DOI: https://doi.org/10.1109/tce.2025.3526992

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Version: Accepted Version

Enhancing Quality of Service in IoT-WSN through Edge-Enabled Multi-Objective Optimization

Shailendra Pratap Singh, Naween Kumar, Gyanendra Kumar, Balamurugan Balusamy, Ali Kashif Bashir, Maryam M. Al Dabel

***Abstract*—The demand for real-time, high-quality services (QoS) is increasing with the proliferation of the resource-constrained nature of edge devices that facilitate the Internet of Things (IoT) and wireless sensor network (WSN) applications. Several existing multi-objective algorithms, such as MOPSO, Elitism MOGA, MODE, and others, are capable of balancing exploration and exploitation; they assist in efficient QoS management for WSN-IoT applications, address resource limitations, and align with the objectives of the applications. However, they suffer from showing robustness in the solution and efficient convergence rates on benchmark functions impacting overall QoS. This paper proposes a multi-objective optimization and edge-intelligent adaptation-based strategy to address QoS management issues, jointly optimize several competing objectives, like energy and latency, and maximize localization and coverage rates while considering the limitations of edge devices. The proposed work uses a novel Grey-wolf optimizer (GWO) Algorithm with an innovative bird-edge-computing adaptation approach to analyze the complex connections between input parameters, edge resources, and QoS indicators to generate Pareto-optimal solutions. The evaluation of the proposed edge intelligence technique with IoT applications demonstrates its effectiveness compared to conventional heuristic-based approaches. This approach enhances the QoS in IoT applications and improves resource utilization and scalability in edge computing environments.**

***Index Terms*—Grey Wolf Optimization, Evolutionary Algorithms, Edge Computing, Internet of Things, Quality of Service.**

I. INTRODUCTION

Effective resource allocation and decision-making in WSN-IoT applications are challenging due to the rapid evolution of heterogeneous sensors, consumer/industry /edge devices, varying QoS requirements, and the growing demand for high-quality services [1]–[3]. To address these challenges, edge computing has emerged as a promising methodology, leveraging processing capabilities at the network edges [4], [5].

Shailendra Pratap Singh is with Department of Computer Science and Engineering, MMMUT Gorakhpur UP, India (e-mail: shail2007singh@gmail.com)
Naween Kumar is with SCSET, Bennett University, Greater Noida, India (e-mail: kr.naween@gmail.com)
Gyanendra Kumar is with Department of IoT and Intelligent Systems, Manipal University Jaipur, Jaipur, India (e-mail: maurya.gyanendra@gmail.com and gyanendra.kumar@jaipur.manipal.edu)
Balamurugan Balusamy is with Shiv Nadar Institution of Eminence , Delhi-National Capital Region (NCR), India (e-mail: balamurugan.balusamy@snu.edu.in)
Ali Kashif Bashir is with Networks and Security, Manchester Metropolitan University, UK(e-mail: dr.alikashif.b@ieee.org)
Maryam M. Al Dabel is with Department of Computer Science and Engineering, College of Computer Science and Engineering, University of Hafr Al Batin, Saudi Arabia(e-mail: maldabel@uhb.edu.sa)
Corresponding Author: Gyanendra Kumar

Recent attention has turned to multi-objective optimization algorithms as robust solutions for handling complex optimization issues with competing objectives in IoT/WSN/IoT-WSN [6]–[8]. These algorithms determine solutions that demonstrate trade-offs between goals, offering a way to achieve QoS objectives, say increasing throughput, increasing energy savings, maximizing packet success-delivery ratio, and minimizing data gathering delay for network efficiency, dependability, and others [9]. By increasing throughput, data collection efficiency is maximized, resource usage is optimized, and user experience can be seamless for high-bandwidth networks. The network lifespan is extended by saving energy expenditure. Efficient data communication is also achieved by enhancing the packet success-delivery ratio and reducing delays. To fully exploit the network's potential, it is necessary to balance these objectives. The dynamic IoT ecosystem requires adaptive optimization strategies to meet changing user demands and network dynamics.

When faced with the complexity of various conflict challenges involving the multi-objective optimization of diverse network parameters, existing multi-objective optimization algorithms may encounter limitations. Although there have been attempts to balance such parameters [10], these methods often fail to achieve a significant balance between exploitation and exploration. Notably, existing multi-objective optimization techniques such as multi-objective-based Integer linear programming [11], MOPSO [7], [8], elitism-based MOGA [12], MOWOA [13], NSGA-III (derived from [14]), MOFOX (derived from [15]), and MOEA-D [16] succeeded in striking a balance between exploration and exploitation, but lack the robustness required for managing complex and multi-modal optimization problems, similar to those posed in our study. These strategies could significantly contribute to efficient QoS management in IoT, effectively addressing resource limitations and aligning with the diverse objectives shared with WSNs; however, they suffered from convergence rates on benchmark functions to ensure overall QoS.

So, there is a pressing need to optimize multiple network parameters efficiently and strike a balance between exploration and exploitation to identify diverse trade-offs in an intelligent way. This improvement is crucial, mainly when focusing on various conflict objectives. Furthermore, there is a challenge in designing an algorithm that can efficiently search for Pareto optimal solutions in high-dimensional solution spaces, considering the conflicting objectives and resource constraints. Furthermore, incorporating edge intelligence requires developing models to capture complex relationships between input parameters, edge resources, and QoS metrics, enabling adaptive and data-driven decision-making. The key contributions of the proposed article are as follows:

- To propose a new method that integrates GWO with multi-objective optimization, drawing inspiration from grey wolves' hunting behavior to explore and exploit high-dimensional spaces efficiently.
- To enhance diversity and accelerate convergence in multi-objective benchmark functions, the approach employs a bird-hunting strategy along with GWO. Its enhanced social behavior in a multi-objective scenario facilitates improved diversity, exploration, adaptability, and effective handling of Pareto dominance.
- To optimize the balance between energy efficiency and network connectivity in IoT systems, the proposed method integrates edge computing seamlessly with GWO.
- To validate the practicality and effectiveness of the proposed approach, two comprehensive tests are conducted: one application based on smart IoT and another focusing on QoS optimization. The performance is compared with the state-of-the-art approaches NSGA-III [14], MOPSO [8], MOWOA [13], and MOEA-D (MODE) [16] algorithms and results show that proposed method performed better in achieving QoS.

The article is structured to provide a comprehensive exploration of the topic. Section II reviews related work, Section III explores IoT's service model, Section IV presents the proposed multi-objective technique, and Section V evaluates its performance. Finally, Section VI concludes the study and outlines future research directions.

II. Literature Review

In this section, we briefly survey a few existing multi-objective optimisation methods relevant to this work.

Zhang et al. [6] proposed a Penalty-based evolutionary algorithm to solve a QoS-based multi-objective problem, however the solution has higher time complexity. Chaudhry et al. [7] developed a MOPSO algorithm with better running-time complexity to address multicast routing, involving performance metrics like energy expenditure, delay, and data loss. Salimian et al. [8] optimize IoT service placement issues by developing an improved MOPSO algorithm. This work aimed to improve fog resource utilization and the QoS in IoT. However, MOPSO has less ability to effectively balance exploration and exploitation than MOGWO during optimization process.

Natesha et al. [12] presented a two-level resource provisioning fog framework using a containerization approach and formulated the service placement problem in a fog computing environment as a multi-objective optimization problem for minimizing the cost, energy depletion, and service time and thus ensuring the QoS of IoT applications by developing elitism-based MOGA algorithm. But, in their algorithm, optimizing the interplay among diversity, speed, and convergence to achieve an efficient solution was difficult. Huang et al. [13] presented MOWOA algorithm based on delay-time and energy expenditure to address the optimal computation offloading mechanism in the context of mobile-edge-computing ecosystems, however it was unable to provide better degree of diversity of the solution set. Shailendra et al. [16] developed a multi-objective differential evolution (MOE-D) algorithm encompassing a rapid mutation operator. The method enhances diversity and convergence rate, with evaluations in IoT ecosystems reporting performance efficacy in optimizing service cost, delay, and network lifetime, however it is a bit slower algorithm. Shrestha et al. [17] proved the dynamism of high-bandwidth networks to enhance the Industrial IoT devices' coverage, but did not consider delay metric. Bairagi et al. [11] addressed the simultaneous optimization of energy consumption minimization and coverage maximization by developing a multi-objective integer-linear-programming method. However, their solution suffers from intractability issues for large-sized networking ecosystems. This is because of the problem's non-deterministic polynomial time-hard complexity.

Wei et al. [18] tried to optimize resource allocation in the vehicular cloud computing ecosystem by exploiting a modified form of NSGA-II using matching factors, dynamic crossover, and mutation probability factors per the provider's and users' viewpoints. Their algorithm showed increased population diversity; however, runtime execution might have been more prolonged. Li et al. [19] exploited the Deep Reinforcement Learning mechanism to address task offloading issues for connected vehicles in UAV-Aided mobile edge computing (MEC) networks. However, the proposed algorithm is complex. Wang et al. [20] devised a microservice-oriented Service Placement technique for MEC-enabled Internet of Vehicle networks to reduce service latency, minimize excessive resource expenditure and ensure perpetual sustainability. This contributed work is for mobile internet vehicles.

Qi et al. [21] devised joint beamforming methods based on multi-objective scenarios for maximizing the network-performance at the cost of minimizing total transmit power. Sheena et al. [22] proposed an energy-efficient seagull-based multi-objective-optimization algorithm to cluster and balance the traffic load for Disastrous Management scenarios by minimizing overhead while achieving superior convergence. Khosroabadi et al. [23] presented the SCATTER scheme for delay-sensitive network applications operating in integrated fog-cloud environments. They tried to solve the service placement problem by prioritizing improving QoS.

Despite extensive research on WSN-IoT networks, there is a need for deeper exploration of conflicting goals, such as data throughput and energy-saving improvement, requiring further refinement of multi-objective optimization techniques in dynamic network environments. Meeting these gaps is crucial for advancing the field and deploying efficient WSN-IoT ecosystems.

The limitations mentioned above emphasize the need for a comprehensive, efficient, multi-objective solution with the following features: (1) exploiting a multi-objective Algorithm to balance multiple IoT parameters simultaneously, (2) an efficient computational algorithm that works effectively in multi-dimensional space, (3) combining various network attributes to ensure reliable data communication and responsiveness, (4) exploiting a hybrid multi-objective variant of grey wolf optimization where the algorithm is having exceptional mathematical ability to balance exploration and exploitation, controls

TABLE I. Comparative analysis of relevant related works

Authors	**Multiple Network Parameters**			
	Energy	**Delay**	**Localization rate**	**Coverage rate**
Chaudhry (2019)	✓	✓	×	×
Huang (2021)	✓	✓	×	×
Shailendra (2022)	✓	✓	×	×
Sheena (2023)	×	✓	×	×
Shrestha (2023)	✓	×	×	✓
Zaborski (2022)	✓	✓	×	×
Natesha (2021)	✓	×	×	×
Khosroabadi (2021)	✓	×	×	×
Salimian (2022)	×	✓	×	×
Bairagi (2022)	✓	✓	×	✓
Our Scheme	✓	✓	✓	✓

population optimization and adaptation and robustness are controlled by Bird-edge-computing adaption strategy for finding a real-time optimal solution to enhance the network performance in adaptable WSN-IoT ecosystem. Table I compares existing and proposed work across four essential parameters: energy, delay, throughput, and packet loss ratio. In this Table I, a cross mark ($\times$) indicates the parameter's absence, and a tick mark ($\checkmark$) signifies its presence in a specific research study.

A. Problem Statement

In the context of a WSN-IoT ecosystem, the goal is to optimise the QoS for a given set $\mathbb{SN}$ of WSN-IoT devices $(S_1, S_2, \ldots, S_N)$ along with their respective data-availability set $\mathbb{DAA} = \{Initial_{Data}(S_1), Initial_{Data}(S_2), \ldots, Initial_{Data}(S_n)\}$ and initial remaining energy distribution set $\mathbb{RE} = \{RE(S_1), RE(S_2), \ldots, RE(S_n)\}$ across a random 2-D plane. The objective is to jointly optimise various conflicting objectives expressed as: (i) *Localization Rate Improvement*: Enhancing localization rate is essential for accurately determining the spatial position of IoT devices. (ii) *Energy Efficiency Improvement*: The optimization framework aims to enhance energy efficiency, maximizing the utilization of available energy resources. (iii) *Coverage Rate Optimization*: Optimizing coverage rate is crucial to ensure adequate sensor coverage across the IoT ecosystem. (iV) *Low Latency Achievement*: Achieving low latency is a key objective, ensuring swift response times in the dynamic environment.

III. Application of IoT-based Wireless Sensors Network

An IoT-based WSN amalgamates two integral technological components: wireless sensor nodes and the IoT. These nodes gather diverse data in varied settings, leveraging IoT for sophisticated data analysis and smarter decision-making. The versatility and adaptability of IoT-based WSNs make them a transformative technology with the potential to revolutionize a wide range of industries and improve the quality of life for individuals and communities. Figure 1 depicts the layered architecture of IoT-based WSNs [24]. The Figure shown encapsulates a multi-layered framework for IoT-based WSNs, encompassing Gateway connectivity, communication network orchestration, application insights, security, scalability, power management, and standardization.

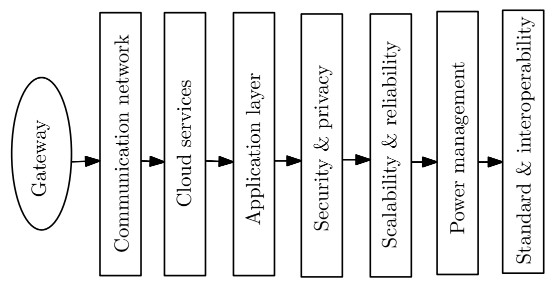


Fig. 1. IoT-WSN framework

Figure 2 representing an IoT-based WSN scenario can be interpreted as a connected undirected graph. Consider a WSN comprised of N wireless sensors strategically deployed within an application. These sensors sense pivotal environmental parameters like temperature, humidity, air quality, etc. The main goal revolves around gathering data from these sensors and effectively transmitting it to a central base station or gateway facilitated by IoT technology.

This article aims to solve an MOP, encapsulating many significant factors to ensure an optimal QoS. The following subsection describes the IoT application services model's Objectives function [9]. The following is the formulation of our proposed MOP, which includes all objectives:

(I) Localization rate (LT): We include LT (the success rate at which the individual sensors are accurately localized) as

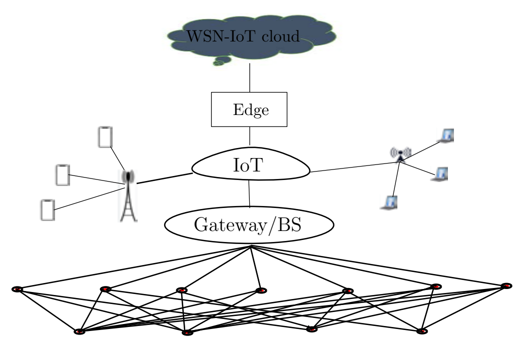


Fig. 2. An instance of an un-directed WSN-IoT network

our first objective (IoT1). However, there isn't a single formula to determine LT, as it depends on the methodology, algorithms, and metrics used for localization. So, we consider a general concept to evaluate this metric in our proposed problem. The goal is to maximize the IoT1 value, as formulated in Equation 1. The ALS_{count} value is computed according to the methodology used in [25].

$$LT = (ALS_{count}/N) * 100, \quad (1)$$

Where, N is a total count of sensors used in our network and ALS_{count} denotes a number of accurately localised sensors.

(II) Energy consumption (E_{total}): We include E_{total} (sum-total energy consumption across all sensors) as our second objective (IoT2). The goal is to minimize IoT2 value, as formulated in Equation 2. Here the representation of different symbols are: S_i: A sensor node with ID i, where $i = 1, 2, \ldots, N$, $RE(S_i)$: Remaining energy level of the S_i before data gathering cycle begins, d_{ij}: Distance between sensors S_i and S_j, E_i: Energy consumed by S_i, $D(S_i)$: $D(S_i) = \min\left\{\text{Initial}_{\text{Data}}(S_i), \frac{RE(S_i)}{E_{T_i}(l_i, d_{ij})}\right\}$ = Maximum amount of data that the node S_i can transmit based on its remaining energy $RE(S_i)$ [26], where $Initial_{Data}$ is initial sensed data availability of sensor S_i. However, the actual data transmitted by the sensor S_i also depends upon the routing method used from [27]. E_{elec} is the quantity of energy consumed by an electronic circuit of a sensor node. The radio energy concept from [28] is considered to determine total network energy expenditure at the time of data communication.

$$\begin{aligned}
\text{Minimize:} \quad & E_{total} = \sum_{i=1}^{N} E_i \\
\text{Subject to:} \quad & E_i \leq RE(S_i) \\
& E_{R_i}(l_i(S_i)) = l_i(S_i) \cdot E_{\text{elec}} \\
& E_{T_i}(l_i, d) = \begin{cases} l_i(S_i)E_{\text{elec}} + l_i(S_i)f_s d^2 & \text{if } d < C^r \\ l_i(S_i)E_{\text{elec}} + l_i(S_i)m_p d^4 & \text{if } d \geq C^r, \end{cases}
\end{aligned} \quad (2)$$

Where f_s and m_p are the energies required to transmit data in the free-space channel and multi-path channel, respectively. C^r is a threshold value for the transmission distance. The term $E_{T_i}(l_i, d)$ specifies the energy depleted by S_i to transmit a data packet of length l_i-bit over the distance d. The term $E_{R_i}(l_i(S_i))$ denotes the energy consumption for receiving l_i bits of data.

(III) Coverage rate (CR_{rate}): The fourth objective (IoT4) of our proposed problem addresses the CR_{rate}, which represents the proportion of the deployment area covered by sensors. In order to calculate CR_{rate}, a circular sensing model is considered. The mathematical representation of CR_{rate} is provided in Equation (3), outlined as follows:

$$CR_{rate} = (Covered_{Area}/Total_{Area}) \times 100, \quad (3)$$

where $Total_{Area}$ denotes the total area of the closed region being monitored and $Covered_{Area} = \sum_{i=1}^{N} \pi * (C^r)^2$. The C^r denotes the sensors' communication range.

(IV) Delay Time (DT): DT is the duration to forward the sensed data of a node S_i to the base-station (BS) using TDMA approach [27] for further processing. As, TDMA allows for the efficient allocation of time slots to each sensor node, thereby facilitating orderly data transmission to the BS. The TDMA has reduced complexity against simultaneous transmissions, as would be the case in models like WCDMA [29] or OFDMA [30]. The Delay Time metric is defined in terms of the fifth objective (IoT5), which aims to minimize its value to ensure real-time data delivery, and is expressed as in Eq.4:

$$DT = \sum_{i=1}^{N} T_i, \quad (4)$$

where T_i denotes the time duration required for sensor node S_i data to be transmitted to the BS.

Proposed Fitness Function This fitness functions of smart IoT application calculated in Eq. 5, all of the objectives $(IoT1, IoT2, IoT3, \&IoT4)$ are turned into a single objective function.

$$\begin{aligned} Fitness = fun_1 \times IoT1 + fun_2 \times IoT2 \\ + fun_3 \times IoT3 + fun_4 \times IoT4 \end{aligned} \quad (5)$$

Where values of fun_1, fun_2, fun_3, & fun_4 are the weights assigned to each of the objective functions.

Fitness function of Multi objectives of the IoT Service Framework: The four objective-based IoT-based services calculated by equations 1, 2, 3, and 4 are non-contradictory. As demonstrated in Eq. 6, all objectives are turned into Multi-objective functions utilizing IoT-based service parameters.

$$\begin{aligned} &\text{obj1(max)} = \text{function in Eq.1}, \ \text{obj2(min)} = \text{function in Eq.2} \\ &\text{obj3(max)} = \text{function in Eq.3}, \ \text{obj4(min)} = \text{function in Eq.4} \end{aligned} \quad (6)$$

The minimization problem for objectives $obj2$ and $obj4$, which represent the estimation of the total energy expenditure across all sensors and the average delay time. The localization rate and coverage rate for Internet of Things services derived from data transmission from sensor nodes are calculated by solving the maximisation problem of objectives $obj1$ and $obj3$, respectively.

IV. Proposed Methodology

In the era of IoT and 5G/6G networks, edge computing has emerged as an upcoming approach to satisfy the growing needs of computation-intensive applications requiring low latency and effective quality assurance. Considering these issues, the next section explains the proposed GWO methodology-based solution to address the proposed problem.

A. Proposed GWO Algorithm

The Grey wolf hunting mechanism used in the GWO algorithm [31] is inspired by nature. In order to create the best hierarchy, the Grey Wolf, a leader, captures the foods of his challenging character. The four components of this system are alpha, beta, delta, and omega. Grey wolves use tracking, surrounding, and attacking techniques as a hunting

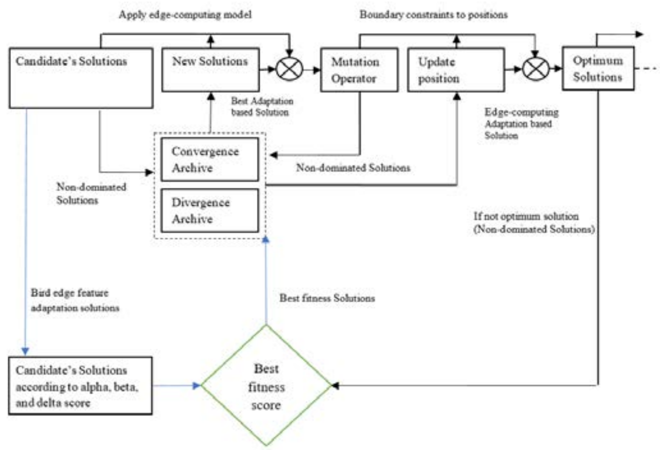


Fig. 3. Proposed architecture of edge computing-based GWO algorithm

Algorithm 1: Proposed GWO Algorithm

Input:
(a) δ_1 = r/3, where random (r) value (0 to 2)
(b) Cr = 0.1 to 0.9
(c) Set the Population Size = 100*D
(d) Iteration (itr) = 1, where itr denotes the current iteration number
Result: Achieve the approximated solution

Step 1: Set the fitness function according to initialization of population
Step 2: Calculate the fitness function of candidate solutions
Step 3: $Rank_i$ helps generate high-ranking best vectors
Step 4: $\vec{P}_\alpha$ = the best search agent
Step 5: $\vec{P}_\beta$ = the second-best search agent
Step 6: $\vec{P}_\delta$ = the third-best search agent
while $(t \neq Max_T)$ **do**
  Step 7: GWO algorithm
  7.1 For each search vector or agent:
  7.2 Calculate the fitness of all search agents
  7.3 Update the position of the current search agent:
  $\vec{P}(itr+1) = \frac{\vec{D}ist(P1_i^t, P2_j^t) + \vec{D}ist(P2_i^t, P3_j^t) + \vec{D}ist(P1_i^t, P3_j^t)}{3}$
  7.4 Update $\vec{P}_\alpha$, $\vec{P}_\beta$, and $\vec{P}_\delta$
  7.5 Update the best optimum value of GWO algorithm
  Step 8: Apply the new bird adaptation-based mutation strategy
  8.1 Apply mutation operator and select donor vector
  if $\vec{P}(itr+1) < \vec{P}T(itr+1)$ **then**
    $\vec{P}(itr+1) = \vec{\alpha}_{\text{best},G} + \delta_1 \times (\vec{P}1_{r_1^i,G} - \vec{P}2_{r_2^i,G}) \times rand(0,1)$
  else
    $\vec{P}T(itr+1) = \vec{\alpha}_{\text{rand},G} + \delta_2 \times (\vec{P}2_{r_1^i,G} - \vec{P}3_{r_2^i,G}) \times rand(0,1)$
  end
  Note: If $\vec{P}(itr+1)$ does not find a better solution (i.e., optimal value), then $\vec{P}T(itr+1)$ will be used for Pareto ranking-based mutation strategy
  8.2 Apply the Pareto ranking-based solutions in the selection strategy
  $t = t + 1$
end
Step 9: Near-optimal Solution with high Convergence Rate

strategy. Hunting involves three steps: looking for prey, surrounding, and attacking prey. The proposed algorithm outlined in Algorithm 1 and its architectural representation in Figure 3 demonstrate the methodology for selecting the optimal solution within the GWO algorithm. This selection process is guided by the hunting feature adaptation strategy, representing a natural process of birds in edge computing, similar to selecting the best food source during their attacks. Further insights into this process of feature selection within the context of edge computing are discussed thereafter.

B. Proposed Bird-edge-computing adaption strategy

The bird-edge-computing feature selection plays a crucial role in the exploration and exploitation strategy of our algorithm. This approach effectively mitigates early convergence and local optima issues, thereby enhancing exploration and exploitation capabilities. The proposed GWO Algorithm employs this strategy as a mutation mechanism, inspired by the adaptive behavior of eagles. The updated position of the object is determined by incorporating the best solutions identified throughout the entire search area. Specifically, this strategy involves:

1. **Exploration of Objects**: The algorithm examines various objects within the search space, adjusting their positions based on relative distances to optimize target values efficiently.

2. **Mathematical Formulation**: The adaptation process is mathematically expressed in Eqs. 7 and 8, which detail how the position update is influenced by the best objects:

$$\vec{P}(itr+1) = \vec{\alpha}_{\text{best},G} + \delta_1 \times (\vec{P}1_{r_1^i,G} - \vec{P}2_{r_2^i,G}) \times rand(0,1) \tag{7}$$

In this equation, $\vec{P}(itr+1)$ represents the updated position of the object, where $\vec{\alpha}_{\text{best},G}$ denotes the best object in the entire search area. The vectors $\vec{P}1_{r_1^i,G}$ and $\vec{P}2_{r_2^i,G}$ are ranking-based vectors, and δ_1 signifies the exploration range, defined by $rand(0,1.7)$.

3. **Distance Optimization**: The term $(\vec{P}1_{r_1^i,G} - \vec{P}2_{r_2^i,G})$ signifies the distance between the hunter (our algorithm) and the target (the optimal solutions). This mechanism addresses the key challenge of optimizing distance in the search area by focusing on the nearest distance between the hunter and the targets.

4. **Exploitation Strategy**: Additionally, the process is refined in Eq. 8 to ensure effective exploitation of the search space:

$$\vec{P}(itr+1) = \vec{\alpha}_{\text{rand},G} + \delta_2 \times (\vec{P}2_{r_1^i,G} - \vec{P}3_{r_2^i,G}) \times rand(0,2) \tag{8}$$

Here, $\vec{\alpha}_{\text{rand},G}$ represents a randomly selected best object across the search area, while δ_2 indicates the exploitation range, defined by $rand(0,2)$. This structured approach ensures that both exploration and exploitation are dynamically balanced, leading to continual improvements in fitness and overall optimization performance.

C. Proposed Hybrid Algorithm

The proposed Hybrid Algorithm is summarized in Algorithm 2. This Algorithm starts by randomly initializing the positions of a population of grey wolves within the search space. Each position represents a potential solution: a configuration of tasks and resources for edge computing. The alpha, beta,

Algorithm 2: Proposed Hybrid Algorithm

Input : Objective function $obj\_function$, Lower bounds lb, Upper bounds ub, Dimensionality dim, Population size $population\_size$, Maximum iterations $max\_iterations$
Output: Best solution found (alpha wolf)

Initialize positions randomly within the search space (denoted as positions$[i]$ for the i^{th} agent);
Initialize alpha position α_{pos}, beta position β_{pos}, delta position δ_{pos};
Initialize alpha score α_{score}, beta score β_{score}, delta score δ_{score};
for $iteration = 1$ **to** $max\_iterations$ **do**
  for $i = 1$ **to** $population\_size$ **do**
    Evaluate fitness $fitness$ of the current position positions$[i]$;
    if $fitness < \alpha_{score}$ **then**
      $\alpha_{score} = fitness$;
      $\alpha_{pos} =$ positions$[i]$;
    end
    if $\alpha_{score} < fitness < \beta_{score}$ **then**
      $\beta_{score} = fitness$;
      $\beta_{pos} =$ positions$[i]$;
    end
    if $\alpha_{score} < fitness < \beta_{score} < \delta_{score}$ **then**
      $\delta_{score} = fitness$;
      $\delta_{pos} =$ positions$[i]$;
    end
    Update position positions$[i]$ using the proposed GWO algorithm;
    Apply boundary constraints to positions$[i]$;
  end
end
return α_{pos} *(Best solution found)*

and delta positions are initialized as empty vectors, and their corresponding scores are set to infinity. This algorithm iterates for a predefined number of maximum iterations. The fitness is evaluated using an objective function for each grey wolf in the population. The objective function measures the solution's performance in terms of edge computing, considering factors such as latency, resource utilization, energy efficiency, or other relevant metrics. Based on the fitness values, the grey wolves with better fitness values than the current alpha, beta, and delta positions are updated. If a grey wolf has better fitness than the alpha position, it becomes the new one. Similarly, if a grey wolf has a fitness value between the alpha and beta positions, it becomes the new beta position. Likewise, if a grey wolf has a fitness value between the alpha, beta, and delta positions, it becomes the new delta position. The position of each grey wolf is updated using the GWO equation. This equation considers the distances to the alpha and beta positions and random coefficients to determine the direction and magnitude of the update. The position update equation aims to imitate the grey wolves' leadership hierarchy and guide the search space exploration. Apply Boundary Constraints: After updating the position, boundary constraints are applied to ensure the new position remains within the defined search space. This step is essential to maintain valid configurations for edge computing environments. After the termination criterion is met (maximum iterations reached), the algorithm returns the best solution found, which corresponds to the alpha position. This solution represents an optimized configuration of tasks and resources for edge computing, considering the objective function and the constraints of the problem. The grey wolves' social behavior and hunting mechanisms inspire the GWO algorithm. It imitates the hierarchy and coordination among alpha, beta, and delta wolves for efficient exploration and exploitation of the search space. Finding the optimal value is archived when applying this technique to edge computing optimization, as seen in the procedure in Fig. 3.

As observed above, the proposed fitness function is formulated for the IOT-WSN environments, which includes the four objective fitness function constraints such as $obj1$, $obj2$, $obj3$, and $obj4$. These constraints are implemented once the position is updated to make sure the new position stays inside the specified search space. Based on the stated objectives and considering the effects of edge computing, the solution's effectiveness is rated. Boundary constraints are enforced once the position is updated to make sure the new position stays inside the specified search space. This step is essential to keep the IoT-based WSN with Edge Computing configured correctly and prevent parameter values from exceeding predetermined limits. The main loop runs for the chosen number of iterations, giving the grey wolves time to investigate and fine-tune their places. The method delivers the best solution discovered, corresponding to the alpha position when the termination requirement is satisfied (maximum iterations attained). Considering the stated aims and restrictions, this solution represents an optimized setup of parameters for the IoT-based WSN with Edge Computing. This method aims to discover the best configurations that improve the system's performance, energy efficiency, network coverage, and edge computing resource utilization by using the GWO algorithm for IoT-based WSN optimization with Edge Computing. By combining edge computing, the IoT-based WSN application may process data effectively closer to the data source, cutting latency and improving overall efficiency.

1) Time Complexity Analysis of Proposed Hybrid Algorithm: The run-time complexity analysis in the stepwise sequence is as follows:

Initialization:

- **Position Initialization:** Initializing positions within the search space takes $\mathcal{O}(population\_size)$.
- **Alpha, Beta, and Delta Initialization:** Setting initial values for α_{pos}, β_{pos}, δ_{pos}, α_{score}, β_{score}, and δ_{score} takes $\mathcal{O}(1)$.

Main Loop: The algorithm iterates for a maximum of $max\_iterations$:

- **Fitness Evaluation:** Evaluating fitness for each position, which depends on the number of objectives B, takes $\mathcal{O}(max\_iterations \times population\_size \times B)$.
- **Update Scores:** Updating α_{score}, β_{score}, and δ_{score} requires constant time, contributing $\mathcal{O}(max\_iterations)$.
- **Position Update:** Updating each position involves $\mathcal{O}(population\_size)$.
- **Apply Boundary Constraints:** Applying boundary constraints also takes $\mathcal{O}(population\_size)$.

The overall time complexity can be summarized as: $\mathcal{O}\left(max\_iterations \times population\_size \times B\right)$. This complexity demonstrates that the introduced hybrid algorithm is

TABLE II. Control Parameters of Evolutionary Algorithms

Sr No.	Algorithm	Parameter	Characteristics of Algorithm
1	NSGA-III [14]	Selection strategy	Survival of the fittest
		Crossover probability (Cr)	Cr (0,1)
		Mutation probability (Mp)	Mp (0,1)
2	MOPSO [8]	Inertia weight	1.3
		Acceleration constants (c1, c2)	c1 (1.35), c2 (2.4)
		Swarm size	100-500
3	MOWOA Algo [13]	Moving convergence rate	0.12
		Local convergence rate	0.35
		Global convergence rate	rand(0,1)
4	MOAE-D Algo [16]	Population size	100
		Scalarization function	According to given algorithm
		Penalty parameter	Random selection
5	Proposed Algorithm	Number of grey wolves	100
		Range of dimensions	3 and 20
		Alpha, Beta, Gamma	Wolf positions
		Number of iterations	1000

suitable for solving the multi-objective problem in the dynamic environment. This indicates that our proposed algorithm has comparable performance characteristics to the other types of multi-objective optimization algorithms namely MOPSO,PAES,SPEA2 and NSGA-II.

V. RESULTS

This section presents the result analysis obtained from the evaluation of the proposed method and comparison of its efficacy to other evolutionary methods such as NSGA-III [14], MOPSO [8], MOWOA [13], and MOEA-D [16]. Considering our novel proposed problem, the proposed method is applied to evaluate the QoS performance of IoT applications.

A. Experimental setup

For the evaluation, we create a setup of an IoT-enabled WSN framework $(150 \times 150)m^2$, where 100 sensors for service requests/responses have been distributed equally. The sensors have initial data availability (0 - 16) Mb [26], remaining energy levels (500 - 1000) mJ [26], data transmission rate 20 $kbps$ [26], and communication range C^r = 20 m [24]. Additionally, we used 75 active sensors that respond to requests from processes, people, and objects. The term "service providers" refers to these 100 sensors. We have chosen the experimental area in a grid of 15 by 15. The 30 randomly generated service requests for the experiment have been chosen to be related to the MATLAB network simulator used to construct the simulation environment, giving exact control over communication models and topology. A true data set derived from actual sensor deployments was used for the experiment, and necessary pre-processing procedures were carried out to guarantee data quality. Multiple repetitions were used in the experimental design to guarantee statistical robustness. Each optimization run was given the ability to reach convergence after a specific number of generations. In order to terminate the optimization process once a good Pareto front (set on non-dominated solutions) was attained, convergence criteria were created. The sensor parametric Table is listed in Table III.

By employing the proposed technique, we estimated and fine-tuned all parameters outlined in Table II. The approach was tested using IoT scenarios involving services. Specifically, the proposed method was evaluated on IoT-based services to compare the resulting Pareto fronts within a 4-objective series. This method was applied to analyze IoT services, encompassing computations related to localization rate, total energy consumption, coverage rate, and delay time.

TABLE III. Sensor-network Parameters

Parameter	Value
IoT-enabled WSN framework	150 m × 150 m
Initial data availability	(0 MB - 16 MB)
Remaining energy levels	(500 mJ - 1000 mJ)
Data transmission rate	20 kbps
Data packet size	4000 bits
Data aggregation ratio	10%
E_{elec}	50 nJ/bit
f_s	10 pJ/bit/m^2
m_p	0.001 pJ/bit/m^4
EDA (Energy for data aggregation)	5 nJ//bit
Communication range C^r	20 m

B. Analysis & Discussion

The proposed approach is subjected to comparison against other state-of-the-art methods to assess its robustness and versatility in identifying both maximized and minimized solutions. The comparative analysis is described as follows:

1) Pareto Front Analysis: The four-objective-based analysis is conducted to demonstrate that the proposed hybrid method achieves better solutions in the context of smart IoT applications. When addressing the multiple conflicting objectives in Equation 6, the proposed method yields a Pareto rank. The values shown in Table IV showcase each algorithm's ability to provide diverse solutions, enabling decision-makers to choose solutions based on their specific priorities and requirements. This comprehensive analysis underscores the trade-offs inherent in multi-objective optimization, and the presented values contribute to the characterization of the Pareto front for each algorithm. From Table IV, it is observed that the proposed hybrid method yields superior performance by obtaining better quality, diversity, and convergence of Pareto front solutions, as evidenced by outperforming NSGA-III, MOEA-D, MOPSO, and MOWOA methods.

2) Localization Rate of IoT-based WSN Framework: We analyze and report localization rate metrics for all algorithms, utilizing the novel fitness function of this metric defined in Eq. 1. The proposed method aims to optimize this metric value for smart IoT applications, ensuring that the population sizes of 50, 100, 150, and 200 align with the optimal number. In a series of 100 trials, the proposed method consistently outperforms the respective values of the four referenced Multi-objective algorithms. This is due to the proposed method's improved capability for exploration and exploitation. The proposed method can more precisely and accurately find optimal solutions by balancing the search process due to the incorporation of the bird-edge adaptation mechanism. Better localization performance is attained due to a tighter clustering of non-dominated solutions and a more accurate convergence towards the true Pareto front.

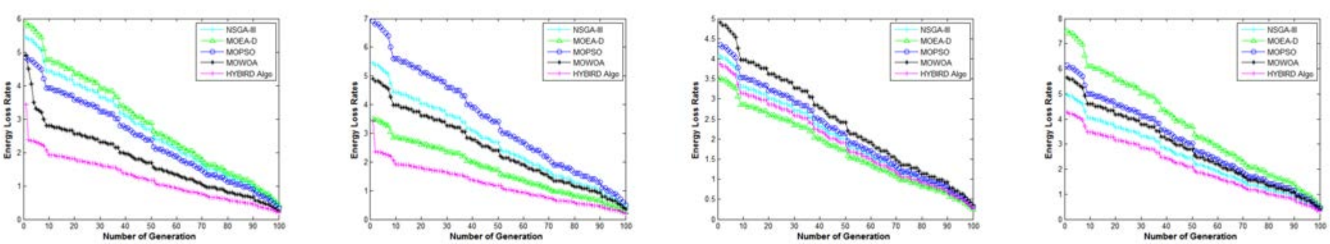
3) Energy consumption of IoT-based WSN Framework: The proposed hybrid and other evolutionary algorithms report the total energy consumption cost metric values. The comparison shows that the proposed algorithm delivers optimal energy utilization values for IoT applications across population sizes ranging from 50 to 200. Across 100 or more trials, the energy consumption values of the proposed method consistently outperform others. Moreover, our proposed algorithm exhibits

TABLE IV. Fitness Cost: The proposed algorithm compared with Evolutionary algorithms on Smart IoT Application

No. of Gen.	No. of Runs	NSGA-III [15]		MOEA-D [17]		MOPSO Algo [8]		MOWOA Algo [14]		Proposed Hybrid Algo	
		Best Fit	Worst Fit	Best Fit	Worst Fit	Best Fit	Worst Fit	Best Fit	Worst Fit	Best Fit	Worst Fit
10	30	0.491062	0.417736	0.486618	0.395516	0.495506	0.424402	0.504394	0.43329	0.508838	0.437734
20	30	0.542555	0.46154	0.537645	0.43699	0.547465	0.468905	0.557285	0.478725	0.562195	0.483635
30	30	0.594048	0.505344	0.588672	0.478464	0.599424	0.513408	0.610176	0.52416	0.615552	0.529536
40	30	0.645541	0.549148	0.639699	0.519938	0.651383	0.557911	0.663067	0.569595	0.668909	0.575437
50	30	0.697034	0.592952	0.690726	0.561412	0.703342	0.602414	0.715958	0.61503	0.722266	0.621338
60	30	0.748527	0.636756	0.741753	0.602886	0.755301	0.646917	0.768849	0.660465	0.775623	0.667239
70	30	0.80002	0.68056	0.79278	0.64436	0.80726	0.69142	0.82174	0.7059	0.82898	0.71314
80	30	0.851513	0.724364	0.843807	0.685834	0.859219	0.735923	0.874631	0.751335	0.882337	0.759041
90	30	0.903006	0.768168	0.894834	0.727308	0.911178	0.780426	0.927522	0.79677	0.935694	0.804942
100	30	0.932399	0.793172	0.923961	0.750982	0.940837	0.805829	0.957713	0.822705	0.966151	0.831143

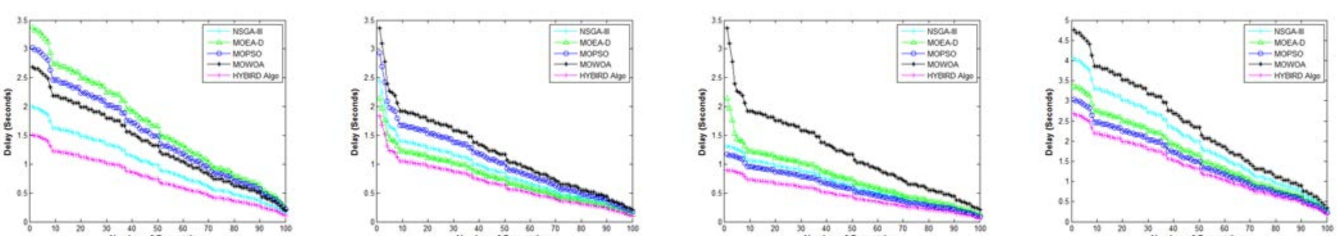
TABLE V. Energy Consumption and Delay: The proposed algorithm compared with Evolutionary algorithms on Smart IoT Application

Objectives	No. of Gen.	NSGA-III [15]		MOEA-D [17]		MOPSO Algo [8]		MOWOA Algo [14]		Proposed Hybrid Algo	
		Worse Fit	Best Fit	Worse Fit	Best Fit	Worse Fit	Best Fit	Worse Fit	Best Fit	Worse Fit	Best Fit
Energy Consumption	10	5.183776	4.409728	5.136864	4.175168	5.230688	4.480096	5.324512	4.57392	5.089952	4.151712
	20	4.179331	3.555268	4.141509	3.366158	4.217153	3.612001	4.292797	3.687645	4.103687	3.347247
	30	2.965599	2.522772	2.938761	2.388582	2.992437	2.563029	3.046113	2.616705	2.911923	2.375163
	40	1.988779	1.691812	1.970781	1.601822	2.006777	1.718809	2.042773	1.754805	1.952783	1.592823
	50	0.952731	0.810468	0.944109	0.767358	0.961353	0.823401	0.978597	0.840645	0.935487	0.763047
	60	0.756041	0.643148	0.749199	0.608938	0.762883	0.653411	0.776567	0.667095	0.742357	0.605517
	70	0.687531	0.584868	0.681309	0.553758	0.693753	0.594201	0.706197	0.606645	0.675087	0.550647
	80	0.643331	0.547268	0.637509	0.518158	0.649153	0.556001	0.660797	0.567645	0.631687	0.515247
	90	0.554931	0.472068	0.549909	0.446958	0.559953	0.479601	0.569997	0.489645	0.544887	0.444447
	100	0.333931	0.284068	0.330909	0.268958	0.336953	0.288601	0.342997	0.294645	0.327887	0.267447
Delay	10	3.928717	3.342076	3.893163	3.164306	3.964271	3.395407	4.035379	3.466515	3.857609	3.146529
	20	3.125161	2.658508	3.096879	2.517098	3.153443	2.700931	3.210007	2.757495	3.068597	2.502957
	30	2.455531	2.088868	2.433309	1.977758	2.477753	2.122201	2.522197	2.166645	2.411087	1.966647
	40	2.209779	1.879812	2.189781	1.779822	2.229777	1.909809	2.269773	1.949805	2.169783	1.769823
	50	1.132183	0.963124	1.121937	0.911894	1.142429	0.978493	1.162921	0.998985	1.111691	0.906771
	60	0.660127	0.561556	0.654153	0.531686	0.666101	0.570517	0.678049	0.582465	0.648179	0.528699
	70	0.546091	0.464548	0.541149	0.439838	0.551033	0.471961	0.560917	0.481845	0.536207	0.437367
	80	0.426972	0.363216	0.423108	0.343896	0.430836	0.369012	0.438564	0.37674	0.419244	0.341964
	90	0.338351	0.287828	0.335289	0.272518	0.341413	0.292421	0.347537	0.298545	0.332227	0.270987
	100	0.291941	0.248348	0.289299	0.235138	0.294583	0.252311	0.299867	0.257595	0.286657	0.233817



(a) Population Size 50 (b) Population Size 100 (c) Population Size 150 (d) Population Size 200

Fig. 4. Fitness Function of Evolutionary Algorithms: Number of generations v/s Energy Consumption

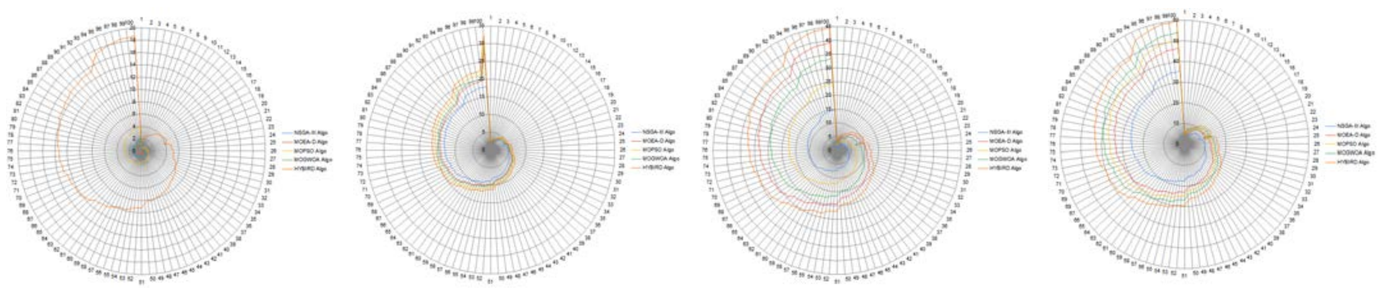


(a) Population Size 50 (b) Population Size 100 (c) Population Size 150 (d) Population Size 200

Fig. 5. Fitness Function of Evolutionary Algorithms: Number of generations v/s Delay

TABLE VI. Coverage Rate: The proposed algorithm compared with Evolutionary algorithms on Smart IoT Application

No. of Gen.	No. of Runs	NSGA-III [15]		MOEA-D [17]		MOPSO Algo [8]		MOWOA Algo [14]		Proposed Hybrid Algo	
		Best Fit	Worse Fit	Best Fit	Worse Fit	Best Fit	Worse Fit	Best Fit	Worse Fit	Best Fit	Worse Fit
10	30	0.466531	0.396868	0.462309	0.375758	0.470753	0.403201	0.479197	0.411645	0.483419	0.415867
20	30	0.510731	0.434468	0.506109	0.411358	0.515353	0.441401	0.524597	0.450645	0.529219	0.455267
30	30	0.554931	0.472068	0.549909	0.446958	0.559953	0.479601	0.569997	0.489645	0.575019	0.494667
40	30	0.599131	0.509668	0.593709	0.482558	0.604553	0.517801	0.615397	0.528645	0.620819	0.534067
50	30	0.643331	0.547268	0.637509	0.518158	0.649153	0.556001	0.660797	0.567645	0.666619	0.573467
60	30	0.687531	0.584868	0.681309	0.553758	0.693753	0.594201	0.706197	0.606645	0.712419	0.612867
70	30	0.731731	0.622468	0.725109	0.589358	0.738353	0.632401	0.751597	0.645645	0.758219	0.652267
80	30	0.775931	0.660068	0.768909	0.624958	0.782953	0.670601	0.796997	0.684645	0.804019	0.691667
90	30	0.820131	0.697668	0.812709	0.660558	0.827553	0.708801	0.842397	0.723645	0.849819	0.731067
100	30	0.908531	0.772868	0.900309	0.731758	0.916753	0.785201	0.933197	0.801645	0.941419	0.809867



(a) Population Size 50 (b) Population Size 100 (c) Population Size 150 (d) Population Size 200

Fig. 6. Fitness Function of Evolutionary Algorithms: Number of generations v/s Coverage Rate

minimal variances in the results across succeeding generations (for the number of runs more than 100). Each algorithm underwent 30 runs to generate average outcomes. Table V compiles the results over 100 generations, revealing a notable reduction in energy usage within the IoT application framework. The proposed technique achieves remarkable energy efficiency by optimizing resource utilization more effectively based on edge computing selection. Its advanced adaptation mechanism improves searching efficiency, leading to better solutions that require less computational effort and energy, as reported in Figure 4: 4(a), 4(b), 4(c), and 4(d). Each Figure illustrates the X-axis representing the number of generations and the Y-axis depicting the total energy consumption metric values (in Joule).

4) Delay time of IoT-based WSN Framework: The proposed method evaluates delay times for IoT-WSN applications across various population sizes (50, 100, 150, and 200), as depicted in Figure 5. It is evident from the figure that our proposed method consistently yields superior results compared to traditional evolutionary techniques. Notably, the conventional optimization methods require larger populations to achieve similar results. A more comprehensive assessment based on 100 generations is also available in Table V. These statistics reinforce that our proposed technique consistently outperforms other evolutionary methods, resulting in shorter delay times. This advantage can be attributed to incorporating an edge computing selection strategy within the GWO algorithm. Remarkably, across 100 trials, the delay time achieved by our proposed method consistently remains lower than that of the referenced four Multi-objective algorithms by optimizing resource allocation more effectively. Its improved adaptation mechanism ensures quicker convergence to optimal solutions, thereby minimizing latency and improving overall network performance. Furthermore, among these four methods, the proposed algorithm excels in diversity and convergence. These qualities are illustrated through graphical representations in Figure 5(a), 5(b), 5(c), and 5(d), where the X-axis represents the number of generations and the Y-axis represents delay time in seconds.

5) Coverage Rate of IoT-based WSN Framework: This subsection presents the coverage rate metric results the proposed algorithm achieves, comparing them to those of traditional evolutionary algorithms using the novel fitness function described in Eq. 4. The proposed method aims to maximize this metric value for smart IoT applications, aligning with ideal population sizes of 50, 100, 150, and 200. Through a series of 100 trials, the proposed method consistently outperforms the corresponding values of the referenced methods by effectively optimizing resource distribution. Its advanced adaptation mechanism ensures a more comprehensive and effective coverage, thereby enhancing the overall network performance and reliability. Notably, among these methods, the proposed algorithm excels in both diversity and convergence, as illustrated in Figure 6: 6(a), 6(b), 6(c), and 6(d).The proposed algorithm demonstrates a better coverage rate among the four traditional algorithms, as outlined in Table VI.

VI. Conclusion and Future Research Direction

This article introduces a state-of-the-art method, which aims to enhance QoS in IoT applications by integrating Multi-objective Optimization and Edge Intelligence. The method leverages bird-edge-computing with GWO to balance global and local optimal solutions and optimise feature combinations. A novel edge computing feature selection concept is presented to address potential convergence issues during the iterative process. The algorithm's efficiency is heightened by incorporating unique attributes, and the bird-edge-computing adaptation continually adjusts to achieve the global optimal solution.

The approach is evaluated in an IoT-based WSN environment, focusing on QoS improvement. Various objectives are considered, and performance evaluation demonstrates better QoS achievement compared to existing evolutionary methods. Future research directions involve adapting the algorithm for diverse domains and exploring its potential for addressing complex problems. Additionally, investigating scalability and robustness through validation with larger problem instances would provide valuable insights for real-world applications.

REFERENCES

[1] M. Li, "A lightweight architecture for query-by-example keyword spotting on low-power iot devices," *IEEE Transactions on Consumer Electronics*, vol. 69, no. 1, pp. 65–75, 2023.

[2] S. Sun, T. Peng, H. Huang, Y. Wang, X. Zhang, and Y. Zhou, "Iot motion tracking system for workout performance evaluation: A case study on dumbbell," *IEEE Transactions on Consumer Electronics*, vol. 69, no. 4, pp. 798–808, 2023.

[3] P. S. Chatterjee, N. K. Ray, and S. P. Mohanty, "Livecare: An iot-based healthcare framework for livestock in smart agriculture," *IEEE Transactions on Consumer Electronics*, vol. 67, no. 4, pp. 257–265, 2021.

[4] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on multi-access edge computing for internet of things realization," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2961–2991, 2018.

[5] H. R. Chi, M. de Fátima Domingues, H. Zhu, C. Li, K. Kojima, and A. Radwan, "Healthcare 5.0: In the perspective of consumer internet-of-things-based fog/cloud computing," *IEEE Transactions on Consumer Electronics*, vol. 69, no. 4, pp. 745–755, 2023.

[6] H. Zhang, J. Luo, Y. Tu, R. Wang, D. Wu, and J. Yang, "Microservice deployment mechanism with diversified qos requirements for smart health system in industry 5.0," *IEEE Transactions on Consumer Electronics*, vol. 69, no. 4, pp. 869–880, 2023.

[7] R. Chaudhry, S. Tapaswi, and N. Kumar, "Fz enabled multi-objective pso for multicasting in iot based wireless sensor networks," *Information Sciences*, vol. 498, pp. 1–20, 2019.

[8] M. Salimian, M. Ghobaei-Arani, and A. Shahidinejad, "An evolutionary multi-objective optimization technique to deploy the iot services in fog-enabled networks: an autonomous approach," *Applied Artificial Intelligence*, vol. 36, no. 1, p. 2008149, 2022.

[9] S.-s. Fang, Z.-y. Chai, and Y.-l. Li, "Dynamic multi-objective evolutionary algorithm for iot services," *Applied Intelligence*, vol. 51, no. 3, pp. 1177–1200, 2021.

[10] B. Barakat, A. Taha, R. Samson, A. Steponenaite, S. Ansari, P. M. Langdon, I. J. Wassell, Q. H. Abbasi, M. A. Imran, and S. Keates, "6g opportunities arising from internet of things use cases: a review paper," *Future Internet*, vol. 13, no. 6, p. 159, 2021.

[11] K. Bairagi, S. Mitra, and U. Bhattacharya, "Multi-objective optimization for coverage aware energy consumption in wireless 3d video sensor network," *Computer Communications*, vol. 195, pp. 262–280, 2022.

[12] B. Natesha and R. M. R. Guddeti, "Adopting elitism-based genetic algorithm for minimizing multi-objective problems of iot service placement in fog computing environment," *Journal of Network and Computer Applications*, vol. 178, p. 102972, 2021.

[13] M. Huang, Q. Zhai, Y. Chen, S. Feng, and F. Shu, "Multi-objective whale optimization algorithm for computation offloading optimization in mobile edge computing," *Sensors*, vol. 21, no. 8, p. 2628, 2021.

[14] A. Zebouchi and Y. Aklouf, "prtmnsga-iii: a novel multi-objective algorithm for qos-aware multi-cloud iot service selection," *Annals of Telecommunications*, pp. 1–22, 2024.

[15] M. Zaborski, M. Woźniak, and J. Mańdziuk, "Multidimensional red fox meta-heuristic for complex optimization," *Applied Soft Computing*, vol. 131, p. 109774, 2022.

[16] S. P. Singh, G. Dhiman, W. Viriyasitavat, and S. Kautish, "A novel multi-objective optimization based evolutionary algorithm for optimize the services of internet of everything," *IEEE Access*, vol. 10, pp. 106 798–106 811, 2022.

[17] R. Shrestha, A. Mishra, R. Bajracharya, S. Sinaei, and S. Kim, "6g network for connecting cps and industrial iot (iiot)," in *Cyber-Physical Systems for Industrial Transformation*. CRC Press, 2023, pp. 17–38.

[18] W. Wei, R. Yang, H. Gu, W. Zhao, C. Chen, and S. Wan, "Multi-objective optimization for resource allocation in vehicular cloud computing networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25 536–25 545, 2022.

[19] C. Li, K. Jiang, Y. Zhang, L. Jiang, Y. Luo, and S. Wan, "Deep reinforcement learning-based mining task offloading scheme for intelligent connected vehicles in uav-aided mec," *ACM Transactions on Design Automation of Electronic Systems*, 2024.

[20] L. Wang, X. Deng, J. Gui, X. Chen, and S. Wan, "Microservice-oriented service placement for mobile edge computing in sustainable internet of vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 9, pp. 10 012–10 026, 2023.

[21] Q. Qi, X. Chen, A. Khalili, C. Zhong, Z. Zhang, and D. W. K. Ng, "Integrating sensing, computing, and communication in 6g wireless networks: Design and optimization," *IEEE Transactions on Communications*, vol. 70, no. 9, pp. 6212–6227, 2022.

[22] B. G. Sheena and N. Snehalatha, "Multi-objective metaheuristic optimization-based clustering with network slicing technique for internet of things-enabled wireless sensor networks in 5g systems," *Transactions on Emerging Telecommunications Technologies*, vol. 34, no. 8, p. e4626, 2023.

[23] F. Khosroabadi, F. Fotouhi-Ghazvini, and H. Fotouhi, "Scatter: Service placement in real-time fog-assisted iot networks," *Journal of Sensor and Actuator Networks*, vol. 10, no. 2, p. 26, 2021.

[24] N. Kumar, D. R. Edla, D. Dash, G. Swain, and T. Shankar, "Energy-efficient and delay-sensitive-based data gathering technique for multi-hop wsn using path-constraint mobile element," *Wireless Networks*, pp. 1–19, 2023.

[25] P. Kristalina, G. Hendrantoro *et al.*, "Node localization methods with improved accuracy for wireless sensor networks," in *TENCON 2011-2011 IEEE Region 10 Conference*. IEEE, 2011, pp. 535–539.

[26] D. Dash, N. Kumar, P. P. Ray, and N. Kumar, "Reducing data gathering delay for energy efficient wireless data collection by jointly optimizing path and speed of mobile sink," *IEEE Systems Journal*, vol. 15, no. 3, pp. 3173–3184, 2020.

[27] F. A. Mohammed, N. Mekky, H. H. Suleiman, and N. A. Hikal, "Sectored leach (s-leach): An enhanced leach for wireless sensor network," *IET Wireless Sensor Systems*, vol. 12, no. 2, pp. 56–66, 2022.

[28] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on wireless communications*, vol. 1, no. 4, pp. 660–670, 2002.

[29] J. Krnic and S. Krco, "Impact of wsn applications' generated traffic on wcdma access networks," in *2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, 2008, pp. 1–5.

[30] H. Zhang, H. Xing, J. Cheng, A. Nallanathan, and V. C. Leung, "Secure resource allocation for ofdma two-way relay wireless sensor networks without and with cooperative jamming," *IEEE Transactions on industrial informatics*, vol. 12, no. 5, pp. 1714–1725, 2015.

[31] S. Mirjalili, S. Saremi, S. M. Mirjalili, and L. d. S. Coelho, "Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization," *Expert systems with applications*, vol. 47, pp. 106–119, 2016.