JANOLI International Journal of
Electronics, Computer Science and Engineering
(A Peer Review Journal)

journal homepage: https://janolijournals.com/index.php/jijcse

Volume 2, Issue 1, 2026 ISSN: 3107-4537

Optimization of Task Scheduling Based on Marine Predators' Algorithm for Fog Computing Based Healthcare System

Ganesh D Gourshete , Dr. Susheel Kumar Tiwari
Department of CS&E, Madhyanchal Professional University Ratibad, Bhopal, India

ARTICLE INFO

Keywords:
Edge Computing, IoT, Fog Computing, Optimization, MPA

Correspondence:
Email:
goursheteganesh@gmail.com

ABSTRACT

High-speed development of Internet of Things (IoT) and fog computing has improved the real-time healthcare monitoring system to great extent because it allows processing the data immediately and with low latency as well as utilization of the resources efficiently. Nevertheless, scheduling of tasks in fogs is a serious issue because of the dynamic workloads, non-homogeneous devices, and high Quality of Service (QoS). This paper presents an optimal way of scheduling the tasks in a healthcare system that utilizes fogs by using the Marine Predators Algorithm (MPA). The given approach utilizes exploration and exploitation opportunities of MPA to make task distribution, latency minimization, and even load distribution over fog nodes efficient. Under the three-layer IoT to fog to cloud architecture, the algorithm will be tested on Fog-Simulator in different scenarios with different numbers of sensors. As experimental findings prove, the proposed method is better than current methods, including MPSO, BLA and traditional MPA, in the aspects of the reduction of latency and scalability. The findings also verify that the suggested model enhances the overall performance of the system and promotes the provision of trustworthy real-time healthcare.

Introduction

The development of Internet of Things (IoT) technologies and next-generation computing paradigms has changed the current healthcare systems radically[1-3]. The use of smart healthcare applications is based on constant monitoring, immediate data processing, and timely decision-making to enhance patient outcomes[4,5]. The conventional architecture of cloud computing systems is however associated with high latency, bandwidth constraints and longer response time, which is not desirable in healthcare applications that have a delay time. Mog computing is proposed to handle these challenges; it is an effective paradigm that provides an extension of cloud services at the network edge to provide low latency processing and real-time data analytics[6]. Fog computing is an important aspect of healthcare systems that creates an immediate processing of medical data produced by wearable devices, sensors, or remote monitoring systems. It minimizes the use of centralized cloud infrastructure and offers quick response to important healthcare services like emergency monitoring, diagnosis, and analysis of patient data[7]. Yet, the dynamic and resource-limited characteristics of the environment of fogs pose serious difficulties in the scheduling of tasks and resource allocation. The scheduling should be carried out efficiently, so that the resources of the fog could be used optimally, the execution delay would be minimum, and fewer energy would be spent. The scheduling of tasks on a fog computing is a multifaceted optimization because of the differences in the devices, workloads, and Quality of Service (QoS) requirements[8]. The conventional methods of scheduling do not usually work well in such dynamic environments. Thus metaheuristic optimization algorithms have been given significant attention towards addressing complex scheduling problems. One of them is the Marine Predators Algorithm (MPA)[22], which is based on the foraging behaviour of marine predators and has shown great ability to search the globe and converge well thereby being a good tool to tackle optimization problems. The paper suggests an optimization of the task scheduling model on the Marine Predators Algorithm of the healthcare system supported by the use of the fog computing concept[10,12]. The suggested treatment will be used to optimize the performance of the system by reducing latency, using less energy, and better load balancing among the fog nodes. The proposed approach will help to create effective and responsive smart healthcare systems by effectively allocating the work and reducing the waste of resources. The remainder of the paper is structured in the following way, in Section II there is the literature analysis, Section III outlines the methodology proposed, Section IV summarizes the findings and analysis, and finally, there is the conclusion of the paper in Section V with the future research directions.

II. Related Work

The progress of the next generation computing technologies is very crucial in the advancement of smart healthcare systems. Fog computing also supports real time data collection and offers timely responses with high suitability to health care applications. Task scheduling is a critical factor in the context of the fog computing environment that has a great impact on successful implementation of tasks. Proper optimization of tasks does not only minimize the computation load on fog nodes, it also guarantees that there is efficient distribution of load between computing environments hence leading to improved system performance and reliability. The advent of fog computing has been a successful paradigm to address the shortcomings of the conventional cloud computing, especially the issues of latency, use of bandwidth and energy saving [8], [15]. It is able to compute nearer to end devices, and is therefore very applicable to delay sensitive internet of things applications. Other studies have also been on the optimization of energy use and the execution delay in the IoT-Fog-Cloud setting. As an example, the multi-objective optimization algorithms like the NSGA-II and Bees Algorithms have been demonstrated to achieve better energy usage and response time as compared to the traditional algorithms [1]. Task scheduling in heterogeneous fog environments has also been suggested to be done using hybrid metaheuristic algorithms, but there are still problems with parameter sensitivity, scalability, and local optima [2]. The tasks offloading strategies have been extensively examined to overcome the constraints of mobility and resources with fog computing. Effective offloading strategies take into account the aspects like cooperation of the fog nodes, application distribution, and movement of entities to reduce delays and operational expenses to the minimum level [3]. Also, QoS-conscious optimization methods such as Ant Colony Optimization (ACO), Bee Colony Optimization, and Particle Swarm Optimization (PSO) have shown enhancement of the system performance and energy efficiency [4], [5]. The recent developments focus on hybridization of optimization algorithms to optimize the process of scheduling and allocation of resources. SOS-GOA, Aquila Optimizer, Elephant Herding Optimization (AOEHO), and Hybrid Meta-Heuristic Optimization Algorithm (HMOA) are the algorithms that have been put forward to enhance the execution time, load balancing, and latency. Such methods are superior to traditional methods in different performance metrics but yet have to be established further in real-life dynamic environments. Scheduling methods that are driven by machine learning have also been mentioned to enhance the use of resources and the scheduling of tasks that are sensitive to time delay [8]. In addition, it has also been suggested that cost-based models can be applied to select resource efficiently in cloud-fog systems based on auction mechanism and optimization techniques [10]. A number of survey and review papers have evaluated the available task scheduling algorithms which have also identified some significant difficulties like slow convergence of ACO based, high energy consumption and more latency [5], [12]. Research gaps that are found in these studies include scalability, adaptability as well as effective management of dynamic workloads. This low latency and real-time processing has demonstrated great possibility that fog computing can be used in the real world, like healthcare monitoring and smart cities where real-time process and low latency are essential [16], [19]. Nonetheless, there are other problems like bandwidth limitations, data privacy and system heterogeneity that still remain troublesome [15]. Recent studies also consider the methods of microservices-based IoT applications placement and hyper-heuristic scheduling, as the means to optimize the performance of the systems [21], [22]. These strategies will seek to maximize several goals such as the cost of execution, latency, and energy use

III. Proposed Methodology

The real-time healthcare monitoring system relies heavily on task scheduling and load management as in any case, prompt processing of patient data can have a direct impact on the diagnosis and treatment results. As the list of devices to connect with and data generated by patients grow, it results in more computational and network loads on the system which can greatly affect the latency in the system and reduce the overall performance. This increase in the time of latency impacts the effectiveness of the resource management and can result in delays or failures in the performance of the vital healthcare tasks. It follows that proper scheduling mechanisms are needed that will allow transmitting high-priority medical information in the timeliest manner, without jeopardizing the stability of the system and optimal use of the available resources. In an attempt to overcome these problems, the paper uses the Marine Predators Algorithm (MPA), which is a nature-inspired optimization method that can be used to effectively allocate tasks. The foraging behaviors of marine predators are modeled by MPA to search and exploit the search space, which allows making more efficient decisions in the allocation of tasks to the available resource spectrum. The algorithm is able to improve task scheduling performance, latency, and resource utilization by dynamically adjusting to changes in workload. According to the proposed framework as depicted in Fig. (1), the entire work-flow of the task scheduling with the help of MPA is presented where the tasks are prioritized, allocated and reallocated to ensure efficiency in the real-time healthcare settings.

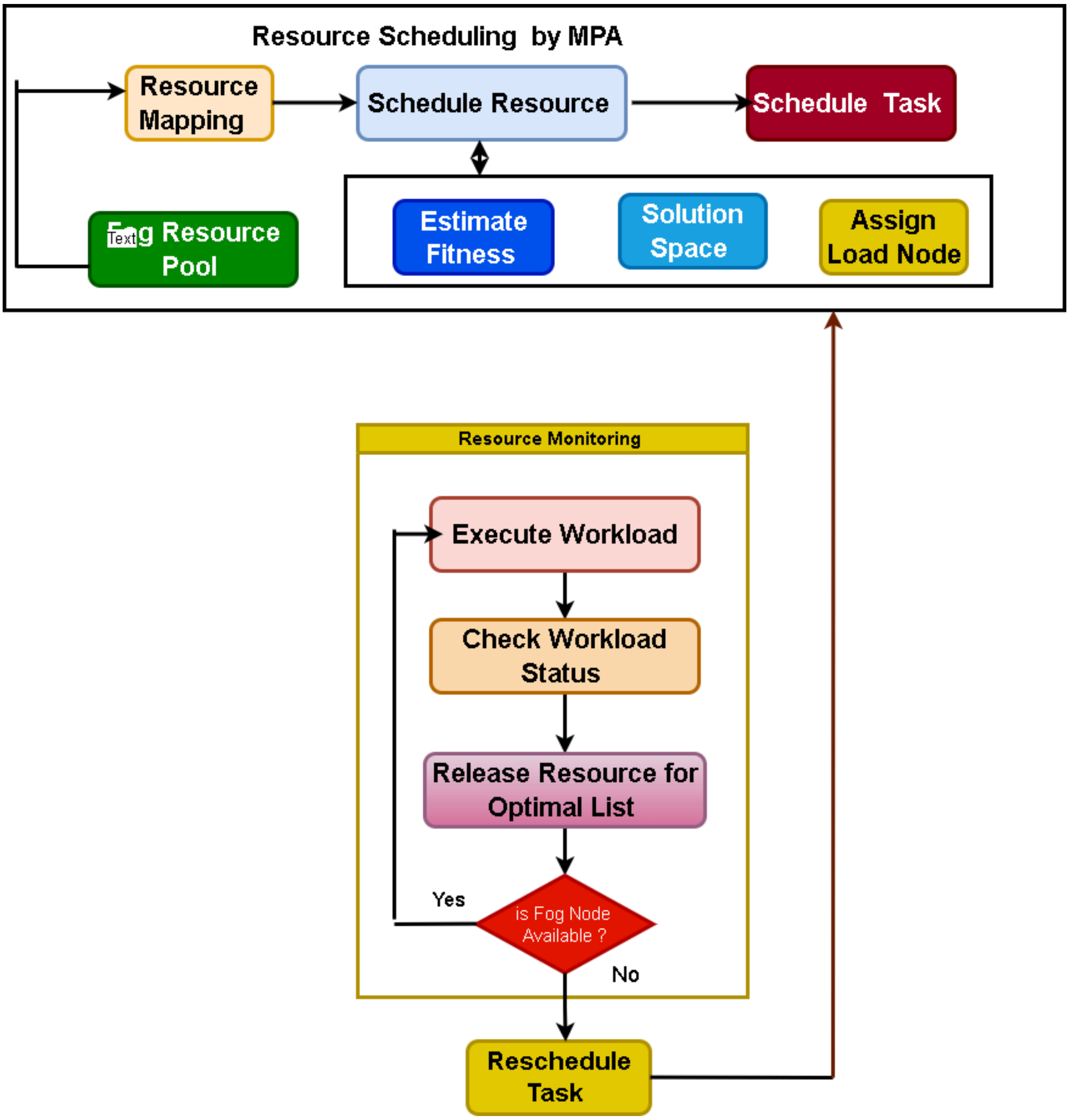

Fig. 1 Proposed Model of Task Scheduling in Fog Computing Environment

Marine Predators Algorithm

The MPA, proposed in 2020, is based on a meta-heuristic optimization approach inspired by the foraging behaviour of swarm intelligence. The algorithm models adaptive movements that allow predators to efficiently search for prey in large and uncertain environments. By virtue of this behavior, MPA exhibits sophisticated movement patterns that allow it to adapt to diverse and dynamic search spaces, similar to those found in large marine ecosystems, where fluctuating conditions result in significant variability and uncertainty [23,24,25].

MPA has attracted considerable attention and wide applicability in various scientific and engineering fields because of its advantages, including simple implementation, low parameter tuning, and strong robustness in convergence toward the optimal solution. In this algorithm predators and prey are represented as agents engaged in the search for resources. The behaviour of these agents is simulated analogously to natural processes, governed by the principle of "survival of the fittest," which increases the probability of successful prey detection and improves the probability of finding optimal solutions in large and complex search spaces.

The position of each solution in MPA is updated based on its state and probabilistic rules that dictate transitions to new locations. The algorithm combines two complementary mechanisms - Lévy flight and Brownian motion—that operate in tandem, enabling both global exploration and local exploitation. This balance allows the algorithm to efficiently search the solution space and converge toward near-optimal solutions. The mathematical formulation of MPA, which details the position update and search behavior, is presented in Equation (1) [24].

$$X_0 = X_{min} + rand\, X\left(X_{max} - X_{min}\right) \quad \dots \quad (1)$$

When X_{max} and X_{min} stands for the maximum and minimum limits of the heuristic space, the interval of the random function is [0,1]. The optimization process in MPA is based on the behavior of each group of predators and prey through three separate phases, summarized as follows.

Phase 1: During Phase 1, the predator has a substantially higher velocity than the prey. The objective of this phase is to describe the heuristic space and to get closer to the prey (i.e., ideal solutions). Equation (2) demonstrates the mathematical model of this phase.

$$While\, i < \frac{1}{3} T_{max}\, i = 1,\ 2, \dots, n \quad \dots \quad (2)$$

$$\{stepsizei = (R_B \otimes (Elitei - R_B \otimes\ preyi))\, preyi = preyi + PXR \otimes Stepsizei$$
(3)

The symbol i indicates the number of iterations during the running process, and the maximum number of iterations of the algorithm is T_{max}. The size of the steps represents the acceleration of the movement. The random vector R_B follows a normal distribution. The predator and prey matrices are represented by $Elitei$, and the symbol ⊗ denotes element-wise multiplication. The numerical value of P is 0.5.

Phase2: Explore the distribution of the population as prey and predators that the formula represents as

$$\frac{1}{3} T_{max} < i < \frac{2}{3} T_{max},\ \ i = 1,\ 2, \dots, \frac{n}{2} \quad \dots \quad (4)$$

$$\{stepsizei = \left(R_L \otimes \left(Elitei - R_L \otimes preyi\right)\right) preyi = preyi + PXR \otimes stepsizei$$
… (5)

$$While\ \frac{1}{3} T_{max} < i < \frac{2}{3} T_{max},\ \ i = \frac{n}{2}, \dots\dots, n \quad \dots \quad (6)$$

$$\{stepsizei = (R_B \otimes (R_B \otimes Elitei - preyi\ ))\, preyi = Elitei + PXCF \otimes stepsizei$$
… (7)

Now, in the R_L motor vector, the moving parameters are controlled by CF and present as

$$CF = \left(1 - \frac{i}{T_{max}}\right) \frac{2i}{T_{max}} \quad \dots \quad (8)$$

Phase 3: the last phase of the algorithm mainly explores the exploitation phase. The predator mainly approaches the prey through the Levy motion. The mathematical formula is shown as

$$While\, i > \frac{2}{3} T_{max,}\ i = 1, 2, \dots, n \quad \dots \quad (9)$$

$$\{stepsizei = (R_L \otimes (R_L \otimes Elitei - preyi))\, preyi = Elitei + PXCF \otimes stepsizei$$
(10)

In the above phase MPA algorithm also include eddy currents and the influence of Fish-Aggregating Devices (FADs), the formula of influence is as follows:

$$preyi = \{preyi + CF\left[X_{min} + R\otimes\left(X_{max} + X_{min}\right)\right]\otimes U \quad if\ r\leq Pf \quad preyi + [Pf(1 - r) + r]X\left(prey_{r1} - prey_{r2}\right) \quad if\ r \quad ... \quad (11)$$

The value of *Pf* represents the probability that FADs influence the optimization process, and *U* is a binary array with values of 0. The variable *r* belongs to the range [0,1], while *r1* and *r2* are random indices that represent positions within the prey matrix.

Algorithm for Task Scheduling

```
Begin
  Set up population of solutions Xi.
  Evaluate fitness of each Xi
  Identify Elite solution
  For Iter = 1 to MaxIter do
    For each search agent Xi do
      If Iter < MaxIter/3 then
        Update on Brownian motion.
      Else if Iter < 2*MaxIter/3 then
        Update using Brownian + Lévy
      Else
        Elite: Update with Lévy flight.
      End If
            Apply boundary constraints
      Evaluate new fitness
    End For
    Update Elite (best solution)
    Use the memory saving (greedy selection).
    Randomization (eddy effect) Introduce randomization.
  End For
  Optimal schedule Return Elite.
End
```

IV. Experimental Analysis

The open-source application Fog-Simulator, which is developed to simulate the fog and cloud-based IoT environment, is used to test the offered task scheduling algorithm of fog computing. The main objective of the simulation is to reduce the latency as opposed to the current methods of managing tasks. The efficiency of the algorithm is evaluated in four cases on a three-layer IoT-based test system. The large amount of IoT sensors in the first layer is used to gather information about the weather, traffic, and surveillance videos. These sensors are randomly positioned with a distance of 100 to 300 meters of the fog nodes with the number of these sensors varying between 200 and 2000. The sensors can produce 250Kb to 1024 Kb of data and delay of up to 100 milliseconds is acceptable depending on the application demands. The second layer is comprised of 8-20 fog nodes that are to process and aggregate the data that is received by the sensors. The node of the m fogs contains several virtual machines, and each VM is dedicated to the processing of certain types of IoT applications. The third layer is a representation of the cloud that is the central unit of long-term data processing and storing. This layer has an instance of a solitary data center linked to the fog layer through the Internet. The general multi-layer design is also meant to optimise the task scheduling, latency and system performance in the fog computing environments. Table 1 gives detailed simulation parameters.

Table 1 Simulation Parameters of Fog-Simulator Configurations [12,13]

Parameters	**Value**
$Sensor\ nodes$	$250 - 2500$
$Fog\ Nodes$	$10 - 25$
Xp	$250 - 1024\ KB$
$ʎi$	$5 - 50$
$ʎf$	$10 - 50$
$ʎc$	$25 - 75$
μf	$50 - 300$
μc	$100 - 400$
$Nants$	40
$Maxiter$	125
P	0.1
A	0.5
ϒ	5.0
Bf	$50\ MB/S$
Bc	$100\ MB/S$

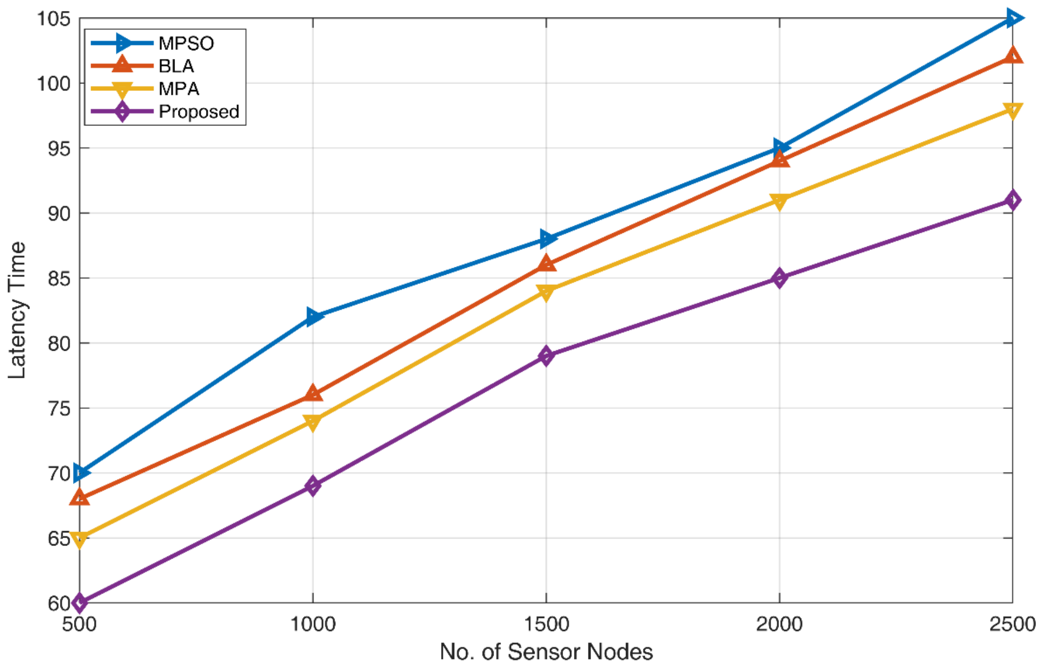

Fig.2 Performance Analysis of Scenario-1 of latency Time

Fig. (2) has provided the performance comparison of various task scheduling algorithms i.e. MPSO, BLA, MPA, and the Proposed method in the varying number of IoT sensors. The performance metric (which is presumably the latency or execution time) of all algorithms rises with the number of sensors (500 to 2500), representing an increase in system load and complexity. High values are consistently seen in MPSO (e.g. 70 at 500 sensors and 105 at 2500 sensors), which shows relatively poor performance with increasing load. The nature-inspired optimization techniques are effective as BLA outperforms MPSO whereas MPA outperforms it further. The Proposed method, however, is best in all the situations with the lowest values of (60 at 500 sensors and 91 at 2500 sensors) which is more efficient in dealing with the task scheduling and load balancing despite the system scale. The findings clearly show that the Proposed algorithm has better scalability and lower latency than those of the existing methods. As the number of sensors grows, the performance difference between the two algorithms also gets larger, indicating that it can withstand high-load applications common to the internet of things in healthcare

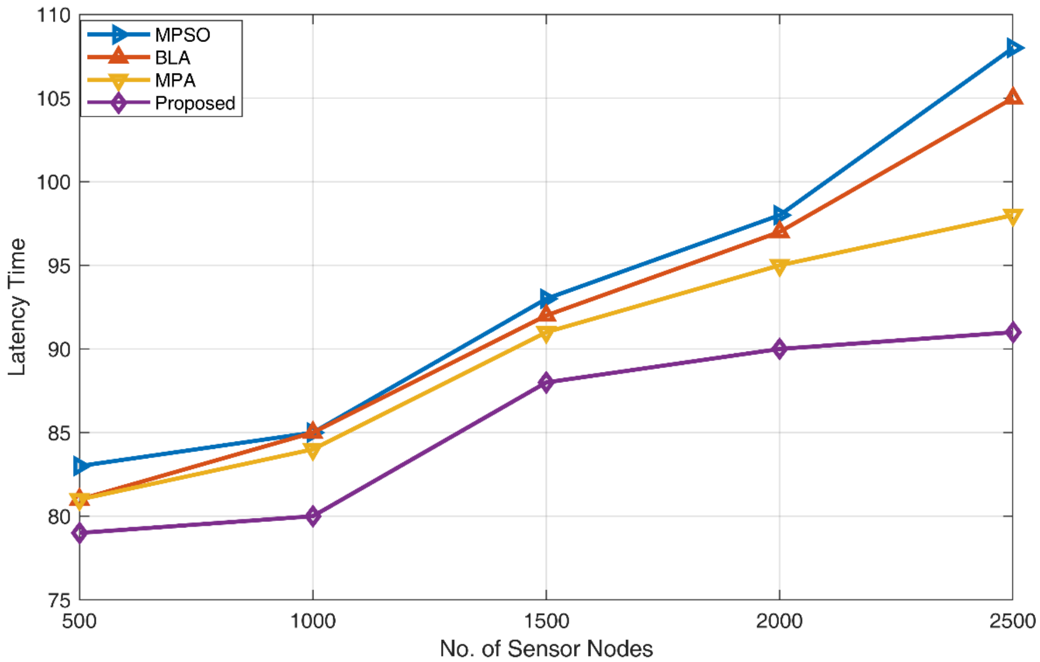

Fig.3 Performance Analysis of Scenario-2 of latency Time

Fig. (3) carries out the comparative analysis of the performance of MPSO and BLA, MPA, and the Proposed algorithm across different measurements of the number of IoT sensors. As can be seen, the performance measure (latency or execution time) of all algorithms increases as the sensor count (number of sensors) increases between 500 and 2500, which reflects the effect of workload increase on the system. Out of the available methods, MPSO has the highest values and it goes up to 108, indicating that it is not as efficient as other methods in managing a large environment. BLA and MPA have better performance as compared to MPSO, although in the majority of the cases, MPA outperforms BLA because of more effective optimization. Nevertheless, the Proposed method is always the lowest in all the situations with the range of 79-91, which is a better tool in scheduling the tasks and resources. In addition, the Proposed algorithm has enhanced scalability with increased size of the system and its performance loss rate remains relatively slower. The performance gap is more evident at a larger number of sensors, especially after 1500 sensors, at which stage the Proposed method is much better than the other algorithms. This implies that it is strong in handling large volumes of data and real time processing needs that would be endangered in the healthcare IoT systems. To improve this even further, it will be suggested to implement improvements in the Proposed approach by adding adaptive and predictive capabilities based on machine learning where prioritization of tasks in a dynamic way and assigning resources intelligently can be done.

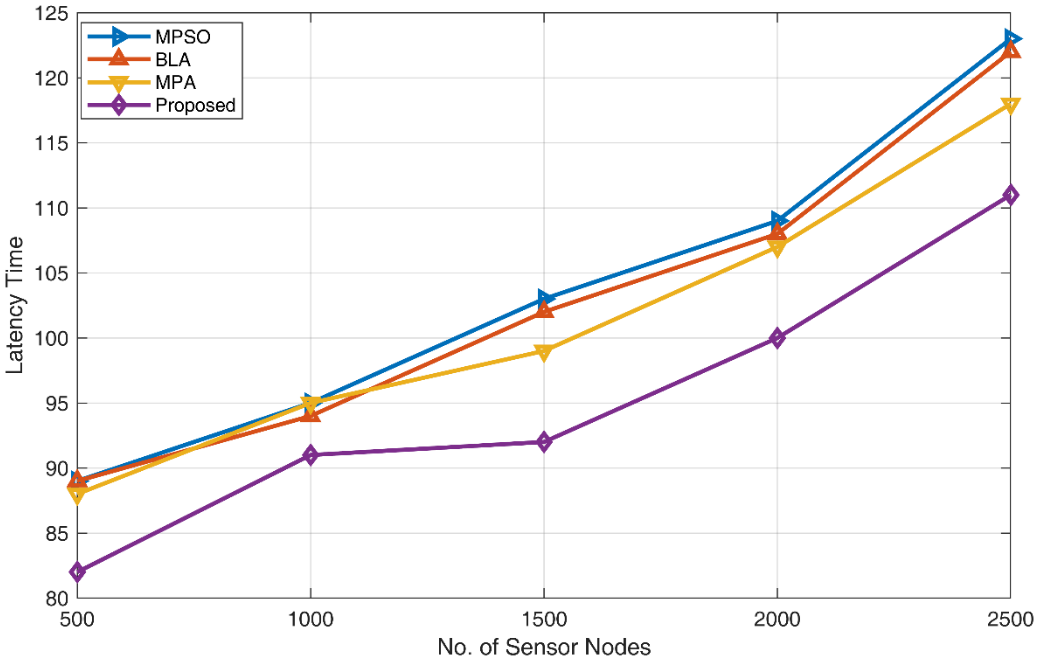

Fig.4 Performance Analysis of Scenario-3 of latency Time

The performance of MPSO, BLA, MPA, and the Proposed algorithm are compared in terms of performance at different numbers of the IoT sensors as shown in fig. (4). It is noted that with the increment of the number of sensors, both 500 and 2500, the performance measure (latency or execution time) rises with all algorithms, and it indicates the increase in the computational load and the network load. Of the available methods, MPSO and BLA reported the highest values of 123 and 122 respectively at 2500 sensors suggesting low efficiency due to heavy workloads. In comparison to MPA, the latter exhibits a better performance, and the values have risen between 88 and 118 due to the higher optimization ability. Nonetheless, the Proposed method always scores the lowest values with 82 to 111 which in turn signifies reduced time spent in the task scheduling and resource usage. More so, the Proposed algorithm exhibits improved scalability and stability with an increase in the number of sensors, with the performance degradation increasing comparatively with other algorithms. The difference in performance is increased with a sensor density of higher values and especially above 1500 sensors where efficiency on load balancing and low latency matters greatly. It underscores the strength of the Proposed approach to manage real-time, large-scale IoT health care settings.

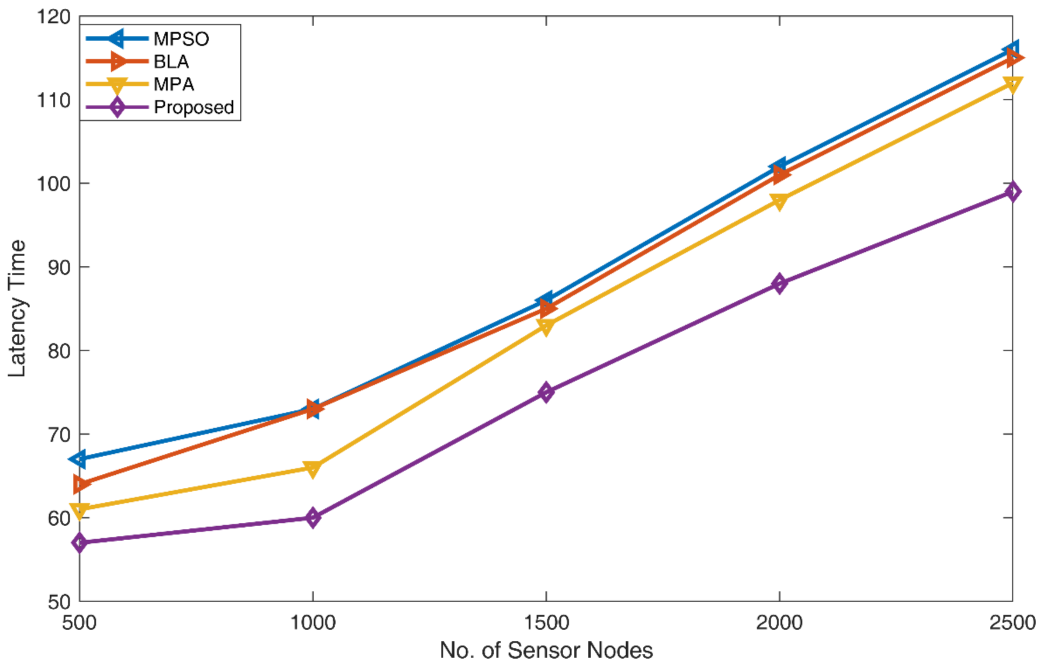

Fig.5 Performance Analysis of Scenario-4 of latency Time

The performance comparison of MPSO, BLA, MPA, and the Proposed algorithm under different numbers of IoT sensors is presented in fig. (5). It is evident that as the number of sensors increases from 500 to 2500, the performance metric (e.g., latency or execution time) increases for all algorithms due to the growing system load and computational complexity. Among the existing methods, MPSO exhibits the highest values, increasing from 67 to 116, indicating relatively lower efficiency. BLA performs slightly better than MPSO, while MPA demonstrates improved performance with values ranging from 61 to 112, highlighting the effectiveness of metaheuristic optimization. However, the Proposed method consistently achieves the lowest values across all scenarios, ranging from 57 to 99, indicating superior task scheduling and load balancing capability. Furthermore, the Proposed algorithm shows better scalability and robustness as the number of sensors increases, with a comparatively slower rise in performance degradation. The performance improvement becomes more prominent at higher sensor counts, particularly beyond 1500 sensors, where efficient resource allocation is crucial. This demonstrates the ability of the Proposed approach to effectively handle large-scale and real-time IoT healthcare environments.

V. Conclusion & Future Work

This article demonstrated an effective trade scheduling model of a healthcare system based on the Marine Predators Algorithm on the basis of the fog computing. The suggested solution is useful in overcoming the main difficulties linked to long latency, resource limitations, and the ability to handle dynamic workload in the IoT-enabled healthcare setting. Through adaptive search procedures of MPA, the model is able to achieve better allocation of tasks, lesser delay of the execution process, and better load balancing among the nodes in the state of fog. The experimental findings using Fog-Simulator in various scenarios reveal that the proposed algorithm is always better than the other currently existing ones, including MPSO, BLA, and the common MPA, especially when the sensor density in the environment grows and the environment is large. Although the performance was promising, there are still a number of directions that can be taken in future research. The model suggested can be generalized to include machine learning methods of predictive and adaptive planning of tasks in dynamic environments. Moreover, the system can be further optimized with energy-conscious methods to improve system efficiency and sustainability. The future work can be also dedicated to real-life application in heterogeneous settings of the IoT-fog, and mobility, network variability, and security constraints should be taken into account. In addition, hybrid optimization methods involving the use of MPA in conjunction with other metaheuristic or deep learning methods can be considered to enhance the performance of more sophisticated healthcare systems in terms of monitoring.

References

[1]. Jafari, Vahid, and Mohammad Hossein Rezvani. "Joint optimization of energy consumption and time delay in IoT-fog-cloud computing environments using NSGA-II metaheuristic algorithm." Journal of Ambient Intelligence and Humanized Computing 14, no. 3 (2023): 1675-1698.

[2]. Kumar, Medishetti Santhosh, and Ganesh Reddy Karri. "AGWO: Cost Aware Task Scheduling in Cloud Fog Environment Using Hybrid Metaheuristic Algorithm." Int. J. Exp. Res. Rev 33 (2023): 41-56.

[3]. Mostafavi, S. A., and Elham Barkhordari. "A Mobility Aware Task Offloading Scheme Based On Ant Colony Optimization Algorithm In Software-Defined Fog Computing." TABRIZ JOURNAL OF ELECTRICAL ENGINEERING 53, no. 4 (2023): 245-256.

[4]. Reyana, A., Sandeep Kautish, Khalid Abdulaziz Alnowibet, Hossam M. Zawbaa, and Ali Wagdy Mohamed. "Opportunities of IoT in Fog Computing for High Fault Tolerance and Sustainable Energy Optimization." Sustainability 15, no. 11 (2023): 8702.

[5]. Low, Choon Keat, and Gar Chi Phoon. "Study of Algorithms for Scheduling of IoT Application Tasks in Fog Computing."

[6]. Mohammadzadeh, Ali, Mahdi Akbari Zarkesh, Pouria Haji Shahmohamd, Javid Akhavan, and Amit Chhabra. "Energy-aware workflow scheduling in fog computing using a hybrid chaotic algorithm." The Journal of Supercomputing (2023): 1-36.

[7]. Mohamed, Ahmed awad, Laith Abualigah, Alhanouf Alburaikan, and Hamiden Abd El-Wahed Khalifa. "AOEHO: a new hybrid data replication method in fog computing for IoT application." Sensors 23, no. 4 (2023): 2189.

[8]. Sharma, Oshin, Geetanjali Rathee, Chaker Abdelaziz Kerrache, and Jorge Herrera-Tapia. "Two-Stage Optimal Task Scheduling for Smart Home Environment Using Fog Computing Infrastructures." Applied Sciences 13, no. 5 (2023): 2939.

[9]. Sharma, Pankaj, and P. K. Gupta. "Optimization of IoT-Fog Network Path and fault Tolerance in Fog Computing based Environment." Procedia Computer Science 218 (2023): 2494-2503.

[10]. Bindu, GB Hima, Kasarapu Ramani, G. Reddy Hemantha, N. Pushpalatha, I. Suneetha, P. Harish, and A. B. Manju. "An Optimized Resource Allocation Model for Cloud Computing Using Ant Colony-based Auction Method." International Journal of Intelligent Systems and Applications in Engineering 11, no. 3 (2023): 818-824.

[11]. Khan, Salman, Ibrar Ali Shah, Muhammad Faisal Nadeem, Sadaqat Jan, Taegkeun Whangbo, and Shabir Ahmad. "Optimal Resource Allocation and Task Scheduling in Fog Computing for Internet of Medical Things Applications." (2023).

[12]. Misirli, Javid, and Emiliano Casalicchio. "An Analysis of Methods and Metrics for Task Scheduling in Fog Computing." Future Internet 16, no. 1 (2023): 16.

[13]. Bhargavi, K., B. Sathish Babu, and Sajjan G. Shiva. "Type-2-Soft-Set Based Uncertainty Aware Task Offloading Framework for Fog Computing Using Apprenticeship Learning." Cybernetics and Information Technologies 23, no. 1 (2023): 38-58.

[14]. Zambuk, Fatima Umar. "Latency-Aware Task Scheduling Based on Hybrid Meta-Heuristic Optimization Algorithm in Fog Computing."

[15]. Vaigandla, K. K., and M. Siluveru. "Fog Computing with Internet of Things: An Overview of Architecture, Algorithms, Challenges and Applications." Journal of Engineering and Technology (JET) 14, no. 1 (2023): 187-220.

[16]. Songhorabadi, Maryam, Morteza Rahimi, AmirMehdi MoghadamFarid, and Mostafa Haghi Kashani. "Fog computing approaches in IoT-enabled smart cities." Journal of Network and Computer Applications 211 (2023): 103557.

[17]. Ortiz-Garcés, Iván, Roberto O. Andrade, Santiago Sanchez-Viteri, and William Villegas-Ch. "Prototype of an Emergency Response System Using IoT in a Fog Computing Environment." Computers 12, no. 4 (2023): 81.

[18]. Fahad, Muhammad, Mohammad Shojafar, Mubashir Abbas, Israr Ahmed, and Humaira Ijaz. "A multi-queue priority-based task scheduling algorithm in fog computing environment." Concurrency and Computation: Practice and Experience 34, no. 28 (2022): e7376.

[19]. Gupta, Arti, and Vijay Kumar Chaurasiya. "Efficient Task-Offloading in IoT-Fog Based Health Monitoring System." In 2022 OITS International Conference on Information Technology (OCIT), pp. 495-500. IEEE, 2022.

[20]. Bhardwaj, Amit Kumar. "Machine Learning based Power Efficient Optimized Communication Ensemble Model with Intelligent Fog Computing for WSNs." (2022).

[21]. Pallewatta, Samodha, Vassilis Kostakos, and Rajkumar Buyya. "Placement of Microservices-based IoT Applications in Fog Computing: A Taxonomy and Future Directions." ACM Computing Surveys (2023).

[22]. Rahbari, Dadmehr. "Analyzing Meta-Heuristic Algorithms for Task Scheduling in a Fog-Based IoT Application." Algorithms 15, no. 11 (2022): 397.

[23]. Abdel-Basset, Mohamed, Reda Mohamed, Mohamed Elhoseny, Ali Kashif Bashir, Alireza Jolfaei, and Neeraj Kumar. "Energy-aware marine predators algorithm for task scheduling in IoT-based fog computing applications." *IEEE Transactions on Industrial Informatics* 17, no. 7 (2020): 5068-5076.
[24]. Mahfouz, Khaled Houssam, Mohammed Azmi Al-Betar, Sharif Naser Makhadmeh, and Qusai Yousef Shambour. "Mitigating the task scheduling Problem in fog computing environments using marine predators optimization algorithm." *Cluster Computing* 28, no. 15 (2025): 973.
[25]. Singh, Simar Preet. "Effective load balancing strategy using fuzzy golden eagle optimization in fog computing environment." *Sustainable Computing: Informatics and Systems* 35 (2022): 100766.