoCV) Environments."** *Sensors* 22.15 (2022): 5647.

[9] Yin, Zhenyu, et al. **"A multi-objective task scheduling strategy for intelligent production line based on cloud-fog computing."** *Sensors* 22.4 (2022): 1555.

[10] Ahmed, Omed Hassan, et al. **"Using differential evolution and Moth–Flame optimization for scientific workflow scheduling in fog computing."** *Applied Soft Computing* 112 (2021): 107744.

[11] Badri, Sahar, et al. **"An Efficient and Secure Model Using Adaptive Optimal Deep Learning for Task Scheduling in Cloud Computing."** *Electronics* 12.6 (2023): 1441.

[12] Harika, Sonti, and B. Chaitanya Krishna. **"Multi-Objective Optimization-Oriented Resource Allocation in the Fog Environment: A New Hybrid Approach."** *International Journal of Information Technology and Web Engineering (IJITWE)* 17.1 (2022): 1-25.

[13] Kumar, M. Santhosh, and Ganesh Reddy Karri. **"Eeoa: cost and energy efficient task scheduling in a cloud-fog framework."** *Sensors* 23.5 (2023): 2445.

[14] Gomathi, B., et al. **"Monarch Butterfly Optimization for Reliable Scheduling in Cloud."** *Computers, Materials & Continua* 69.3 (2021).

[15] Najafizadeh, Abbas, et al. **"Multi-objective Task Scheduling in cloud-fog computing using goal programming approach."** *Cluster Computing* 25.1 (2022): 141-165.

[16] Nguyen, Binh Minh, et al. **"Evolutionary algorithms to optimize task scheduling problem for the IoT based bag-of-tasks application in cloud–fog computing environment."** *Applied Sciences* 9.9 (2019): 1730.

[17] Huang, Xingwang, et al. **"Task scheduling in cloud computing using particle swarm optimization with time varying inertia weight strategies."** *Cluster Computing* 23 (2020): 1137-1147.

[18] Movahedi, Zahra, and Bruno Defude**. "An efficient population-based multi-objective task scheduling approach in fog computing systems."** *Journal of Cloud Computing* 10.1 (2021): 1-31.

[19] Singh, Gyan, and Amit K. Chaturvedi. **"Hybrid modified particle swarm optimization with genetic algorithm (GA) based workflow scheduling in cloud-fog environment for multi-objective optimization."** *Cluster Computing* (2023): 1-18.

[20] Saravanan, T., and S. Saravanakumar. **"Enhancing investigations in data migration and security using sequence cover cat and cover particle swarm optimization in the fog paradigm."** *International Journal of Intelligent Networks* 3 (2022): 204-212.

[21] Jakwa, Ali Garba, et al. **"Performance Evaluation of Hybrid Meta-Heuristics-Based Task Scheduling Algorithm for Energy Efficiency in Fog Computing."** *International Journal of Cloud Applications and Computing (IJCAC)* 13.1 (2023): 1-16.

[22] Liu, Weimin, et al. **"Fog Computing Resource-Scheduling Strategy in IoT Based on Artificial Bee Colony Algorithm."** *Electronics* 12.7 (2023): 1511.

[23] Abd Elaziz, Mohamed, et al. **"Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution."** *Knowledge-Based Systems* 169 (2019): 39-52.

[24] Mangalampalli, Sudheer, Ganesh Reddy Karri, and Ahmed A. Elngar. **"An Efficient Trust-Aware Task Scheduling Algorithm in Cloud Computing Using Firefly Optimization."** *Sensors* 23.3 (2023): 1384.

[25] Awad, A. I., N. A. El-Hefnawy, and H. M. Abdel_kader. **"Enhanced particle swarm optimization for task scheduling in cloud computing environments."** *Procedia Computer Science* 65 (2015): 920-929.

[26] Moon, YoungJu, et al. **"A slave ants based ant colony optimization algorithm for task scheduling in cloud computing environments."** *Human-centric Computing and Information Sciences* 7.1 (2017): 1-10.

[27] Almezeini, Nora, and Alaaeldin Hafez. **"Task scheduling in cloud computing using lion optimization algorithm."** *International Journal of Advanced Computer Science and Applications* 8.11 (2017).

[28] Singhal, Shweta, et al**. "Fault Coverage-Based Test Case Prioritization and Selection Using African Buffalo Optimization."** *Computers, Materials & Continua* 74.3 (2023).

[29] Supreeth, S., et al. **"An efficient policy-based scheduling and allocation of virtual machines in cloud computing environment."** *Journal of Electrical and Computer Engineering* 2022 (2022).

[30] Gollapalli, Mohammed, et al. **"Modeling Algorithms for Task Scheduling in Cloud Computing Using CloudSim."** *Mathematical Modelling of Engineering Problems* 9.5 (2022).