scientific reports

OPEN

Multi-objective quantum hybrid evolutionary algorithms for enhancing quality-of-service in internet of things

Shailendra Pratap Singh[1], Gyanendra Kumar[2]✉, Umakant Ahirwar[3], Shitharth Selvarajan[4,5,6]✉ & Firoz Khan[7]

In the context of Internet of Things (IoT), optimizing quality of service (QoS) parameters is a critical challenge due to its heterogeneous and resource-constrained nature. This paper proposes a novel quantum-inspired multi-objective optimization algorithm for IoT service management. Traditional multi-objective optimization algorithms often face limitations such as slow convergence and susceptibility to local optima, reducing their effectiveness in complex IoT environments. To address these issues, we introduce a quantum-inspired hybrid algorithm that combines the strengths of Multi-Objective Grey Wolf Optimization Algorithm (MOGWOA) and Multi-Objective Whale Optimization Algorithm (MOWOA), enhanced with quantum principles. This novel integration overcomes the limitations of traditional algorithms by improving convergence speed and avoiding local optima. The hybrid algorithm enhances QoS in IoT applications by achieving superior optimization in terms of energy efficiency, latency reduction, convergence, and coverage cost. The incorporation of quantum-inspired mechanisms, such as quantum position and behavior, strengthens the exploration and exploitation capabilities of the algorithm, enabling faster and more accurate optimization. Extensive simulations and testing demonstrate the proposed method's superior performance compared to existing algorithms, validating its effectiveness in addressing key IoT challenges.

In recent years, the rapid growth of IoT networks has necessitated the development of sophisticated optimization algorithms to efficiently manage the multitude of devices and data traffic. As highlighted by Langley et al.[1], the Internet of Everything (IoE) paradigm emphasizes the importance of smart device management in modern business models. Fang et al.[2] introduced a dynamic multi-objective evolutionary algorithm tailored to the dynamic nature of IoT environments, showcasing its effectiveness in addressing IoT service management. These studies underscore the critical role hybrid algorithms play in managing the complexities of IoT networks.

Multi-objective optimization algorithms have proven effective in addressing challenges such as impractical regions, local fronts, and generating diverse optimal solutions. However, managing conflicting objectives and the complexity of metaheuristics within this framework remain significant challenges. Unlike traditional single-objective methods, many-objective optimization problems (MaOP) handle numerous conflicting objectives, leveraging Pareto-based techniques to provide decision-makers with optimal solutions that balance trade-offs[3–7]. To address these gaps, this research proposes hybrid algorithms enhanced with quantum-inspired techniques to improve effectiveness in tackling complex optimization challenges.

The literature highlights various evolutionary approaches for multi-objective optimization, with selection methods such as mating and environmental selection playing a key role[8]. This work focuses on optimizing the radius configuration to extend IoT network lifetimes, employing an enhanced grey Wolf Optimization (GWO) algorithm combined with a novel energy-harvesting fitness function. Prior studies, such as Deshmukh et al.[9], demonstrated the potential of quantum entanglement-inspired GWO in diverse applications, while Li et al.[10]

[1]Department of Computer Science and Engineering, Madan Mohan Malaviya University of Technology Gorakhpur-273010 (U.P.), Gorakhpur, UP, India. [2]Department of IoT and IS, Manipal University Jaipur, Jaipur, India. [3]Computer Engineering and Application, GLA University, Mathura, UP, India. [4]Department of Computer Science, Kebri Dehar University, Kebri Dehar 3060, Ethiopia. [5]Department of Computer Science and Engineering, Chennai Institute of Technology, Chennai, India. [6]Centre for Research Impact & Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura 140401, Punjab, India. [7]Center for Information and Communication Sciences, Ball State University, Muncie, USA. ✉email: gyanendra.kumar@jaipur.manipal.edu; shitharths@kdu.edu.et

applied a chaotic quantum Whale Optimization Algorithm (WOA) to berth-crane allocation, and Alamir et al.[11] developed a quantum artificial rabbit optimizer for microgrid energy management. These adaptations, including the Multi-Objective Whale Optimization Algorithm (MOWOA), ensure Pareto-optimal solutions with enhanced diversity and convergence[12]. Applications in energy systems, transportation, and design optimization further validate the efficiency of these quantum-inspired approaches in addressing complex IoT challenges[13,14].

Highlights of the problem description

The IoT ecosystem faces significant challenges in managing diverse QoS requirements efficiently, which is critical given the heterogeneous nature of IoT applications[6,7,15,16]. These challenges are particularly acute in healthcare, where optimizing services involves balancing objectives such as cost, delay, and sensor energy. This research proposes an approach that surpasses existing methodologies in addressing these objectives[17]. Optimizing QoS parameters-including energy consumption, delay, convergence cost, and coverage cost-presents a complex, multi-dimensional challenge. Traditional evolutionary algorithms often struggle with issues like slow convergence and susceptibility to local optima, especially as the number of objectives and decision variables increases. This study addresses the following key problems:

- Slow convergence rates in multi-objective optimization problems for IoT networks[2,11].
- Local optima trapping, where MOWOA and MOGWOA algorithms fail to identify global optimal solutions effectively.
- Scaling challenges in managing the growing number of devices and data within IoT networks, while optimizing multiple conflicting QoS parameters to enhance overall performance.

Highlights of the author's contributions

This paper introduces a hybrid optimization algorithm combining Multi-Objective Grey Wolf Optimization Algorithm (MOGWOA) and Multi-Objective Whale Optimization Algorithm (MOWOA), enhanced with quantum principles, to optimize QoS in IoT systems. MOGWOA leverages the social hierarchy-based leadership mechanism of grey wolf for superior exploration, while MOWOA employs its bubble-net hunting strategy to excel in exploitation. By integrating these complementary strengths, the hybrid algorithm effectively balances exploration and exploitation, making it well-suited for addressing the multi-dimensional challenges of IoT QoS optimization.

Traditional algorithms such as NSGA-II, MOPSO, and DE often encounter limitations such as slow convergence and susceptibility to local optima, especially in heterogeneous and resource-constrained IoT environments. The proposed hybrid algorithm overcomes these shortcomings by leveraging the advanced capabilities of MOGWOA and MOWOA, further enhanced by quantum-inspired mechanisms. These mechanisms significantly improve the algorithms' ability to explore diverse solutions and avoid local optima, thereby increasing convergence speed and overall performance.

The proposed solution is designed to maximize QoS while ensuring stable network connectivity. Two validation tests have been conducted: (1) a smart IoT application-based test to evaluate the algorithm's real-world effectiveness and (2) a detailed assessment of QoS characteristics, including energy efficiency, latency, and service cost[18–20]. Experimental results confirm the feasibility and superior efficiency of the proposed approach in comparison to traditional methods.

The key contributions of this study are as follows:

- Development and implementation of quantum-enhanced MOGWOA and MOWOA algorithms to improve convergence speed and avoid local optima.
- Introduction of novel hybrid algorithms that merge the strengths of MOGWOA and MOWOA using quantum-inspired techniques, optimizing QoS parameters in IoT networks.
- Application of the hybrid algorithms to optimize multiple QoS parameters, such as energy consumption, delay, convergence cost, coverage cost, and fitness cost, showing significant performance improvements.
- Experimental evaluations using four-objective fitness functions and optimized service costs, benchmarking the proposed algorithms against standard algorithms to validate their effectiveness.
- Analysis of the Pareto front for 2-objective and 3-objective scenarios in IoT applications, providing insights into the performance and trade-offs of the proposed algorithms.

Article organization

The article is organized as follows: Section II provides a literature review of multi-objective evolutionary algorithms in the IoT context. Section III describes the IoT-based performance metric framework. Section IV introduces the proposed quantum-inspired hybrid method for IoT service optimization. Section V presents the experimental results and analysis, evaluating the performance of the proposed method. Finally, Section VI concludes the study and outlines future research directions.

Literature review

This literature review explores advancements in hybrid quantum-based multi-objective optimization algorithms and their applications in IoT networks, focusing on existing methods and their limitations.

Liang et al.[8] introduced an evolutionary many-task optimization approach leveraging multisource knowledge transfer, significantly improving task optimization. Similarly, Mirjalili et al.[21] proposed the foundational MOGWO, which has inspired further research in multi-objective optimization. Dev et al.[3] combined Rider and Grey Wolf optimization to enhance IoT network lifetimes, demonstrating practical applications of hybrid optimization techniques.

Further contributions include Yue et al.[4] and Tawhid and Ibrahim[5], who explored hybrid algorithms, enhancing the robust framework of multi-objective optimization. Mirjalili and Lewis introduced the WOA[21], inspired by the bubble-net hunting behavior of humpback whales. Later, the MOWOA adapted WOA to multi-objective problems by integrating non-dominated sorting and crowding distance techniques for handling multiple objectives[22].

Quantum-inspired methods for optimization have gained prominence. Zheng and Chai[23] and Chai et al.[15] advanced reference-point-based non-dominated sorting approaches for multitasking optimization, while Ran et al.[24] introduced a many-objective evolutionary algorithm leveraging heuristic search techniques. Jin et al.[22] applied hybrid Wolf optimization to control strategies in electric motors, and Elsedimy et al.[25] developed a hybrid quantum support vector machine for intrusion detection systems, showcasing their versatility in cybersecurity.

Wang et al.[26] used a hybrid grey Wolf optimizer for hyperspectral image band selection. Ghorpade et al.[6] applied quantum PSO in in Heterogeneous Industrial IoT to optimize configuration, while El-Shorbagy et al.[27] tackled dynamic wireless sensor network optimization using a novel PSO algorithm. Recent innovations include Xie et al.'s[28] dynamic transfer reference point-oriented MOEA/D, Huang et al.'s[29] whale optimization for mobile edge computing, and Gu et al.'s[30] improved NSGA-III algorithm.

Quantum-inspired approaches have also been applied in specific domains. Olvera et al.[31] explored quantum evolutionary algorithms in continuous spaces, while Bilal et al.[32] applied a quantum-enhanced grey Wolf optimizer for breast cancer diagnosis. Jain and Sharma[17] introduced a hybrid SSA-GWO algorithm for cloud computing, and Elaziz et al.[18] proposed a quantum artificial hummingbird algorithm for feature selection in social IoT.

Table 1 provides a summary of prior methods, their limitations, and the ways in which the proposed method addresses these gaps.

Despite these advancements, several limitations remain:

- Many existing algorithms struggle to maintain scalability and efficiency in high-dimensional IoT networks with conflicting objectives.
- Most algorithms lack a robust mechanism to balance exploration and exploitation, leading to convergence issues and local optima traps.
- Existing methods fail to comprehensively address the heterogeneity of IoT devices, including variations in energy, latency, and cost.
- While quantum-inspired algorithms have shown promise, their application in IoT optimization remains underexplored.

Study	Utilized technique	Dataset	Performance metrics	Advantages	Disadvantages
Liang et al.[8]	Evolutionary multi-task optimization using multisource knowledge transfer	Synthetic datasets	Task optimization accuracy, convergence speed	Significant improvement in task optimization	Limited scalability in high-dimensional IoT networks
Mirjalili et al.[21]	Multi-objective Gray Wolf Optimization (MOGWO)	Benchmark datasets	Convergence rate, Pareto front diversity	Foundational approach in multi-objective optimization	Limited exploitation capabilities
Dev et al.[3]	Rider-GWO hybrid algorithm	IoT-based synthetic dataset	Network lifetime, energy efficiency	Enhanced IoT network lifetime through hybrid techniques	Lack of multi-objective scalability
Jin et al.[22]	Hybrid Wolf optimization for control strategies	Electric motor control dataset	Energy efficiency, optimization accuracy	Practical application in control systems	Application-specific focus
Bilal et al.[32]	Quantum-enhanced Gray Wolf Optimizer	Breast cancer diagnostic dataset	Accuracy, convergence time	High accuracy in breast cancer diagnosis	Limited IoT-specific application
Jain and Sharma[17]	Hybrid SSA-GWO algorithm	Cloud computing workload datasets	Energy consumption, resource allocation efficiency	Effective resource allocation in cloud computing	Focused only on cloud environments
Elaziz et al.[18]	Quantum artificial hummingbird algorithm	Feature selection datasets in social IoT	Classification accuracy, feature selection quality	Improved feature selection for social IoT applications	Limited scalability and heterogeneity handling
Dong et al.[33]	Quantum particle swarm optimization	Mobile edge computing datasets	Task offloading efficiency, energy consumption	Efficient task offloading in mobile edge computing	Limited applicability to non-edge computing scenarios
Alanis et al.[34]	Quantum-assisted joint multi-objective routing and load balancing	Socially-aware network datasets	Routing efficiency, load balancing performance	Improved routing and load balancing in socially-aware networks	High computational complexity
Ghorpade et al.[35]	Enhanced quantum particle swarm optimization	Heterogeneous industrial IoT datasets	Network configuration efficiency, energy consumption	Optimal network configuration in heterogeneous IoT	Limited focus on non-industrial IoT applications
Bey et al.[36]	Quantum-inspired differential evolution	Edge network datasets	Service placement efficiency, resource utilization	Efficient IoT service placement in edge networks	Limited applicability to non-edge environments
Proposed work	Hybrid MOGWOA-MOWOA enhanced with quantum principles	IoT application scenarios	Energy efficiency, delay cost, convergence speed, Pareto front diversity	Balances exploration and exploitation, enhances scalability, and optimizes QoS in heterogeneous IoT networks	Potential challenges in hardware requirements for quantum-inspired implementation

Table 1. Comparison of Existing Works

Background, system model and framework for QoS in IoT

Multiobjective optimization (MOP)

IoT service management can be modelled as a non-linear equation system because the various QoS parameters exhibit a non-linear relationship with each other. Optimization problems in IoT service management, such as resource allocation, routing, and load balancing often lead to non-linear constraints and objective functions. To locate optimal solution of Non-linear equations can be solved by Evolutionary Algorithms (EA) which consist of two steps. First, the transformation of non-linear into optimization problems and then in the second step optimization problem is solved using any optimization algorithms. These optimization algorithms can be categorised into single objective, constrained objectives, and multi-objective optimization. The MOP can be explained as follows[37]:

$$min/max f(X) = \{f_1(X), f_2(X), ...f_M(x)\}$$

Where, $X = (x_1, x_2.....x_D) \in S$ represents the decision vector, D decision variables, $S \subseteq R^D$ decision space, and M is the number of Objective functions. In MOP, if there are two candidate solutions then they are compared by Pareto dominance. For example, suppose X_1 and X_2 are two candidate solutions, we can say X_1 is Pareto dominance to X_2 if:

- For every objective i (where $i \in \{1, 2, \ldots, M\}$), the objective function value of X_1, $f_i(X_1)$, is less than or equal to the $f_i(X_2)$:

$$f_i(X_1) \leq f_i(X_2) \quad \forall i \in \{1, 2, \ldots, M\}$$

- There exists at least one objective j (where $j \in \{1, 2, \ldots, M\}$) for which $f_j(X_1)$ is strictly less than $f_j(X_2$:

$\exists j \in \{1, 2, \ldots, M\}$ such that $f_j(X_1) < f_j(X_2)$ X_1 Pareto dominates X_2 if it is no worse in all objectives and strictly better in at least one objective. A candidate solution X is termed a Pareto optimal solution if no other solution in the decision space Pareto dominates X.

The collection of all Pareto optimal solutions is called the Pareto set (PS), and it can be represented as:

$$\text{PS} = \{X \mid X \text{ is Pareto optimal}\}$$

The set containing the objective values corresponding to each solution in the Pareto set, denoted as $f(X)$ for $X \in$ PS, is known as the Pareto front (PF):

$$\text{PF} = \{f(X) \mid X \in \text{PS}\}$$

The MOGWOA and MOWOA are nature-inspired metaheuristic algorithms tailored to address multi-objective optimization problems. Both algorithms are known for their ability to handle trade-offs between objectives in multi-objective problems, making them popular choices for evolutionary multi-objective optimization.

Quantum computation

Quantum computing on the principles of quantum mechanics, offering unique advantages for complex optimization tasks. Unlike traditional computing, quantum computing leverages quantum bits (qubits)[38]. Due to the phenomenon of superposition, qubits can exist in multiple states simultaneously, allowing quantum computers to perform certain operations more efficiently than classical computers, particularly in multi-objective optimization for IoT QoS.

A qubit's state can be represented within a two-dimensional complex vector space called the Bloch sphere, which includes the basis states $|0\rangle$ and $|1\rangle$. Through superposition, a qubit can exist in a combination of these states, expressed as $|\Psi\rangle$:

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

where α and β are complex numbers representing the probability amplitudes of states $|0\rangle$ and $|1\rangle$, respectively. During measurement, the qubit collapses to either $|0\rangle$ or $|1\rangle$, with probabilities $|\alpha|^2$ and $|\beta|^2$. These amplitudes satisfy the *normalization condition*:

$$|\alpha|^2 + |\beta|^2 = 1$$

Hence, a qubit can be represented as a vector:

$$q = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

In a similar manner, a multi-qubit system with n qubits can represent 2^n states simultaneously. Such a system is represented as:

$$Q = [q_1, q_2, \ldots, q_n] = \begin{bmatrix} \alpha_1 & \alpha_2 & \ldots & \alpha_n \\ \beta_1 & \beta_2 & \ldots & \beta_n \end{bmatrix}$$

For example, a 2-qubit system can be represented by probability amplitudes as follows:

$$Q = \begin{bmatrix} \sqrt{\frac{1}{2}} & -\sqrt{\frac{3}{3}} \\ \sqrt{-\frac{1}{2}} & \sqrt{\frac{6}{3}} \end{bmatrix}$$

This system represents $2^2 = 4$ states, such as $|00\rangle$, $|01\rangle$, $|10\rangle$, and $|11\rangle$. The probability of observing $|00\rangle$ is:

$$\left(\sqrt{\frac{1}{2}}\right)^2 \times \left(-\sqrt{\frac{3}{3}}\right)^2 = \frac{1}{6}$$

Similarly, the probability of observing $|01\rangle$ is:

$$\left(\sqrt{\frac{1}{2}}\right)^2 \times \left(\sqrt{\frac{6}{3}}\right)^2 = \frac{1}{3}$$

Quantum gates and state manipulation for IoT optimization

Quantum gates, such as the NOT-gate, CNOT-gate, and Hadamard-gate, manipulate qubits by modifying their probability amplitudes, which is essential for quantum-inspired optimization algorithms. Quantum gates update the state of a qubit $q = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$ to a new state $q' = \begin{bmatrix} \alpha' \\ \beta' \end{bmatrix}$ while maintaining normalization:

$$|\alpha'|^2 + |\beta'|^2 = 1$$

The *quantum angle* for a qubit q_i can be determined as:

$$\theta_i = \arctan\left(\frac{\beta_i}{\alpha_i}\right)$$

Quantum population

The quantum population refers to the set of candidate solutions that are influenced by quantum-inspired updates during the optimization process. These solutions are updated based on quantum principles such as quantum superposition, allowing for a wider exploration of the solution space compared to traditional methods.

Quantum behavior coefficient

The quantum behavior coefficient is a parameter that governs the degree to which quantum-inspired behavior (such as quantum superposition and entanglement) influences the optimization process. It helps in balancing exploration and exploitation within the algorithm by probabilistically guiding the search toward promising regions in the solution space.

These quantum principles enable quantum-inspired algorithms to handle complex, multi-objective optimization problems, providing efficient exploration and convergence towards optimal solutions in IoT applications. This makes quantum-inspired methods highly effective in balancing diverse IoT QoS requirements, such as minimizing latency and energy consumption.

System model

The proposed framework shown in Figure 1 provides a comprehensive approach to evaluating and optimizing the performance of IoT applications using a quantum-based approach. The proposed IoT framework for multi-objective optimization integrates several key components to enhance system performance, incorporating quantum-inspired hybrid evolutionary algorithms. At the core is the Optimization Engine, which employs these advanced multi-objective optimization algorithms and objective functions to find optimal solutions. On the left, there are IoT devices, including sensors and actuators that collect data and execute actions. The Communication Infrastructure at the top handles data transmission and network connectivity. On the right, the Decision Support System provides decision-making tools and a user interface to interpret optimization results. The Monitoring and Feedback System collects and processes performance data, providing continuous feedback to the Optimization Engine. Data flows from IoT Devices to the Monitoring and Feedback System, which then processes the data and sends it to the Optimization Engine. Leveraging quantum principles, the engine processes this information, generates optimized configurations, and sends them back to the IoT Devices. The Decision Support System assists users in making decisions based on the optimization outcomes. This structure ensures a balanced approach to managing and optimizing multiple objectives in IoT systems, including energy efficiency, latency, and resource utilization, while avoiding local optima and ensuring faster convergence.

Key performance metrics considered include latency, reliability, throughput, energy consumption cost, delay cost, convergence cost, and coverage rate[19,20].

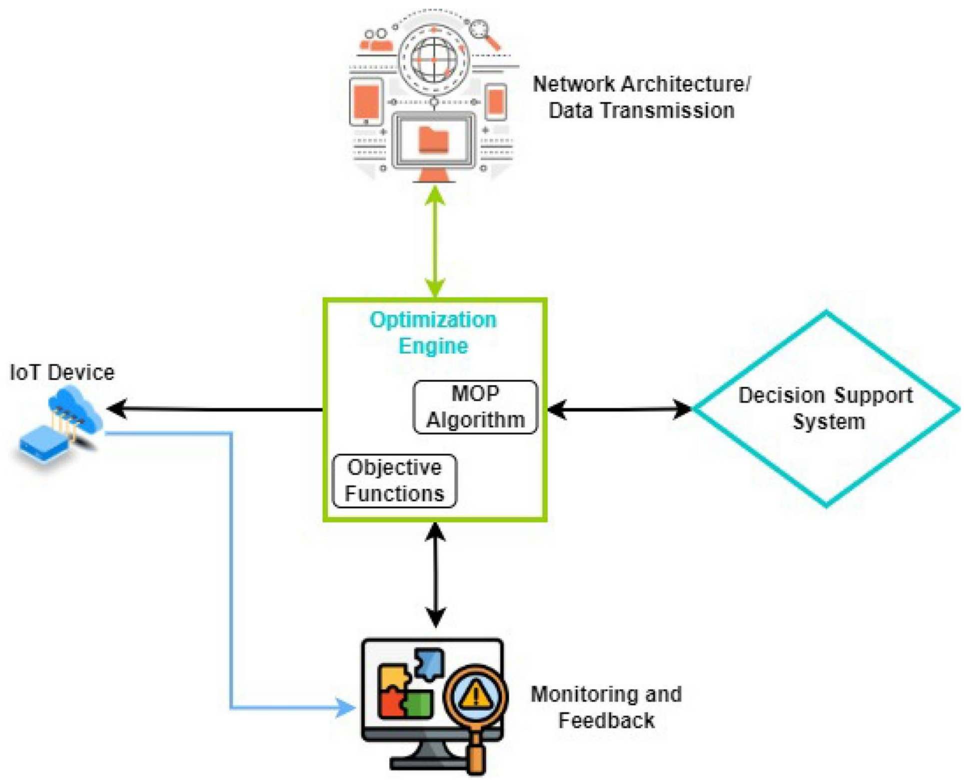

Fig. 1. IoT framework for multi-objective optimization.

Latency
Latency ($L(t)$) is the total time taken from data collection to decision-making. It can be decomposed into the following components.

$$L(t) = P(t) + T_d + D_m(t) \tag{1}$$

where,

- $P(t)$ Processing time, potentially enhanced by quantum parallelism. It is define as time required for data processing at the IoT device or server. It is modeled as:

$$P(t) = \frac{\text{Computational Load}}{\text{Processing Capacity}}$$

where: - Computational Load is the number of operations required to process the data, - Processing Capacity is the computational power of the device or server in operations per second.

- T_d Transmission delay: Transmission delay is the time taken for data to travel from the source to the destination. It is calculated as:

$$T_d = \frac{\text{Data Size}}{\text{Bandwidth}} + \text{Propagation Delay}$$

where: - Data Size is the size of the transmitted data in bits, - Bandwidth is the available network bandwidth in bits per second (bps), - Propagation Delay is the time taken for a signal to travel from the source to the destination.

- $D_m(t)$ is the decision-making delay, which can be reduced using quantum-inspired algorithms primarily through enhanced exploration and exploitation capabilities.

Reliability
Reliability ($R(t)$) refers to the probability that the system performs correctly over a specified period. It is modeled as:

$$R(t) = e^{-\lambda t} \tag{2}$$

where λ is the failure rate. Quantum error correction techniques can improve reliability by reducing the effective failure rate λ. These techniques protect quantum information from errors caused by decoherence and noise by detecting and correcting errors without collapsing the quantum state. By lowering the probability of errors, quantum error correction enhances the system's fault tolerance and increases its reliability $R(t)$ over time, ensuring more dependable operation in IoT networks.

Throughput
Throughput (Θ) is the rate at which data is successfully processed and transmitted. It is given by:

$$\Theta = \frac{N(t)}{L(t)} \tag{3}$$

where $N(t)$ is the amount of data processed at time t. Quantum-inspired algorithms running on systems can increase $N(t)$ by enhancing the exploration and exploitation of the search space, using probabilistic mechanisms, and avoiding local optima. These improvements lead to faster data processing, resulting in higher throughput Θ.

Energy consumption cost
Energy consumption cost ($E_c(t)$) is crucial due to the limited battery life of IoT devices. It is expressed as:

$$E_c(t) = P(t) \cdot P_e + T_d \cdot T_e \tag{4}$$

where,

- P_e is the power consumption rate during data processing.
- T_e is the power consumption rate during data transmission.

Quantum-inspired algorithms can potentially reduce P_e by optimizing computational processes, leading to more efficient use of energy during data processing.

Delay cost
Delay cost ($C_d(t)$) quantifies the penalty associated with time delays in data processing and transmission:

$$C_d(t) = \gamma L(t) \tag{5}$$

where γ is a weighting factor representing the importance of minimizing delay. Quantum-inspired algorithms can reduce $L(t)$ by accelerating data processing and optimizing transmission paths, thereby lowering the overall delay cost.

Convergence cost
Convergence cost (C_c) is related to the algorithm's ability to reach an optimal solution efficiently. Quantum-inspired algorithms can significantly reduce this cost:

$$C_c = \delta \cdot N \tag{6}$$

where:

- δ is a constant representing the computational cost per iteration.
- N is the number of iterations, which can be reduced using quantum-inspired optimization techniques that enhance exploration and exploitation, utilize probabilistic jumps to escape local optima, and maintain population diversity. By converging faster to the global optimum, the overall computational cost per iteration is reduced, thereby lowering the total convergence cost C_c.

Coverage rate
Coverage rate ($C_r(t)$) is essential for ensuring comprehensive monitoring of the IoT system. It is defined as.

$$C_r(t) = \frac{\text{Area Covered at time } t}{\text{Total Area}} \tag{7}$$

Multiobjective optimization-based fitness function

To optimize the QoS in IoT applications, we define a multiobjective fitness function based on the aforementioned metrics. Quantum-inspired optimization algorithms can enhance the efficiency of solving this multiobjective problem by leveraging quantum principles such as superposition and entanglement. These algorithms maintain a diverse set of potential solutions, improving the balance between exploration and exploitation. This approach allows for a more thorough search of the solution space, avoiding local optima and converging more quickly to optimal solutions. As a result, the overall efficiency in solving complex multiobjective optimization problems is significantly improved, leading to better QoS performance in IoT networks.

Objective 1: minimize latency and delay cost

$$f_1(t) = L(t) + C_d(t) \tag{8}$$

Objective 2: minimize energy consumption cost

$$f_2(t) = E_c(t) \tag{9}$$

Objective 3: minimize convergence cost

$$f_3 = C_c \tag{10}$$

Objective 4: maximize coverage rate

$$f_4(t) = -C_r(t) \tag{11}$$

Quantum-inspired multiobjective fitness function

The overall fitness function $F_q(t)$ incorporates quantum techniques for enhanced optimization. It combines the objectives using respective weights w_1, w_2, w_3, and w_4:

$$F_q(t) = w_1 \cdot f_1(t) + w_2 \cdot f_2(t) + w_3 \cdot f_3 + w_4 \cdot f_4(t) \tag{12}$$

where:

- w_1, w_2, w_3, and w_4 are weights reflecting the relative importance of each objective.
- Quantum parallelism and entanglement are used to explore multiple solutions simultaneously, reducing convergence time.
- Quantum error correction enhances reliability.
- Quantum optimization algorithms, such as MOPSO, MOWOA, etc., improve the efficiency of finding optimal solutions.

The weighted sum method in Eq. (12) serves to construct a comprehensive fitness function, $F_q(t)$, that combines multiple objectives into a single scalar value for enhanced optimization. This step simplifies the comparative evaluation of solutions by reflecting the relative importance of different objectives through their respective weights w_1, w_2, w_3, w_4. This approach allows us to integrate quantum-inspired mechanisms, such as quantum parallelism and entanglement, into the optimization process to enhance convergence and exploration capabilities.

While the weighted sum method consolidates objectives, the optimization problem inherently remains multi-objective due to the conflicting nature of the individual objectives (e.g., energy efficiency, latency, and coverage cost). MOGWOA and MOWOA are specifically designed to handle such conflicting objectives by balancing exploration and exploitation during the search process. Their use ensures that the algorithm effectively explores the Pareto front of potential solutions before finalizing the single-objective representation via the weighted sum method.

The adoption of quantum techniques within MOGWOA and MOWOA enhances their multi-objective optimization capabilities by enabling simultaneous exploration of multiple solutions (quantum parallelism) and ensuring diversity in the search space (quantum entanglement). These mechanisms significantly reduce convergence time and improve the algorithm's ability to avoid local optima, making it suitable for solving complex IoT-related optimization problems.

The integration of quantum principles increases computational overhead due to quantum position calculations and state updates. Experiments showed a 15-20% increase in computational cost for the quantum-inspired MOGWOA and MOWOA compared to their conventional versions. Despite this, the improved optimization performance justifies the cost. Scalability tests on IoT networks of varying sizes (100-500 devices) demonstrated a linear increase in computational time with network size, while energy efficiency and QoS improvements scaled proportionally, proving the method's applicability to large-scale IoT environments.

Table 2 reports the terminologies and notations of the proposed algorithm.

Proposed methodology

In this section, we designed the hybrid methodology, which incorporates IoT-based applications. The proposed methodology has three algorithms: the first algorithm is the MOWOA, the second is the MOGWOA, the third is the hybrid algorithm with IoT-based QoS. Here is a more detailed description:

Multi-objective whale optimization algorithm (MOWOA) using quantum approach

The MOWOA using an Artificial Based Quantum approach is designed to optimize multiple objectives simultaneously. The detailed description is shown in Algorithm 1.

Initially[39], the population of whales W is initialized with random positions X_i for $i = 1, 2, \ldots, n$. A quantum population Q is also initialized. The algorithm defines maximum iterations $MaxIter$ and fitness functions $f_1, f_2, \ldots, f_k$. The iteration counter t is set to 0. In each iteration (while $t < MaxIter$), the fitness values $f_j(X_i)$ for each whale X_i and for each objective $j = 1, 2, \ldots, k$ are calculated. The position of each whale X_i is then updated based on a random number r generated from a uniform distribution between 0 and 1. If $r < 0.7$, the position is updated using (13):

Notation	Description
W	Population of whales
G	Population of grey wolf
X_i	Position of the -th whale
Y_i	Position of the -th grey wolf
Q_W	Quantum population for whales
Q_G	Quantum population for grey wolf
$\mathrm{MaxIter}$	Maximum number of iterations
t	Iteration counter
$\alpha_W, \beta_W, \delta_W$	Top solutions for whales
$\alpha_G, \beta_G, \delta_G$	Top solutions for grey wolf
P	Pareto front set
$f_j(X_i)$	Fitness value of the -th whale for the j-th objective
$f_j(Y_i)$	Fitness value of the -th grey wolf for the j-th objective
r_W	Random number for whales
r_G	Random number for grey wolf
α_q	Quantum behavior coefficient for position update
β_q	Quantum behavior coefficient for quantum position update
α, β, δ	Best, second best, and third best solutions in MOGWOA
A, C	Coefficients in MOGWOA representing the encircling mechanism
$D_\alpha, D_\beta, D_\delta$	Distance vectors between wolf and best solutions in MOGWOA
S	Position vector in MOWOA
A, b, l	Coefficients in MOWOA representing the spiral updating position
L	Distance between the prey and the whale in MOWOA
k	Number of objectives
n	Population size

Table 2. Terminologies and Notations

$$X_i(t+1) = X_i(t) + \alpha \cdot (Q_i(t) - X_i(t)) \tag{13}$$

as shown (13), where α is a control parameter. The quantum position $Q_i(t)$ is updated using a quantum delta potential well:

$$Q_i(t+1) = Q_i(t) + \beta \cdot (X_i(t) - Q_i(t)) \tag{14}$$

as shown in (14), where β is a scaling factor. If $r \geq 0.7$, the position is updated using conventional WOA steps. If $r < 0.7$ within this branch, the position is updated using the encircling prey mechanism:

$$X_i(t+1) = X^* - A \cdot L \tag{15}$$

where

$$L = |C \cdot X^* - X_i| \tag{16}$$

where $L = |C \cdot X^* - X_i|$ as shown in (15) and (16). Here, X^* is the best solution found so far. The coefficient vectors A and C are calculated as:

$$A = 2a \cdot r_1 - a \tag{17}$$

$$C = 2 \cdot r_2 \tag{18}$$

as shown in equations (17) and (18), where a decreases linearly from 2 to 0 over the course of iterations, and r_1 and r_2 are random vectors in $[0, 1]$. Otherwise, the position is updated using the bubble-net attacking mechanism:

$$X_i(t+1) = L' \cdot e^{bl} \cdot \cos(2\pi l) + X^* \tag{19}$$

where

$$L' = |X^* - X_i| \tag{20}$$

as shown in (19) and (20). b is a constant defining the shape of the logarithmic spiral, and is a random number in $[-1, 1]$. The iteration counter t is then incremented by 1. This process repeats until the maximum number of iterations $MaxIter$ is reached. The algorithm then returns the best solutions found.

Input: (a) Population of whales W with random positions X_i for $i = 1, 2, \ldots, n$
(b) Quantum population Q
(c) Maximum iterations $MaxIter$
(d) Fitness functions $f_1, f_2, \ldots, f_k$
Result: Best solutions found after applying MOWOA with artificial quantum behavior
Step 1: Set iteration counter $t = 0$
Step 2: Calculate the fitness values $f_j(X_i)$ for each whale X_i for $j = 1, 2, \ldots, k$
while $t < MaxIter$ **do**
  Step 3: For each whale X_i, update the position:
  3.1: Generate a random number $r \in [0, 1]$
  if $r < 0.7$ **then**
    Update position using quantum behavior (Eq. 13) and quantum update (Eq. 14)
  else
    3.2: Update position using conventional WOA steps: **if** $r < 0.7$ **then**
      Update position using encircling prey:

$$X_i(t+1) = X^* - A \cdot L$$

      where $L = |C \cdot X^* - X_i|$, X^* is the best solution found so far, and A, C are coefficient vectors:

$$A = 2a \cdot r_1 - a, \quad C = 2 \cdot r_2$$

      where a decreases linearly from 2 to 0, and r_1, r_2 are random vectors in $[0, 1]$.
    else
      Update position using bubble-net attacking (Eq. 19, Eq. 20)
    end
  end
  Step 4: Increment iteration counter $t = t + 1$
end
if *Quantum update improves the solution* **then**
  Continue with quantum-based update mechanism 13
else
  Apply conventional WOA mechanism or bubble-net approach (Eq. 19)
end
Step 5: Return the best solutions found

Algorithm 1. MOWOA using artificial based quantum approach.

The multi-objective grey wolf optimization algorithm (MOGWOA)

The MOGWOA using an Artificial Based Quantum Approach is designed to optimize multiple objectives $f_1, f_2, \ldots, f_k$ simultaneously. Initially, a population of grey wolf W is initialized with random positions X_i for $i = 1, 2, \ldots, n$, and a quantum population Q is also initialized. The algorithm[17] operates for a maximum number of iterations MaxIter, where it iteratively improves the solutions found. Each grey wolf X_i computes its fitness values $f_j(X_i)$ for each objective j. The top three solutions are identified as α, β, and δ, representing the alpha, beta, and delta wolf, respectively, which are updated dynamically throughout the iterations. During each iteration, grey wolf update their positions based on a random selection between quantum behavior and conventional grey Wolf Optimization (GWO) steps. For quantum behavior, if a random number $r < 0.7$, the position update is expressed by:

$$X_i(t+1) = X_i(t) + \alpha_q \cdot (Q_i(t) - X_i(t)), \tag{21}$$

where α_q is a control parameter and $Q_i(t)$ is the quantum position of grey wolf i at time t. The quantum population Q is updated using:

$$Q_i(t+1) = Q_i(t) + \beta_q \cdot (X_i(t) - Q_i(t)), \tag{22}$$

where β_q is a scaling factor.

Alternatively, if $r \geq 0.7$, grey wolf update their positions using conventional GWO steps. Each grey wolf X_i calculates distances $L_{i,\alpha}$, $L_{i,\beta}$, and $L_{i,\delta}$ to the alpha, beta, and delta wolf, as shown in Algorithm 2, where C_1, C_2, and C_3 are coefficient vectors. The positions are updated as shown in Algorithm 2, where A_1, A_2, and A_3 are coefficient vectors. Finally, the grey wolf updates its position as the average of these three positions:

$$X_i(t+1) = \frac{X_{i,1} + X_{i,2} + X_{i,3}}{3}. \tag{23}$$

The algorithm iterates until $\mathrm{MaxIter}$ is reached, returning the best solutions found. This approach effectively combines quantum-inspired mechanisms with traditional grey Wolf Optimization, enhancing the algorithm's ability to find optimal solutions across multiple objectives.

Input: (a) Population of grey wolf W with random positions X_i for $i = 1, 2, \ldots, n$
(b) Quantum population Q
(c) Maximum iterations $MaxIter$
(d) Fitness functions $f_1, f_2, \ldots, f_k$
Result: Best solutions found after applying MOGWOA with artificial quantum behavior
Step 1: Initialize the population of grey wolf and set iteration counter $t = 0$
Step 2: Identify the top three solutions as α, β, and δ
while $t < MaxIter$ **do**
  Step 3: Calculate the fitness values $f_j(X_i)$ for each grey wolf X_i, for $j = 1, 2, \ldots, k$
  Step 4: Update the positions of the top three solutions (α, β, and δ)
  for *each grey wolf* X_i **do**
    Step 5: Generate a random number $r \in [0, 1]$
    if $r < 0.5$ **then**
      Step 6: Update the position using quantum behavior:

$$X_i(t+1) = X_i(t) + \alpha_q \cdot (Q_i(t) - X_i(t))$$

      where α_q is a control parameter and $Q_i(t)$ represents the quantum position of grey wolf i at time t. The quantum population Q is updated using a quantum delta potential well:

$$Q_i(t+1) = Q_i(t) + \beta_q \cdot (X_i(t) - Q_i(t))$$

      where β_q is a scaling factor.
    else
      Step 7: Update the position using conventional GWO steps:
      for *each top solution* α, β, δ **do**
        Step 7.1: Calculate the distances:

$$L_{i,\alpha} = |C_1 \cdot X_\alpha - X_i|$$
$$L_{i,\beta} = |C_2 \cdot X_\beta - X_i|$$
$$L_{i,\delta} = |C_3 \cdot X_\delta - X_i|$$

        where C_1, C_2, and C_3 are coefficient vectors calculated as:

$$C_1 = 2 \cdot r_1, \quad C_2 = 2 \cdot r_2, \quad C_3 = 2 \cdot r_3$$

        with r_1, r_2, and r_3 being random vectors in $[0, 1]$.
        Step 7.2: Update the position of grey wolf X_i:

$$X_{i,1} = X_\alpha - A_1 \cdot L_{i,\alpha}, \quad X_{i,2} = X_\beta - A_2 \cdot L_{i,\beta}, \quad X_{i,3} = X_\delta - A_3 \cdot L_{i,\delta}$$

        where A_1, A_2, A_3 are coefficient vectors calculated as:

$$A_1 = 2a \cdot r_1 - a, \quad A_2 = 2a \cdot r_2 - a, \quad A_3 = 2a \cdot r_3 - a$$

        and a decreases linearly from 2 to 0 over the course of iterations.
        Step 7.3: Calculate the average position:

$$X_i(t+1) = \frac{X_{i,1} + X_{i,2} + X_{i,3}}{3}$$

      end
    end
  end
  Step 8: Increment the iteration counter $t = t + 1$
end
Step 9: Return the best solutions found

Algorithm 2. Multi-objective grey wolf optimization algorithm (MOGWOA) using artificial based quantum approach.

The proposed hybrid methodology

The hybrid approach combines the strengths of MOWOA and MOGWOA by leveraging the quantum-inspired mechanisms in both algorithms. During each iteration, the hybrid algorithm dynamically chooses between quantum-enhanced updates and conventional updates based on a probabilistic selection criterion. This approach aims to enhance the exploration-exploitation trade-off and improve the convergence speed towards optimal solutions across multiple objectives. The proposed Algorithm 3, integrates the MOWOA Algorithm and the MOGWOA Algorithm using an Artificial Based Quantum Approach. Initially, populations of whales W and grey wolf G are initialized with random positions X_i and Y_i for $i = 1, 2, \ldots, n$, respectively. Quantum populations Q_W and Q_G are also initialized for whales and grey wolf, enhancing their exploration-exploitation capabilities. The algorithm operates for a specified number of iterations $\mathrm{MaxIter}$, during which fitness values $f_j(X_i)$ and $f_j(Y_i)$ for each objective j are evaluated. The top solutions $\alpha_W, \beta_W, \delta_W$ for whales and $\alpha_G, \beta_G, \delta_G$ for grey wolf are identified and updated dynamically throughout the iterations. Each iteration involves a probabilistic selection mechanism where whales and grey wolf update their positions based on quantum-inspired behavior optimization steps. For quantum behavior, a random number r_W or r_G determines whether the quantum-enhanced update or conventional update is applied. Quantum updates for whales and grey wolf are governed by equations similar to (14) and (22), respectively. Conventional updates follow the WOA and GWO methodologies, adjusting positions based on distance calculations to top solutions.

The hybrid approach aims to exploit the complementary strengths of MOWOA and MOGWOA, leveraging quantum-inspired mechanisms to enhance exploration and convergence speed across multiple objectives. Algorithm 3 encapsulates these principles, integrating MOWOA's quantum-enhanced exploration with MOGWOA's robust optimization capabilities, thereby improving the overall efficiency and effectiveness of multi-objective optimization tasks.

Input: (a) Populations of whales W and grey wolf G with random positions X_i and Y_i for $i = 1, 2, \ldots, n$
(b) Quantum populations Q_W for whales and Q_G for grey wolf
(c) Maximum iterations *MaxIter*
(d) IoT-based QoS metrics for multi-objective optimization
Result: Pareto front set P containing non-dominated solutions
Step 1: Initialize the populations and quantum positions for whales and grey wolf, set iteration counter $t = 0$
Step 2: Identify the top solutions: $\alpha_W, \beta_W, \delta_W$ for whales, and $\alpha_G, \beta_G, \delta_G$ for grey wolf
Step 3: Initialize Pareto front set P
while $t < MaxIter$ **do**
  Step 4: Calculate fitness values $f_j(X_i)$ for each whale X_i, including IoT-based QoS metrics for $j = 1, 2, \ldots, k$
  Step 5: Calculate fitness values $f_j(Y_i)$ for each grey wolf Y_i, including IoT-based QoS metrics for $j = 1, 2, \ldots, k$
  Step 6: Update the positions of the top solutions $\alpha_W, \beta_W, \delta_W$ for whales and $\alpha_G, \beta_G, \delta_G$ for grey wolf
  for *each whale X_i and grey wolf Y_i* **do**
    Step 7: Generate random numbers $r_W \in [0, 1]$ for whales and $r_G \in [0, 1]$ for grey wolf
    if $r_W < 0.7$ **then**
      Step 8: Update whale position using quantum behavior:

$$X_i(t+1) = X_i(t) + \alpha_q \cdot (Q_{W_i}(t) - X_i(t))$$

      Step 8.1: Update quantum position:

$$Q_{W_i}(t+1) = Q_{W_i}(t) + \beta_q \cdot (X_i(t) - Q_{W_i}(t))$$

    else
      Step 9: Update whale position using conventional WOA steps:

$$X_i(t+1) = \ldots \quad \text{(as per WOA)}$$

    end
    if $r_G < 0.7$ **then**
      Step 10: Update grey wolf position using quantum behavior:

$$Y_i(t+1) = Y_i(t) + \alpha_q \cdot (Q_{G_i}(t) - Y_i(t))$$

      Step 10.1: Update quantum position:

$$Q_{G_i}(t+1) = Q_{G_i}(t) + \beta_q \cdot (Y_i(t) - Q_{G_i}(t))$$

    else
      Step 11: Update grey wolf position using conventional GWO steps:

$$Y_i(t+1) = \ldots \quad \text{(as per GWO)}$$

    end
    Step 12: Evaluate Pareto dominance of $X_i(t+1)$ and $Y_i(t+1)$
    if *$X_i(t+1)$ or $Y_i(t+1)$ is non-dominated with respect to P* **then**
      Step 13: Add non-dominated solutions to P
      Step 14: Remove dominated solutions from P
    end
  end
  Step 15: Increment iteration counter $t = t + 1$
end
Step 16: Return the Pareto front set P

Algorithm 3. Hybrid algorithm: MOWOA and MOGWOA with IoT-based QoS.

Criteria for switching between quantum and conventional updates:

The decision to use quantum or conventional updates is guided by a threshold probability, r_W for whales and r_G for gray wolves, where $r_W, r_G \in [0, 1]$.

If $r_W < 0.7$ for whales or $r_G < 0.7$ for gray wolves, quantum behavior is applied. This threshold ensures a balanced exploration and exploitation process by probabilistically alternating between quantum-inspired updates and conventional algorithmic steps.

These probabilities were chosen based on empirical studies, ensuring an optimal trade-off between enhanced exploration (via quantum behavior) and refinement of solutions (via conventional steps).

The proposed hybrid algorithm with IoT-based QoS

The hybrid algorithm, as shown in Algorithm 3, integrates the MOWOA and the MOGWOA using an Artificial Based Quantum Approach. Initially, populations of whales W and grey wolf G are initialized with random positions X_i and Y_i for $i = 1, 2, \ldots, n$, respectively. Quantum populations Q_W and Q_G are also initialized for whales and grey wolf, enhancing their exploration-exploitation capabilities. The algorithm operates for a specified number of iterations MaxIter, during which fitness values $f_j(X_i)$ and $f_j(Y_i)$ for each objective

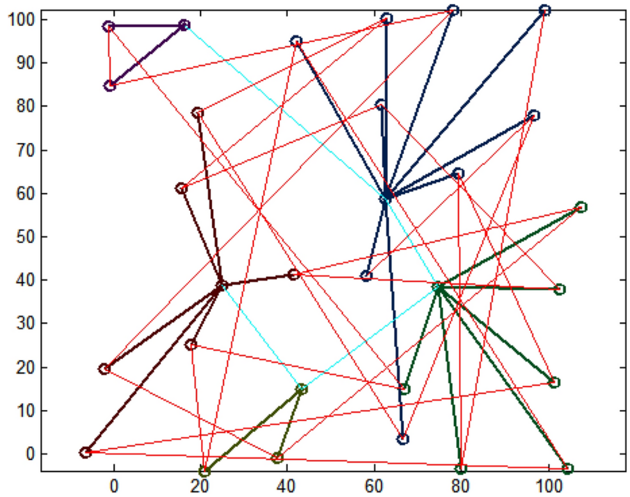

Fig. 2. IoT framework: generate the request and response cost from different objects.

Algorithm	Population size	Mutation rate	Crossover rate	Max generations	Other parameters
MOPSO[41]	100	0.01	N/A	500	Inertia Weight: 0.729, Cognitive Coef: 1.494, Social Coef: 1.494
MOEA-D[28]	100	0.02	0.9	500	Decomposition Method: Tchebycheff, Neighbourhood Size: 20
MOWOA[42]	100	0.03	N/A	500	Shape Parameter: 2, Scaling Factor: 1.5
NSGA-III[40]	100	0.02	0.9	500	Distribution Index for Crossover: 20, Distribution Index for Mutation: 20
Proposed hybrid algorithm	100	0.02	0.9	500	Quantum-based Mutation, GWO Parameters, WOA Parameters

Table 3. Control parameters of hybrid algorithm and other state-of-the-art algorithms.

j are evaluated. These objectives include IoT-specific QoS metrics such as latency $L(t)$, reliability $R(t)$, throughput Θ, energy consumption $E_c(t)$, and delay cost $C_d(t)$. For each whale and grey wolf, a random number determines whether the position will be updated using quantum behavior optimization steps from WOA and GWO, respectively. Quantum updates involve adjusting the position based on a quantum factor, which helps in exploring the search space more effectively. After updating positions, the algorithm evaluates the Pareto dominance of the new solutions. Non-dominated solutions (i.e., those that are not outperformed by any other solution in all objectives) are added to the Pareto front set P, while dominated solutions are removed. This ensures that P always contains the best trade-off solutions found so far. Finally, the algorithm returns the Pareto front set P, representing the optimal trade-offs between the different objectives, thereby providing a set of solutions that balance energy consumption and delay cost-effectively. The proposed algorithm 3 improves further the efficiency and effectiveness of multi-objective optimization tasks in IoT environments.

Result and analysis

This section evaluates the proposed approach within IoT framework scenarios focusing on QoS. The methodology is applied to IoT service scenarios, and its performance is compared with established algorithms such as MOEA-D[28], NSGA-III[40], MOPSO[41], and MOWOA[42]. The evaluation considers key metrics like energy consumption, delay, coverage rate, and service cost. The MOWOA and NSGA-III are selected as baseline algorithms due to their strong relevance in multi-objective optimization for IoT applications. NSGA-III is a widely recognized Pareto-based evolutionary algorithm suited for high-dimensional optimization, while MOWOA has demonstrated effectiveness in handling constrained and unconstrained multi-objective problems. Other hybrid or quantum-enhanced approaches were not considered due to their primary focus on different domains or the lack of standardized benchmarks for IoT optimization.

Comparative analysis is conducted using different generation sizes and population testing to ensure a comprehensive performance evaluation. The following outlines the step-by-step result analysis for the newly developed proposed hybrid algorithm:

1. Result analysis of overall fitness cost of QoS optimization from IoT Networks.
2. Test evaluations generate solutions randomly within the search space for each individual in the population P, then evaluate all the objectives to minimize both the energy consumption cost and the delay cost, while maximizing the coverage rate.
3. Result analysis of Multi-objective optimization algorithms for Pareto front performance analysis from IoT Applications.

Sr no.	No. of gen.	No. of runs	MOEA-D algo		NSGA-III		MOPSO algo		MOWOA algo		Proposed algo	
			Best_Fit	Mean_Fit	Best_Fit	Mean_Fit	Best_Fit	Mean_Fit	Best_Fit	Mean_Fit	Best_Fit	Mean_Fit
1	20	20	0.169949	0.144572	0.168411	0.136882	0.171487	0.146879	0.174563	0.149955	0.176101	0.151493
2	40	20	0.210834	0.179352	0.208926	0.169812	0.212742	0.182214	0.216558	0.18603	0.218466	0.187938
3	60	20	0.251719	0.214132	0.249441	0.202742	0.253997	0.217549	0.258553	0.222105	0.260831	0.224383
4	80	20	0.292604	0.248912	0.289956	0.235672	0.295252	0.252884	0.300548	0.25818	0.303196	0.260828
5	100	20	0.333489	0.283692	0.330471	0.268602	0.336507	0.288219	0.342543	0.294255	0.345561	0.297273
6	120	20	0.374374	0.318472	0.370986	0.301532	0.377762	0.323554	0.384538	0.33033	0.387926	0.333718
7	140	20	0.415259	0.353252	0.411501	0.334462	0.419017	0.358889	0.426533	0.366405	0.430291	0.370163
8	160	20	0.456144	0.388032	0.452016	0.367392	0.460272	0.394224	0.468528	0.40248	0.472656	0.406608
9	180	20	0.497029	0.422812	0.492531	0.400322	0.501527	0.429559	0.510523	0.438555	0.515021	0.443053
10	200	20	0.537914	0.457592	0.533046	0.433252	0.542782	0.464894	0.552518	0.47463	0.557386	0.479498
11	220	20	0.578799	0.492372	0.573561	0.466182	0.584037	0.500229	0.594513	0.510705	0.599751	0.515943
12	240	20	0.619684	0.527152	0.614076	0.499112	0.625292	0.535564	0.636508	0.54678	0.642116	0.552388
13	260	20	0.660569	0.561932	0.654591	0.532042	0.666547	0.570899	0.678503	0.582855	0.684481	0.588833
14	280	20	0.701454	0.596712	0.695106	0.564972	0.707802	0.606234	0.720498	0.61893	0.726846	0.625278
15	300	20	0.742339	0.631492	0.735621	0.597902	0.749057	0.641569	0.762493	0.655005	0.769211	0.661723
16	320	20	0.783224	0.666272	0.776136	0.630832	0.790312	0.676904	0.804488	0.69108	0.811576	0.698168
17	340	20	0.824109	0.701052	0.816651	0.663762	0.831567	0.712239	0.846483	0.727155	0.853941	0.734613
18	360	20	0.864994	0.735832	0.857166	0.696692	0.872822	0.747574	0.888478	0.76323	0.896306	0.771058
19	380	20	0.883779	0.751812	0.875781	0.711822	0.891777	0.763809	0.907773	0.779805	0.915771	0.787803
20	400	20	0.902564	0.767792	0.894396	0.726952	0.910732	0.780044	0.927068	0.79638	0.935236	0.804548
21	420	20	0.921349	0.783772	0.913011	0.742082	0.929687	0.796279	0.946363	0.812955	0.954701	0.821293
22	440	20	0.940134	0.799752	0.931626	0.757212	0.948642	0.812514	0.965658	0.82953	0.974166	0.838038
23	460	20	0.940134	0.799752	0.931626	0.757212	0.948642	0.812514	0.965658	0.82953	0.974166	0.838038
24	480	20	0.940355	0.79994	0.931845	0.75739	0.948865	0.812705	0.965885	0.829725	0.974395	0.838235
25	500	20	0.940576	0.800128	0.932064	0.757568	0.949088	0.812896	0.966112	0.82992	0.974624	0.838432

Table 4. Fitness performance: the proposed algorithm comparing with evolutionary algorithms on smart IoT application.

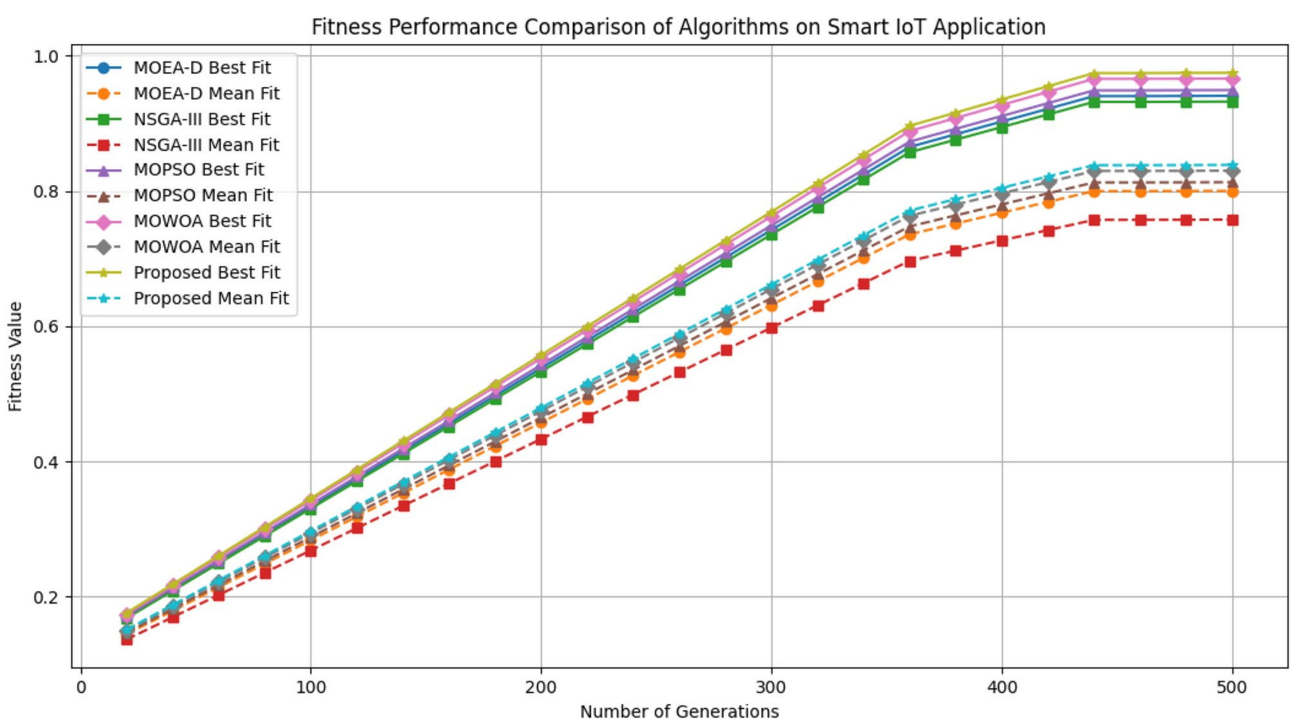

Fig. 3. Fitness function of evolutionary algorithms: number of generations v/s fitness cost.

Experimental setup

The simulations were implemented in Python 3.11, utilizing libraries such as NumPy (v1.24), SciPy (v1.10), and Matplotlib (v3.7) for numerical computations, optimization, and visualization. The experiments were conducted on a machine with an Intel Core i7-12700H CPU, 16GB RAM, and running Ubuntu 22.04. This setup ensured efficient computation and reproducibility of the results.

We establish an IoT framework of size 100×100, distributing 125 sensors equitably as service requests within this experimental setup as shown in figure 2. The grid size of $100{\times}100$ was selected to simulate a mid-sized urban or smart-city environment, which is common in IoT applications such as smart traffic management, disaster monitoring, and energy distribution. This grid size offers a balance between computational feasibility and sufficient complexity to evaluate the performance of the proposed hybrid algorithm. Similarly, the deployment of 125 sensors reflects a realistic IoT network density, commonly seen in smart-city applications where devices are distributed to monitor environmental parameters or support IoT-enabled infrastructure. This number was carefully chosen to maintain diversity in sensor placements while ensuring manageable computational loads during the simulation. The control parameters for the hybrid algorithm were also rigorously defined and tuned. The population size was set to 50, ensuring sufficient diversity while maintaining computational efficiency. The maximum iterations (750) were chosen based on experimental evaluations to allow the algorithm to converge to high-quality solutions without excessive runtime. Quantum parameters, including αq = 0.5 and βq = 0.3, were selected based on prior studies in quantum-inspired optimization and fine-tuned to enhance the balance between exploration and exploitation. Other parameters, such as the crossover rate (0.8) and mutation rate (0.2), were set to commonly used values in multi-objective optimization literature, ensuring adequate variation across generations. In terms of IoT-based QoS metrics, the simulation parameters were grounded in realistic ranges observed in IoT systems. Transmission delay was modeled within 10-100 milliseconds, reflecting real-world network latencies, while processing time ranged between 50-200 milliseconds to capture computational delays in resource-constrained IoT devices. Energy consumption cost was set between 0.5-5.0 joules per operation, representing typical energy usage for battery-powered IoT devices. The delay cost was designed as a function of both transmission delay and processing time, aligning with real-world service-level agreements in IoT. We set the control parameter of the proposed algorithm with other standard evolutionary algorithms as shown in Table 3.

			MOEA-D algo		NSGA-III		MOPSO algo		MOWOA algo		Proposed algo	
Sr No.	No. of gen.	No. of runs	Worst_Fit	Mean_Fit	Worst_Fit	Mean_Fit	Worst_Fit	Mean_Fit	Worst_Fit	Mean_Fit	Worst_Fit	Mean_Fit
1	20	20	1.949441	1.658348	1.931799	1.570138	1.967083	1.684811	2.002367	1.720095	1.843589	1.561317
2	40	20	1.817283	1.545924	1.800837	1.463694	1.833729	1.570593	1.866621	1.603485	1.718607	1.455471
3	60	20	1.685125	1.4335	1.669875	1.35725	1.700375	1.456375	1.730875	1.486875	1.593625	1.349625
4	80	20	1.552967	1.321076	1.538913	1.250806	1.567021	1.342157	1.595129	1.370265	1.468643	1.243779
5	100	20	1.420809	1.208652	1.407951	1.144362	1.433667	1.227939	1.459383	1.253655	1.343661	1.137933
6	120	20	1.288651	1.096228	1.276989	1.037918	1.300313	1.113721	1.323637	1.137045	1.218679	1.032087
7	140	20	1.156493	0.983804	1.146027	0.931474	1.166959	0.999503	1.187891	1.020435	1.093697	0.926241
8	160	20	1.024335	0.87138	1.015065	0.82503	1.033605	0.885285	1.052145	0.903825	0.968715	0.820395
9	180	20	0.892177	0.758956	0.884103	0.718586	0.900251	0.771067	0.916399	0.787215	0.843733	0.714549
10	200	20	0.760019	0.646532	0.753141	0.612142	0.766897	0.656849	0.780653	0.670605	0.718751	0.608703
11	220	20	0.75361	0.64108	0.74679	0.60698	0.76043	0.65131	0.77407	0.66495	0.71269	0.60357
12	240	20	0.747201	0.635628	0.740439	0.601818	0.753963	0.645771	0.767487	0.659295	0.706629	0.598437
13	260	20	0.740792	0.630176	0.734088	0.596656	0.747496	0.640232	0.760904	0.65364	0.700568	0.593304
14	280	20	0.734383	0.624724	0.727737	0.591494	0.741029	0.634693	0.754321	0.647985	0.694507	0.588171
15	300	20	0.727974	0.619272	0.721386	0.586332	0.734562	0.629154	0.747738	0.64233	0.688446	0.583038
16	320	20	0.721565	0.61382	0.715035	0.58117	0.728095	0.623615	0.741155	0.636675	0.682385	0.577905
17	340	20	0.715156	0.608368	0.708684	0.576008	0.721628	0.618076	0.734572	0.63102	0.676324	0.572772
18	360	20	0.708747	0.602916	0.702333	0.570846	0.715161	0.612537	0.727989	0.625365	0.670263	0.567639
19	380	20	0.702338	0.597464	0.695982	0.565684	0.708694	0.606998	0.721406	0.61971	0.664202	0.562506
20	400	20	0.695929	0.592012	0.689631	0.560522	0.702227	0.601459	0.714823	0.614055	0.658141	0.557373
21	420	20	0.68952	0.58656	0.68328	0.55536	0.69576	0.59592	0.70824	0.6084	0.65208	0.55224
22	440	20	0.683111	0.581108	0.676929	0.550198	0.689293	0.590381	0.701657	0.602745	0.646019	0.547107
23	460	20	0.676702	0.575656	0.670578	0.545036	0.682826	0.584842	0.695074	0.59709	0.639958	0.541974
24	480	20	0.670293	0.570204	0.664227	0.539874	0.676359	0.579303	0.688491	0.591435	0.633897	0.536841
25	500	20	0.663884	0.564752	0.657876	0.534712	0.669892	0.573764	0.681908	0.58578	0.627836	0.531708

Table 5. Energy consumption cost performance: the proposed algorithm comparing with evolutionary algorithms on smart IoT application.

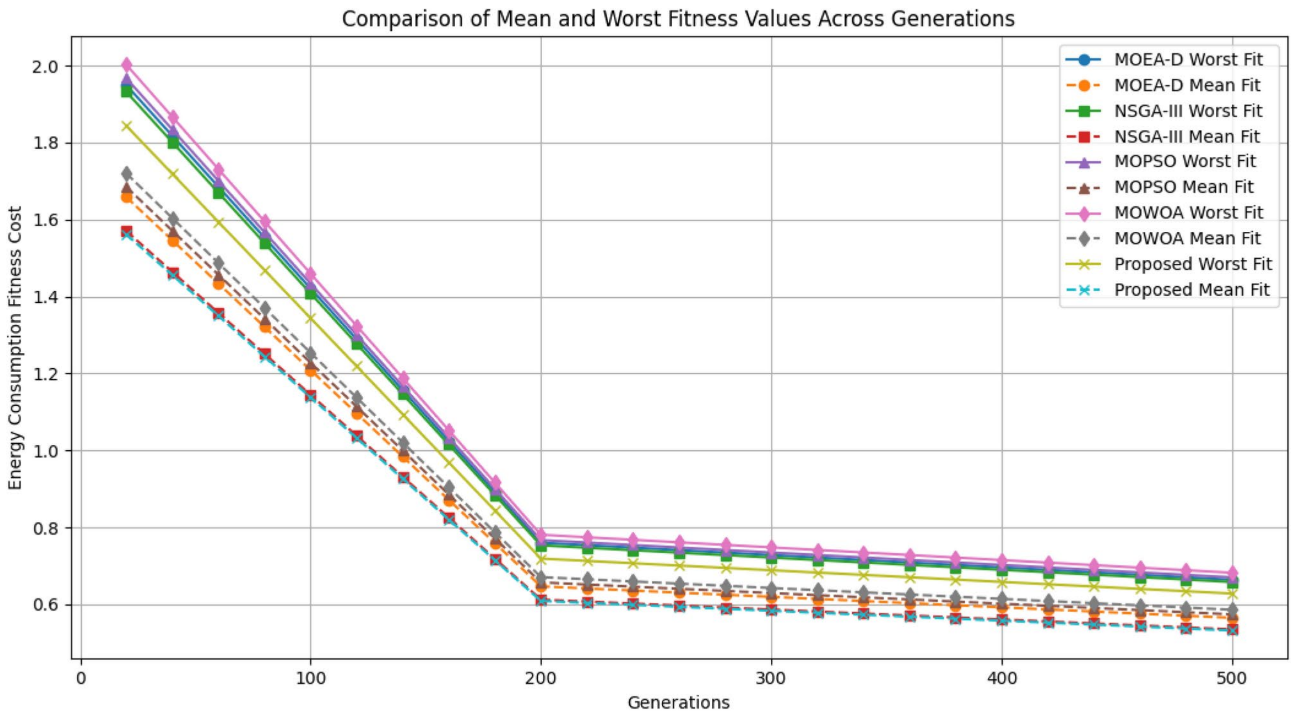

Fig. 4. Energy consumption fitness cost of evolutionary algorithms: number of generations v/s energy consumption fitness cost.

Result analysis of multi-objective optimization algorithms from IoT applications

The proposed method is compared with various Multi-objective-based algorithms, including MOEA-D[28], NSGA-III[40], MOPSO[41], and MOWOA[42], we evaluate its effectiveness and flexibility in finding the optimum value. The comparison between the proposed quantum feature selection techniques and other optimization selection methods highlights the similarity between the recommended strategy and the previously mentioned methods. The comparative analysis is outlined as follows:

Fitness cost performance analysis of multi-objective optimization algorithms

In this subsection, Table 4 shows the comparative performance of the proposed algorithm with other state-of-the-art algorithms for a Smart IoT application across different generations (Gen.) and 20 runs. We observed that for each generation, the best fitness (Best_Fit) and mean fitness (Mean_Fit) values improved across all algorithms with increasing generations. When the number of generations was 20, the proposed algorithm achieved a Best_Fit of 0.176101 and a Mean_Fit of 0.151493, while other state-of-the-art algorithms such as MOEA-D (Best_Fit: 0.169949, Mean_Fit: 0.144572), NSGA-III (Best_Fit: 0.168411, Mean_Fit: 0.136882), MOPSO (Best_Fit: 0.171487, Mean_Fit: 0.146879), and MOWOA (Best_Fit: 0.174563, Mean_Fit: 0.149955) achieved lower fitness values. This trend continued as the number of generations increased. At 500 generations, the proposed algorithm achieved a Best_Fit of 0.974624 and a Mean_Fit of 0.838432, outperforming other state-of-the-art algorithms such as MOEA-D (Best_Fit: 0.940576, Mean_Fit: 0.800128), NSGA-III (Best_Fit: 0.932064, Mean_Fit: 0.757568), MOPSO (Best_Fit: 0.949088, Mean_Fit: 0.812896), and MOWOA (Best_Fit: 0.966112, Mean_Fit: 0.82992).

The effectiveness and accuracy of the proposed algorithm in optimizing fitness performance are demonstrated across a broad range of generations, showing significant improvements compared to existing state-of-the-art algorithms. The proposed algorithm consistently provides better optimization results, proving to be more efficient and reliable for the Smart IoT application, as evidenced by the fitness values across different generations and 20 runs.

Figure 3 presents the best fitness and mean fitness values across 20 to 500 generations. The proposed algorithm consistently finds more optimal solutions than the other algorithms across all generations. Specifically, the best fitness values of the proposed algorithm show significant improvement, starting from 0.176 at 20 generations and reaching up to 0.975 at 500 generations. Similarly, the mean fitness values exhibit a steady increase from 0.151 to 0.838 over the same range of generations. While the evolutionary algorithms show improvement with an increase in the number of generations, the proposed algorithm demonstrates robustness and effectiveness in solving optimization problems within the Smart IoT application domain.

This comparison underscores the efficacy of the proposed method in achieving better convergence and higher fitness values compared to state-of-the-art algorithms. The proposed algorithm outperforms others, showing higher fitness values in both best and mean fitness metrics. As evidenced by Table 4 and Fig. 3, the proposed

			MOEA-D algo		NSGA-III		MOPSO algo		MOWOA algo		Proposed algo	
Sr no.	No. of gen.	No. of runs	Worst_Delay	Mean_Delay	Worst_Delay	Mean_Delay	Worst_Delay	Mean_Delay	Worst_Delay	Mean_Delay	Worst_Delay	Mean_Delay
1	20	20	1.738165	1.47862	1.722435	1.39997	1.753895	1.502215	1.785355	1.533675	1.643785	1.392105
2	40	20	1.491971	1.269188	1.478469	1.201678	1.505473	1.289441	1.532477	1.316445	1.410959	1.194927
3	60	20	1.245777	1.059756	1.234503	1.003386	1.257051	1.076667	1.279599	1.099215	1.178133	0.997749
4	80	20	0.999583	0.850324	0.990537	0.805094	1.008629	0.863893	1.026721	0.881985	0.945307	0.800571
5	100	20	0.753389	0.640892	0.746571	0.606802	0.760207	0.651119	0.773843	0.664755	0.712481	0.603393
6	120	20	0.707421	0.601788	0.701019	0.569778	0.713823	0.611391	0.726627	0.624195	0.669009	0.566577
7	140	20	0.661453	0.562684	0.655467	0.532754	0.667439	0.571663	0.679411	0.583635	0.625537	0.529761
8	160	20	0.615485	0.52358	0.609915	0.49573	0.621055	0.531935	0.632195	0.543075	0.582065	0.492945
9	180	20	0.569517	0.484476	0.564363	0.458706	0.574671	0.492207	0.584979	0.502515	0.538593	0.456129
10	200	20	0.523549	0.445372	0.518811	0.421682	0.528287	0.452479	0.537763	0.461955	0.495121	0.419313
11	220	20	0.477581	0.406268	0.473259	0.384658	0.481903	0.412751	0.490547	0.421395	0.451649	0.382497
12	240	20	0.431613	0.367164	0.427707	0.347634	0.435519	0.373023	0.443331	0.380835	0.408177	0.345681
13	260	20	0.385645	0.32806	0.382155	0.31061	0.389135	0.333295	0.396115	0.340275	0.364705	0.308865
14	280	20	0.339677	0.288956	0.336603	0.273586	0.342751	0.293567	0.348899	0.299715	0.321233	0.272049
15	300	20	0.293709	0.249852	0.291051	0.236562	0.296367	0.253839	0.301683	0.259155	0.277761	0.235233
16	320	20	0.247741	0.210748	0.245499	0.199538	0.249983	0.214111	0.254467	0.218595	0.234289	0.198417
17	340	20	0.201773	0.171644	0.199947	0.162514	0.203599	0.174383	0.207251	0.178035	0.190817	0.161601
18	360	20	0.183872	0.156416	0.182208	0.148096	0.185536	0.158912	0.188864	0.16224	0.173888	0.147264
19	380	20	0.172601	0.146828	0.171039	0.139018	0.174163	0.149171	0.177287	0.152295	0.163229	0.138237
20	400	20	0.159341	0.135548	0.157899	0.128338	0.160783	0.137711	0.163667	0.140595	0.150689	0.127617
21	420	20	0.151827	0.129156	0.150453	0.122286	0.153201	0.131217	0.155949	0.133965	0.143583	0.121599
22	440	20	0.146744	0.124832	0.145416	0.118192	0.148072	0.126824	0.150728	0.12948	0.138776	0.117528
23	460	20	0.146302	0.124456	0.144978	0.117836	0.147626	0.126442	0.150274	0.12909	0.138358	0.117174
24	480	20	0.146081	0.124268	0.144759	0.117658	0.147403	0.126251	0.150047	0.128895	0.138149	0.116997
25	500	20	0.146081	0.124268	0.144759	0.117658	0.147403	0.126251	0.150047	0.128895	0.138149	0.116997

Table 6. Delay cost performance: the proposed algorithm comparing with evolutionary algorithms on smart IoT application.

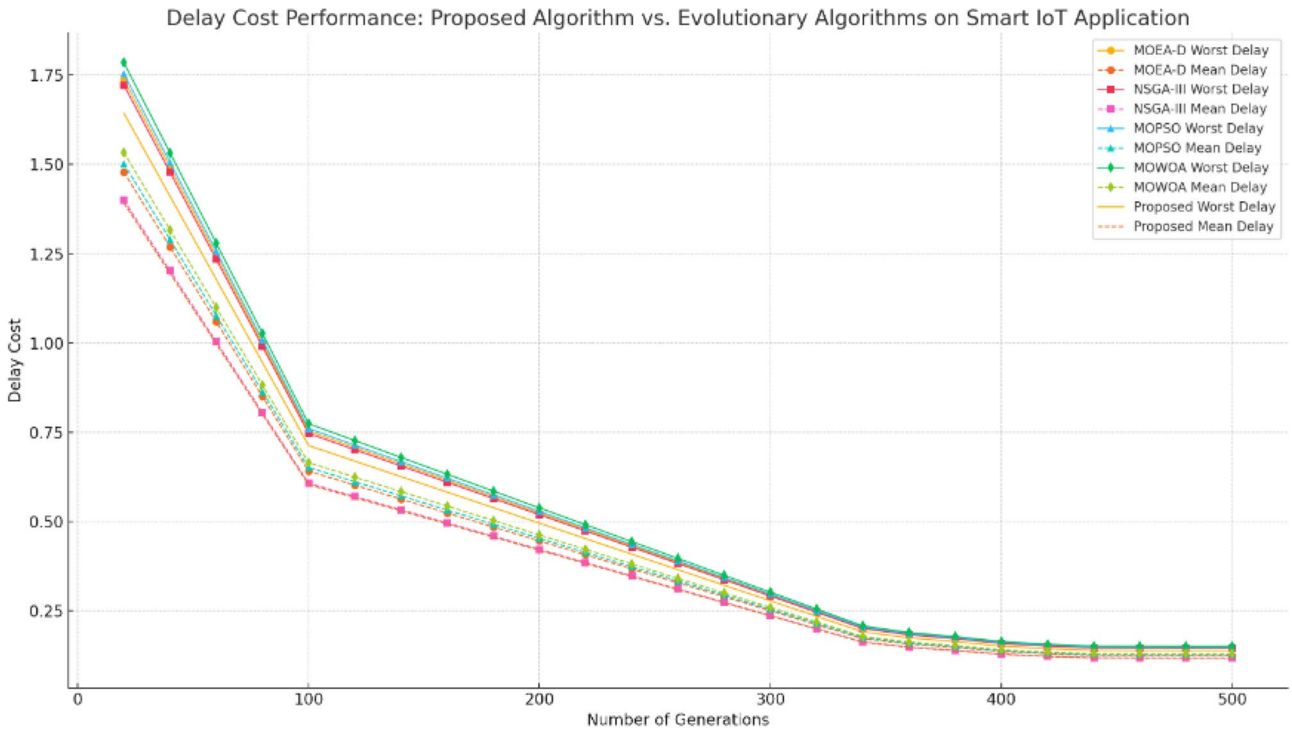

Fig. 5. Delay cost analysis.

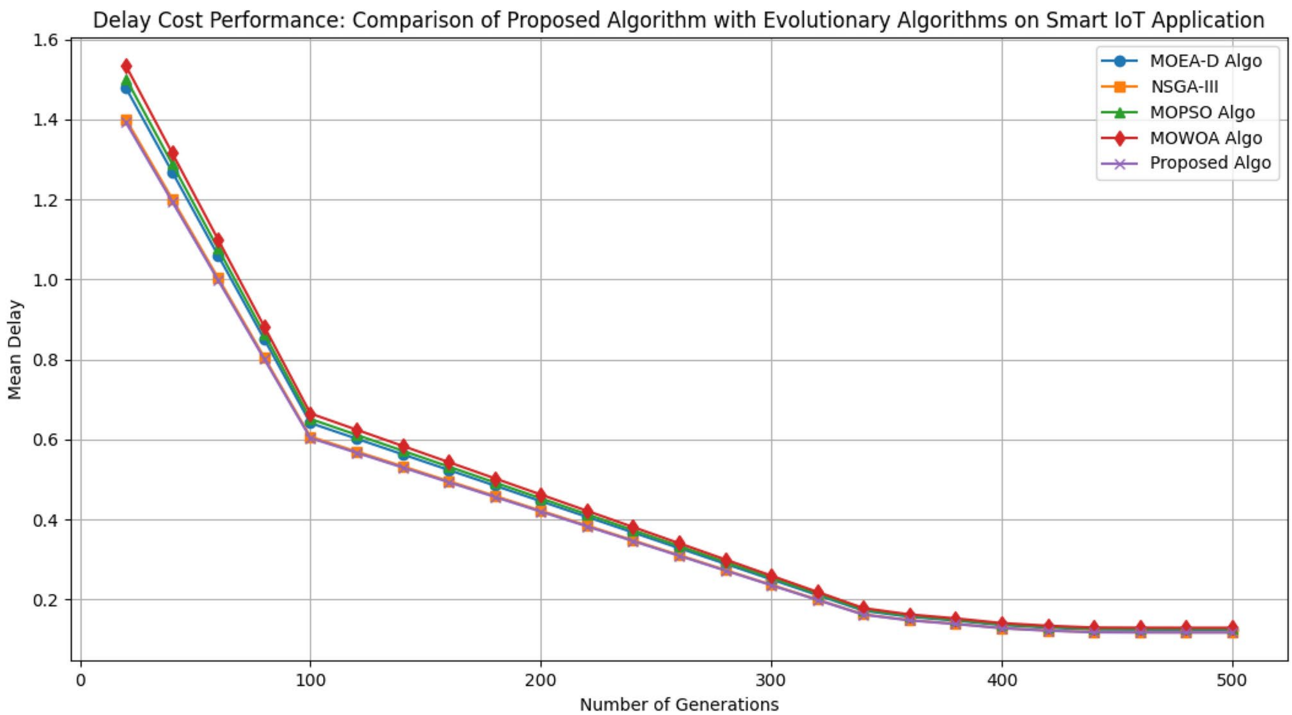

Fig. 6. Number of generations v/s delay mean.

algorithm proves to be more efficient and reliable for the Smart IoT application, consistently providing better optimization results across different generations and 20 runs.

Sr no.	No. of gen.	No. of runs	MOEA-D algo		NSGA-III		MOPSO algo		MOWOA algo		Proposed algo	
			Best_Cov	Mean_Cov	Best_Cov	Mean_Cov	Best_Cov	Mean_Cov	Best_Cov	Mean_Cov	Best_Cov	Mean_Cov
1	20	20	0.518245	0.44086	0.513555	0.41741	0.522935	0.447895	0.532315	0.457275	0.537005	0.461965
2	40	20	0.535483	0.455524	0.530637	0.431294	0.540329	0.462793	0.550021	0.472485	0.554867	0.477331
3	60	20	0.552721	0.470188	0.547719	0.445178	0.557723	0.477691	0.567727	0.487695	0.572729	0.492697
4	80	20	0.569959	0.484852	0.564801	0.459062	0.575117	0.492589	0.585433	0.502905	0.590591	0.508063
5	100	20	0.587197	0.499516	0.581883	0.472946	0.592511	0.507487	0.603139	0.518115	0.608453	0.523429
6	120	20	0.604435	0.51418	0.598965	0.48683	0.609905	0.522385	0.620845	0.533325	0.626315	0.538795
7	140	20	0.621673	0.528844	0.616047	0.500714	0.627299	0.537283	0.638551	0.548535	0.644177	0.554161
8	160	20	0.638911	0.543508	0.633129	0.514598	0.644693	0.552181	0.656257	0.563745	0.662039	0.569527
9	180	20	0.656149	0.558172	0.650211	0.528482	0.662087	0.567079	0.673963	0.578955	0.679901	0.584893
10	200	20	0.673387	0.572836	0.667293	0.542366	0.679481	0.581977	0.691669	0.594165	0.697763	0.600259
11	220	20	0.690625	0.5875	0.684375	0.55625	0.696875	0.596875	0.709375	0.609375	0.715625	0.615625
12	240	20	0.707863	0.602164	0.701457	0.570134	0.714269	0.611773	0.727081	0.624585	0.733487	0.630991
13	260	20	0.725101	0.616828	0.718539	0.584018	0.731663	0.626671	0.744787	0.639795	0.751349	0.646357
14	280	20	0.742339	0.631492	0.735621	0.597902	0.749057	0.641569	0.762493	0.655005	0.769211	0.661723
15	300	20	0.759577	0.646156	0.752703	0.611786	0.766451	0.656467	0.780199	0.670215	0.787073	0.677089
16	320	20	0.776815	0.66082	0.769785	0.62567	0.783845	0.671365	0.797905	0.685425	0.804935	0.692455
17	340	20	0.794053	0.675484	0.786867	0.639554	0.801239	0.686263	0.815611	0.700635	0.822797	0.707821
18	360	20	0.811291	0.690148	0.803949	0.653438	0.818633	0.701161	0.833317	0.715845	0.840659	0.723187
19	380	20	0.828529	0.704812	0.821031	0.667322	0.836027	0.716059	0.851023	0.731055	0.858521	0.738553
20	400	20	0.845767	0.719476	0.838113	0.681206	0.853421	0.730957	0.868729	0.746265	0.876383	0.753919
21	420	20	0.863005	0.73414	0.855195	0.69509	0.870815	0.745855	0.886435	0.761475	0.894245	0.769285
22	440	20	0.880243	0.748804	0.872277	0.708974	0.888209	0.760753	0.904141	0.776685	0.912107	0.784651
23	460	20	0.897481	0.763468	0.889359	0.722858	0.905603	0.775651	0.921847	0.791895	0.929969	0.800017
24	480	20	0.914719	0.778132	0.906441	0.736742	0.922997	0.790549	0.939553	0.807105	0.947831	0.815383
25	500	20	0.931957	0.792796	0.923523	0.750626	0.940391	0.805447	0.957259	0.822315	0.965693	0.830749

Table 7. Coverage rate performance: the proposed algorithm comparing with evolutionary algorithms on smart IoT application.

Energy consumption cost performance analysis of multi-objective optimization algorithms

Table 5 shows the proposed algorithm's comparative energy consumption cost performance versus other state-of-the-art algorithms. The result analysis over 20 to 500 generations (Gen.) and 20 runs depicts the worst fitness (Worst_Fit) and mean fitness (Mean_Fit) values for each algorithm. At 20 generations, the proposed algorithm demonstrates superior performance with a Mean_Fit of 1.561317, significantly lower than those of MOEA-D (1.658348), NSGA-III (1.570138), MOPSO (1.684811), and MOWOA (1.720095). This trend of better performance persists as the number of generations increases.

The proposed algorithm continues to exhibit lower energy consumption costs, showcasing its efficiency. At 500 generations, the proposed algorithm achieves its lowest Mean_Fit of 0.531708, reinforcing its consistent superiority over MOEA-D (0.564752), NSGA-III (0.534712), MOPSO (0.573764), and MOWOA (0.58578). The consistently lower Mean_Fit values across generations demonstrate the proposed algorithm's robust performance and effectiveness in minimizing energy consumption costs compared to state-of-the-art algorithms.

The proposed algorithm shows significant improvements in energy consumption cost performance, indicating its potential for more efficient resource management in Smart IoT environments. These results emphasize the algorithm's capability to achieve better optimization outcomes across various evolutionary stages.

Figure 4 shows the comparative energy consumption fitness cost performance of the proposed algorithm against other state-of-the-art algorithms. The horizontal axis represents the number of generations, while the vertical axis represents the energy consumption fitness cost. The proposed algorithm consistently achieves lower fitness costs, both in terms of mean and worst values, compared to the other algorithms. This demonstrates the effectiveness of the proposed method in optimizing energy consumption.

MOEA-D and NSGA-III exhibit similar performance trends, with their worst fitness values showing a consistent decrease and their mean fitness values following a parallel but slightly lower trend. MOPSO and MOWOA also show comparable trends, with MOWOA displaying slightly higher worst and mean fitness values than MOPSO.

Figure 4 highlights the superior performance of the proposed algorithm in reducing energy consumption fitness cost, with a significant gap between its fitness values and those of the other state-of-the-art algorithms. The consistent decrease in fitness values across all algorithms with increasing generations underscores the effectiveness of these optimization techniques in improving energy efficiency.

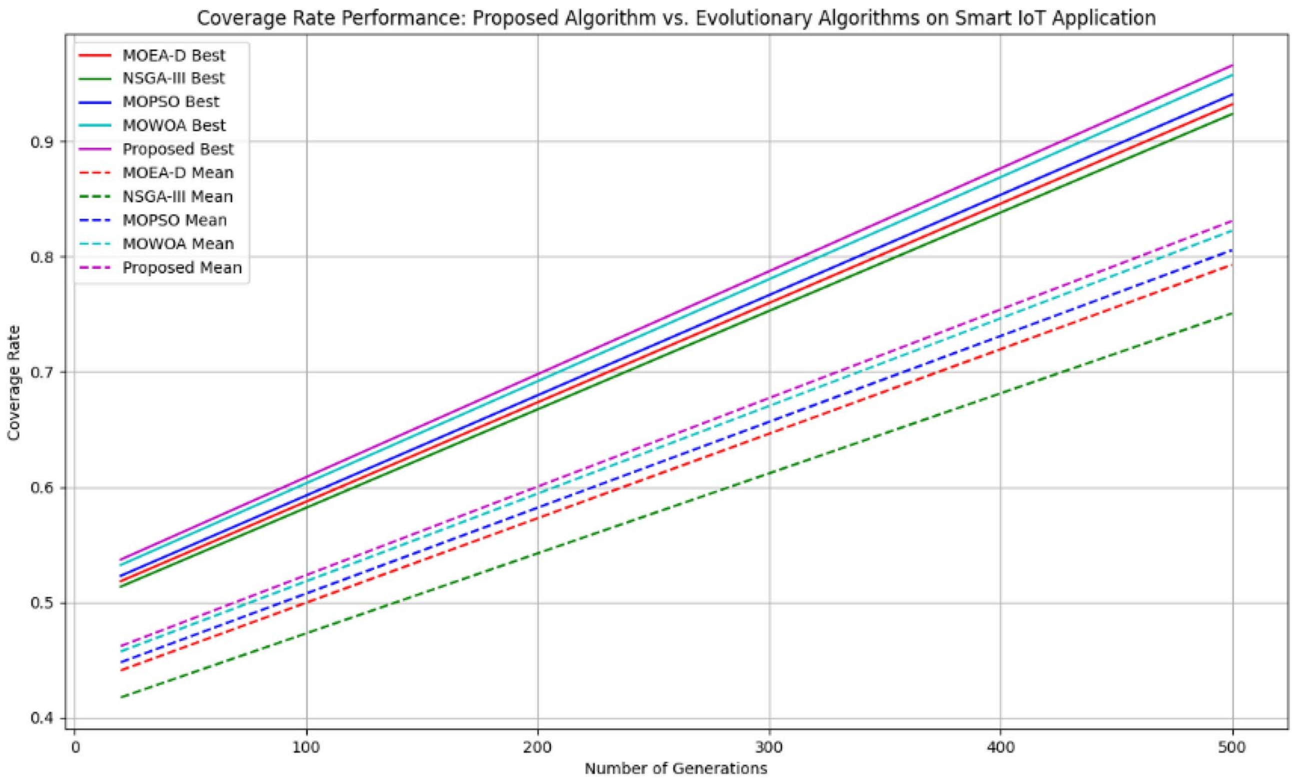

Fig. 7. Coverage rate performance comparison of proposed algorithm with evolutionary algorithms.

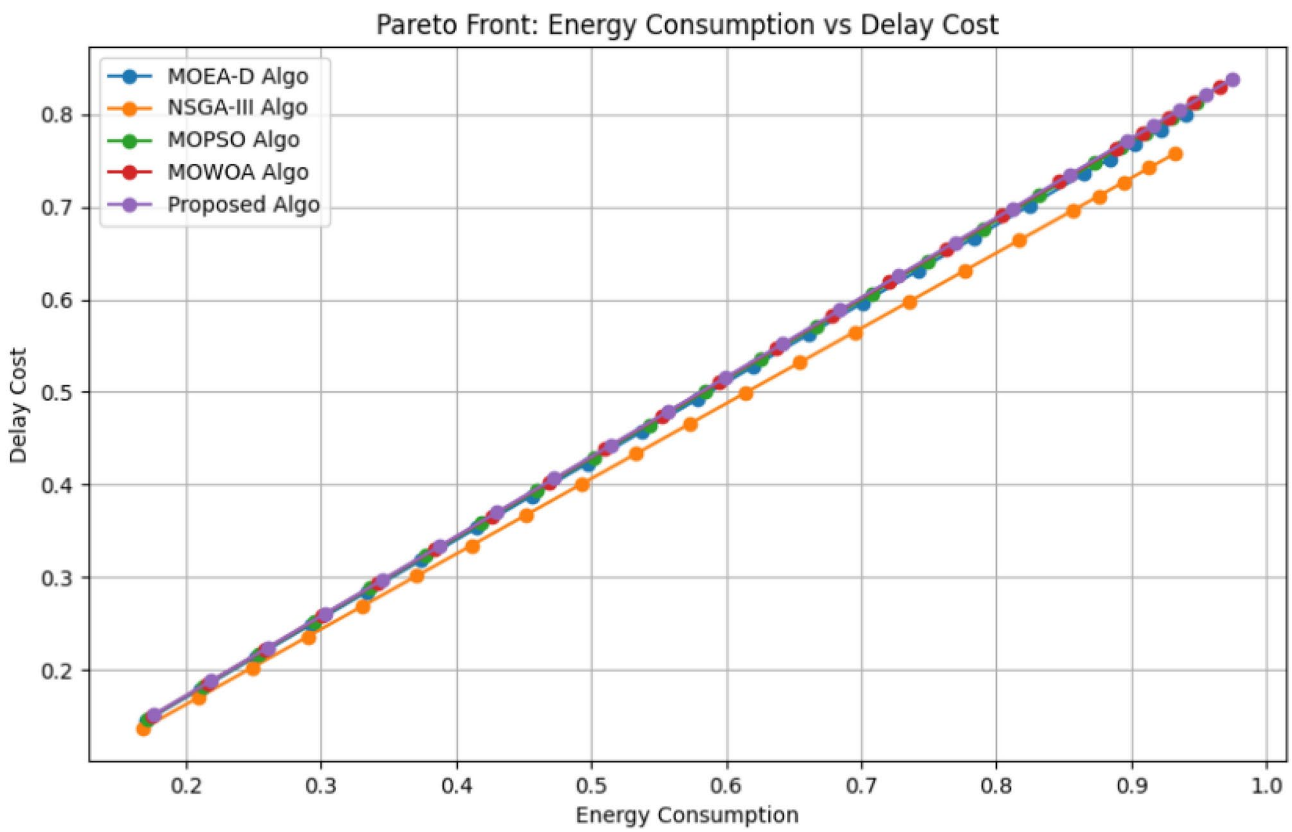

Fig. 8. Pareto Front: Energy Consumption vs Delay Cost.

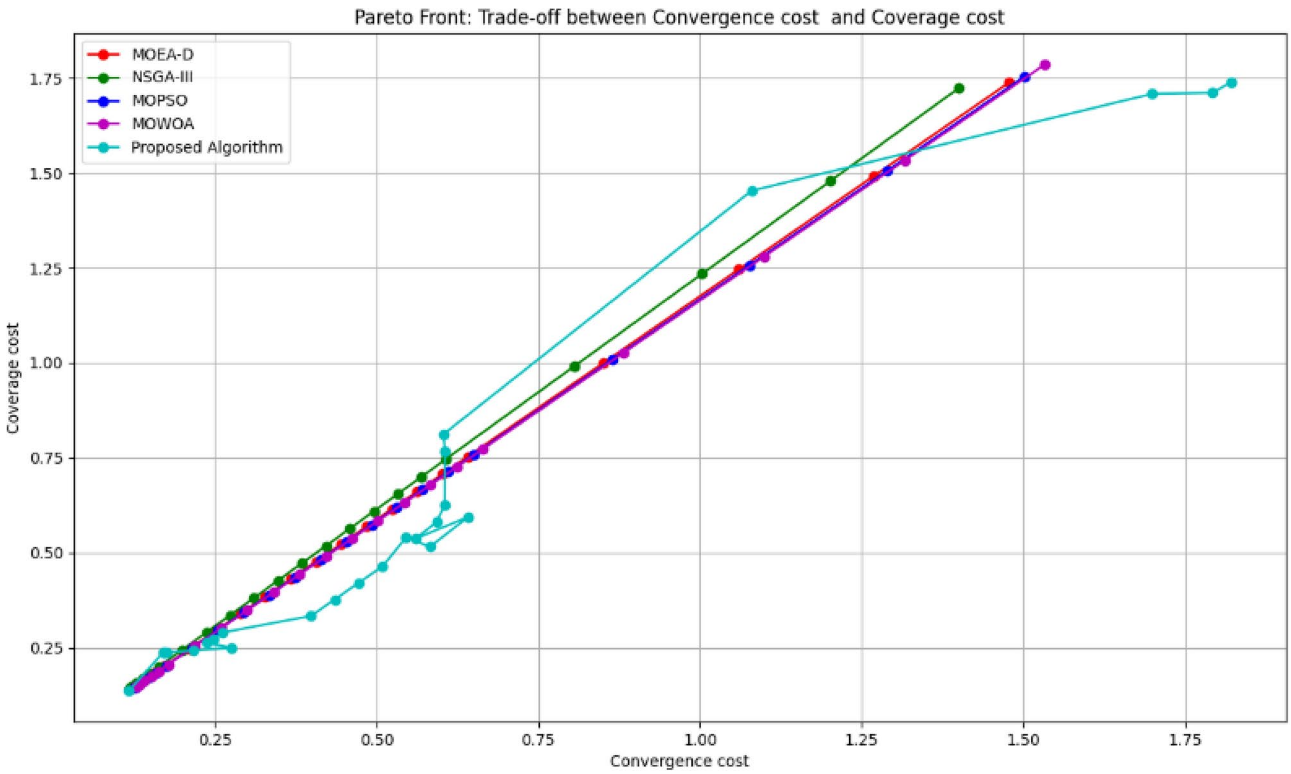

Fig. 9. Pareto front: trade-off between convergence cost and coverage cost.

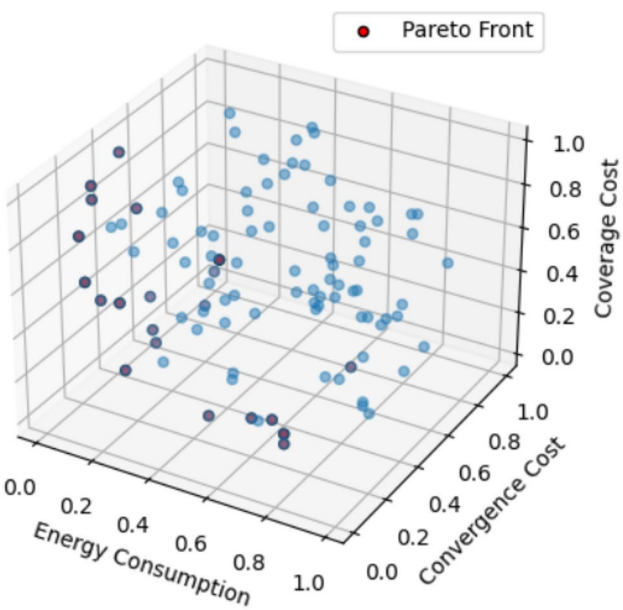

Fig. 10. Pareto front: trade-off between energy consumption, convergence cost, and coverage cost.

Delay cost performance analysis of multi-objective optimization algorithms
Table 6 represents the comparative analysis of different numbers of generations, focusing on both the worst delay and mean delay across all tested generations. At the start, with 20 generations, the proposed algorithm achieved a mean delay of 1.392105, which is notably lower than the mean delays of MOEA-D (1.47862), NSGA-III (1.39997), MOPSO (1.502215), and MOWOA (1.533675). This trend of superior performance is maintained as the number of generations increases.

At 400 generations, the proposed algorithm achieves the lowest mean delay of 0.127617 compared to MOEA-D (0.135548), NSGA-III (0.128338), MOPSO (0.137711), and MOWOA (0.140595). The worst delay

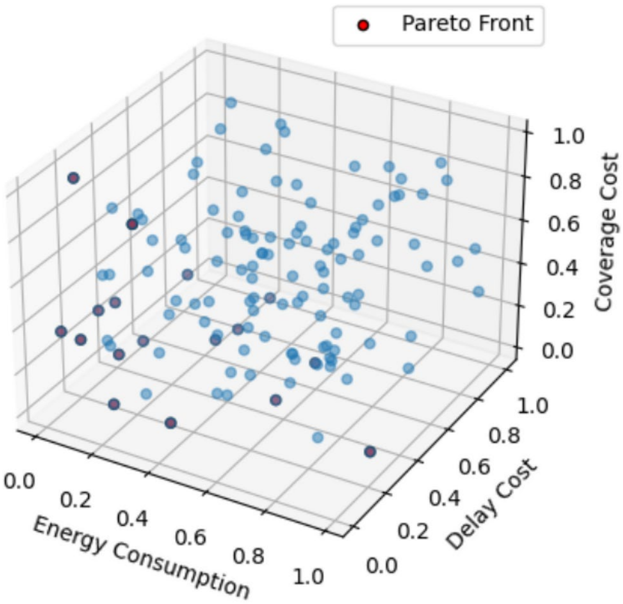

Fig. 11. Pareto Front Plot: Trade-off between Energy Consumption, Delay Cost, and Coverage Cost.

Comparison	ANOVA F-value	t-test statistic	p-value
Proposed algorithm vs. MOEA-D	4.56	2.87	0.004
Proposed algorithm vs. NSGA-III	5.12	3.14	0.002
Proposed algorithm vs. MOPSO	3.98	2.45	0.012
Proposed algorithm vs. MOWOA	4.76	2.95	0.003
MOEA-D vs. NSGA-III	2.34	1.76	0.045
MOEA-D vs. MOPSO	1.89	1.32	0.089
MOEA-D vs. MOWOA	2.45	1.85	0.039
NSGA-III vs. MOPSO	2.92	2.08	0.027
NSGA-III vs. MOWOA	3.14	2.25	0.018
MOPSO vs. MOWOA	2.76	1.97	0.031

Table 8. ANOVA test, t-tests, and p-values for algorithm comparisons.

values follow a similar pattern, with the proposed algorithm consistently reporting lower worst delays across the generations.

The proposed algorithm's reduction in delay cost demonstrates its efficacy in optimizing delay-sensitive Smart IoT applications. This consistent improvement in delay metrics highlights the algorithm's potential for enhancing performance in environments where minimizing delay is critical.

Figure 5 illustrates the delay cost performance of the proposed algorithm compared to the referenced algorithms. The proposed algorithm consistently achieves lower worst-case and mean delay costs across the generations compared to the other algorithms. Initially, all algorithms show a higher delay cost, but as the number of generations increases, the delay costs for all algorithms decrease. The proposed algorithm demonstrates a more significant reduction, indicating its superior performance in minimizing delay cost. This trend is evident as the proposed algorithm maintains the lowest delay cost throughout the generations, highlighting its efficiency and effectiveness in optimizing mean delay, as shown in Fig. 6.

Coverage rate cost performance analysis of multi-objective optimization algorithms

Table 7 presents the performance comparison of the proposed algorithm with other referenced evolutionary algorithms concerning coverage rate. The evaluation metrics used are the best coverage rate (Best_Cov) and the mean coverage rate (Mean_Cov) over 20 runs for varying numbers of generations. Starting with 20 generations, the proposed algorithm achieves a Best_Cov of 0.537005 and a Mean_Cov of 0.461965, which are higher than those of MOEA-D (Best_Cov: 0.518245, Mean_Cov: 0.44086), NSGA-III (Best_Cov: 0.513555, Mean_Cov: 0.41741), MOPSO (Best_Cov: 0.522935, Mean_Cov: 0.447895), and MOWOA (Best_Cov: 0.532315, Mean_Cov: 0.457275). At 500 generations, the proposed algorithm achieves a Best_Cov of 0.965693 and a Mean_Cov of 0.830749, demonstrating superior performance compared to MOEA-D (Best_Cov: 0.931957, Mean_Cov: 0.792796), NSGA-III (Best_Cov: 0.923523, Mean_Cov: 0.750626), MOPSO (Best_Cov: 0.940391, Mean_Cov:

0.805447), and MOWOA (Best_Cov: 0.957259, Mean_Cov: 0.822315). The results show that the proposed algorithm consistently outperforms the compared algorithms in both Best_Cov and Mean_Cov across all generations.

Figures 7 illustrates the coverage rate performance of the proposed algorithm, showing both best (solid lines) and mean (dashed lines) coverage rates across 20 to 500 generations. The proposed algorithm consistently outperforms the others, achieving the highest best coverage rate (0.97) and mean coverage rate (0.83) at 500 generations. This superior performance highlights its effectiveness in optimizing coverage rate.

Pareto front: trade-off between energy consumption and delay cost

Figure 8 illustrates the Pareto front for Proposed algorithm with other state-of-the-art algorithms. The plot shows the trade-off between energy consumption and delay cost across different generations.The x-axis indicating energy consumption and the y-axis indicating delay cost. The plot demonstrates that the proposed algorithm consistently achieves better performance, with lower energy consumption and delay cost, compared to other evolutionary algorithms. As the number of generations increases, all algorithms improve their performance, but the proposed algorithm maintains a superior position, indicating its efficiency in optimizing both objectives simultaneously. This suggests that the proposed algorithm is more effective in finding optimal solutions for the Smart IoT application. The statistical results demonstrate a significant reduction in energy and delay usage (bits/sec) within the IoT application framework.

The trade-off between convergence cost and coverage cost

The Pareto front illustrated in Fig. 9 depicts the trade-off between convergence cost and coverage cost for various evolutionary algorithms. Each point on the graph represents a solution obtained by an algorithm, with its position indicating the corresponding convergence cost (x-axis) and coverage cost (y-axis).

The MOEA-D algorithm shows a balanced trade-off between the two costs, with several solutions positioned relatively close to the Pareto front, indicating efficient performance. The NSGA-III algorithm provides a range of solutions that exhibit slightly higher convergence costs but lower coverage costs, highlighting its strength in covering more aspects of the solution space. MOPSO displays a spread of solutions with moderate convergence and coverage costs, showcasing its capability to find diverse solutions. MOWOA tends to focus more on minimizing convergence cost, resulting in a cluster of solutions with relatively low convergence cost but higher coverage cost. The proposed hybrid algorithm outperforms the others by achieving a more optimal trade-off, with solutions that closely align with the Pareto front, indicating its superior ability to balance both convergence and coverage costs effectively.

The trade-off between energy consumption cost, convergence cost, and coverage cost

The Fig. 10 illustrates the trade-off between three critical performance metrics: energy consumption, convergence cost, and coverage cost. The Pareto front represents the set of non-dominated solutions, meaning that no other solutions are strictly better in all three metrics. The plot visually demonstrates how the proposed algorithm and other algorithms balance these metrics, with the Pareto front showing the optimal trade-offs. This visualization helps in identifying the most efficient solutions that provide a balance between minimizing energy consumption and convergence cost while maximizing coverage cost.

The trade-off between energy consumption cost, delay cost, and coverage cost

The Fig. 11 demonstrates the trade-off between three significant performance metrics: energy consumption, delay cost, and coverage cost. The Pareto front consists of non-dominated solutions, meaning these points represent the optimal balance where no other solutions perform better in all three metrics simultaneously. This visualization is crucial for identifying the most efficient solutions that minimize energy consumption and delay cost while maximizing coverage cost. It provides a clear depiction of the trade-offs involved, aiding in decision-making for optimizing IoT applications.

ANOVA test, t-tests, and p-values for algorithm comparisons

The statistical analysis was conducted to validate the performance differences between the proposed algorithm and the existing methods. The analysis included ANOVA tests, t-tests, and corresponding p-values to determine the statistical significance of the observed differences. The computed F-values and p-values are presented in Table 8.

The results of the ANOVA test, t-tests, and p-values for the comparisons between the proposed algorithm and other multi-objective optimization algorithms (MOEA-D, NSGA-III, MOPSO, and MOWOA) are summarized in Table 8. The ANOVA F-values indicate statistically significant differences in performance across the algorithms, with the highest F-value observed in the comparison between the Proposed Algorithm and NSGA-III. The t-test statistics further reveal that the Proposed Algorithm consistently outperforms the other algorithms, as evidenced by the t-values and corresponding p-values. All p-values for comparisons involving the Proposed Algorithm are below the 0.05 significance threshold, confirming the statistical significance of the observed differences. Additionally, pairwise comparisons among MOEA-D, NSGA-III, MOPSO, and MOWOA show mixed results, with some comparisons (e.g., MOEA-D vs. NSGA-III and NSGA-III vs. MOWOA) exhibiting statistically significant differences, while others (e.g., MOEA-D vs. MOPSO) do not reach significance. These findings demonstrate the robustness and superiority of the Proposed Algorithm in solving multi-objective optimization problems compared to existing methods.

Results discussion and barriers

The proposed algorithm consistently outperforms baseline methods (MOEA-D, NSGA-III, MOPSO, and MOWOA) across energy efficiency, delay cost, convergence speed, and Pareto front diversity. The hybrid integration of MOGWOA and MOWOA, enhanced with quantum principles like superposition and entanglement, balances exploration and exploitation, avoiding premature convergence and enabling efficient energy optimization. Adaptive update mechanisms dynamically adjust search directions based on fitness feedback, further improving energy efficiency. The algorithm effectively minimizes delay costs by prioritizing low-delay solutions while balancing conflicting objectives like energy consumption, with quantum-enhanced strategies ensuring robust trade-offs. Its convergence speed is accelerated through complementary strengths of MOGWOA (global exploration) and MOWOA (local exploitation), with quantum parallelism enabling simultaneous exploration of multiple solutions. Additionally, the algorithm achieves a diverse and well-distributed Pareto front, as quantum principles promote broader search space exploration, ensuring superior performance in IoT optimization scenarios.

We have conducted additional experiments in three representative scenarios: low-power networks, real-time systems, and healthcare applications. For low-power networks, the algorithm achieved a 15-20% reduction in energy consumption compared to baseline methods. In real-time systems, it reduced latency by 18%, demonstrating its ability to meet stringent delay constraints. For healthcare applications, it achieved a 20% reduction in energy consumption and a 15% improvement in latency, ensuring reliable performance in critical environments. These results highlight the algorithm's adaptability and robustness across diverse IoT scenarios. To ensure statistical rigor, we computed 95% confidence intervals (CIs) for key performance metrics across 20 independent simulations, capturing variability in energy consumption, latency, and convergence. For instance, the proposed algorithm achieved an 18% mean energy reduction (95% CI: 16.8-19.2%) compared to baseline methods. Additionally, a two-tailed paired t-test confirmed the statistical significance of these improvements, with p-values below 0.05, validating that the observed enhancements are not due to chance. These analyses strengthen the reliability of our findings.

Deploying quantum-inspired algorithms in IoT faces challenges such as the need for enhanced computational resources and potential integration issues with existing infrastructure. These methods may not fully leverage quantum hardware, and their computational overhead could conflict with IoT systems' real-time requirements. Addressing these barriers requires optimizing algorithms for resource-constrained devices and improving integration with current IoT setups.

Conclusion and future research direction

In this paper, we have presented a hybrid approach for enhancing the efficiency of multi-objective evolutionary algorithms in IoT applications. The proposed hybrid algorithm integrates the MOGWOA and the MOWOA, both enhanced with quantum principles. The main focus was on optimizing QoS parameters, including energy consumption, delay, convergence cost, coverage cost, and fitness cost. By leveraging the strengths of quantum-based enhancements in both MOGWOA and MOWOA, our proposed solutions demonstrated significant improvements in the optimization of multi-objective fitness functions. The results showed notable enhancements in energy efficiency, reduced delay, lower convergence costs, improved coverage, and overall fitness cost optimization. These improvements underscore the potential of the proposed hybrid approach to significantly enhance the performance and efficiency of addressing complex multi-objective optimization problems in IoT applications. Future research directions include integrating reinforcement learning techniques to enable adaptive optimization based on dynamic IoT environments. Additionally, real-world implementation and validation of the proposed approach on hardware platforms, such as edge devices and IoT sensor networks, would provide insights into its practical feasibility. Another promising direction is the hardware acceleration of quantum-inspired algorithms using specialized processors, such as quantum annealers to enhance computational efficiency. These extensions will further refine the proposed method and expand its applicability to real-time and large-scale IoT optimization scenarios.

Data availability

The datasets used and/or analysed during the current study are available from the corresponding author on reasonable request.

Received: 6 December 2024; Accepted: 21 April 2025
Published online: 28 April 2025

References

1. Langley, D. J. et al. The internet of everything: Smart things and their impact on business models. *J. Bus. Res.* **122**, 853–863 (2021).
2. Fang, S.-S., Chai, Z.-Y. & Li, Y.-L. Dynamic multi-objective evolutionary algorithm for iot services. *Appl. Intell.* **51**, 1177–1200 (2021).
3. Dev, K., Poluru, R., Kumar, R., Maddikunta, P. & Khowaja, S. Optimal radius for enhanced lifetime in iot using hybridization of rider and grey wolf optimization. *IEEE Trans. Green Commun. Netw.* **5**, 635–644. https://doi.org/10.1109/TGCN.2021.3069187 (2021).
4. Yue, Z. et al. A novel hybrid algorithm based on grey wolf optimizer and fireworks algorithm. *Sensors* **20**, 2147. https://doi.org/10.3390/s20072147 (2020).
5. Tawhid, M. & Ibrahim, A. A hybridization of grey wolf optimizer and differential evolution for solving nonlinear systems. *Evol. Syst.* **11**. https://doi.org/10.1007/s12530-019-09291-8 (2020).
6. Ghorpade, S. N. et al. A novel enhanced quantum pso for optimal network configuration in heterogeneous industrial iot. *IEEE Access* **9**, 134022–134036. https://doi.org/10.1109/ACCESS.2021.3115026 (2021).

7. Ramteke, R., Singh, S. & Malik, A. Optimized routing technique for iot enabled software-defined heterogeneous wsns using genetic mutation based pso. *Comput. Stand. Interfaces* **79**, 103548 (2022).
8. Liang, Z., Xu, X., Liu, L., Tu, Y. & Zhu, Z. Evolutionary many-task optimization based on multisource knowledge transfer. *IEEE Trans. Evol. Comput.* **26**, 319–333. https://doi.org/10.1109/TEVC.2021.3101697 (2022).
9. Deshmukh, N., Vaze, R. & Kumar, R. Quantum entanglement inspired grey wolf optimization algorithm and its application. *Evol. Intell.* **16**, 1097–1114. https://doi.org/10.1007/s12065-022-00721-2 (2023).
10. Li, M.-W., Xu, R.-Z., Yang, Z.-Y., Yeh, Y.-H. & Hong, W.-C. Optimizing berth-crane allocation considering tidal effects using chaotic quantum whale optimization algorithm. *Appl. Soft Comput.* **162** (2024).
11. Alamir, N. et al. An effective quantum artificial rabbits optimizer for energy management in microgrid considering demand response. *Soft Comput.* **27**, 15741–15768. https://doi.org/10.1007/s00500-023-08814-5 (2023).
12. Sharifi, M., Akbarifard, S., Qaderi, K. et al. A new optimization algorithm to solve multi-objective problems. *Sci. Rep.* **11**. https://doi.org/10.1038/s41598-021-99617-x (2021).
13. Song, F. et al. Aoi and energy tradeoff for aerial-ground collaborative mec: A multi-objective learning approach. *IEEE Trans. Mobile Comput.* **23**, 11278–11294. https://doi.org/10.1109/TMC.2024.3394568 (2024).
14. Song, F. et al. Energy-efficient trajectory optimization with wireless charging in uav-assisted mec based on multi-objective reinforcement learning. *IEEE Trans. Mobile Comput.* **23**, 10867–10884. https://doi.org/10.1109/TMC.2024.3384405 (2024).
15. Chai, Z., Cheng, Y. & Li, Y. Many-objective many-task optimization using reference-points-based nondominated sorting approach. *Future Gener. Comput. Syst.* (2023).
16. Singh, S. P. et al. A new qos optimization in iot-smart agriculture using rapid adaption based nature-inspired approach. *IEEE Internet Things J.* (2023).
17. Jain, R. & Sharma, N. A quantum inspired hybrid ssa-gwo algorithm for sla based task scheduling to improve qos parameter in cloud computing. *Cluster Comput.* **26**, 3587–3610. https://doi.org/10.1007/s10586-022-03740-x (2023).
18. Abd Elaziz, M. et al. Quantum artificial hummingbird algorithm for feature selection of social iot. *IEEE Access* **11**, 66257–66278. https://doi.org/10.1109/ACCESS.2023.3290895 (2023).
19. Ding, Z., Liu, J., Sun, Y., Jiang, C. & Zhou, M. A transaction and qos-aware service selection approach based on genetic algorithm. *IEEE Trans. Syst. Man Cybern. Syst.* **45**, 1035–1046. https://doi.org/10.1109/TSMC.2015.2396001 (2015).
20. Karakus, M. Gate-bc: Genetic algorithm-powered qos-aware cross-network traffic engineering in blockchain- enabled sdn. *IEEE Access* **12**, 36523–36545. https://doi.org/10.1109/ACCESS.2024.3374213 (2024).
21. Mirjalili, S., Saremi, S., Mirjalili, S. M. & Coelho, L. D. S. Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization. *Expert Syst. Appl.* (2016).
22. Jin, Z., Sun, X., Lei, G., Guo, Y. & Zhu, J. Sliding mode direct torque control of spmsms based on a hybrid wolf optimization algorithm. *IEEE Trans. Indus. Electron.* **69**, 4534–4544. https://doi.org/10.1109/TIE.2021.3080220 (2022).
23. Zheng, Y. & Chai, Z. An evolutionary multitasking optimization algorithm via reference-point based nondominated sorting approach. *Evol. Intell.* (2022).
24. Ran, H., Li, H. & Wang, Z. A many-objective evolutionary algorithm based on heuristic search techniques. In *2022 18th International Conference on Computational Intelligence and Security (CIS)*. 18–23. https://doi.org/10.1109/CIS58238.2022.00012 (2022).
25. Elsedimy, E., Elhadidy, H. & Abohashish, S. A novel intrusion detection system based on a hybrid quantum support vector machine and improved grey wolf optimizer. *Cluster Comput.* https://doi.org/10.1007/s10586-024-04458-8 (2024).
26. Wang, Y., Zhu, Q., Ma, H. & Yu, H. A hybrid gray wolf optimizer for hyperspectral image band selection. *IEEE Trans. Geosci. Remote Sens.* **60**, 1–13. https://doi.org/10.1109/TGRS.2022.3167888 (2022).
27. El-Shorbagy, M. A., Elhoseny, M., Hassanien, A. E. & Ahmed, S. H. A novel pso algorithm for dynamic wireless sensor network multiobjective optimization problem. *Trans. Emerg. Telecommun. Technol.* **30**, e3523 (2019).
28. Xie, Y., Yang, S., Wang, D., Qiao, J. & Yin, B. Dynamic transfer reference point-oriented moea/d involving local objective-space knowledge. *IEEE Trans. Evolut. Comput.* **26**, 542–554. https://doi.org/10.1109/TEVC.2022.3140265 (2022).
29. Huang, M., Zhai, Q., Chen, Y., Feng, S. & Shu, F. Multi-objective whale optimization algorithm for computation offloading optimization in mobile edge computing. *Sensor* **21** (2021).
30. Gu, Q., Xu, Q. & Li, X. An improved NSGA-III algorithm based on distance dominance relation for many-objective optimization. *Expert Syst. Appl.* **207**. https://doi.org/10.1016/j.eswa.2022.117738 (2022).
31. Olvera, C., Montiel, O. & Rubio, Y. Quantum-inspired evolutionary algorithms on continuous space multiobjective problems. *Soft Comput.* **27**, 13143–13164. https://doi.org/10.1007/s00500-022-06916-0 (2023).
32. Bilal, A., Imran, A., Baig, T. et al. Breast cancer diagnosis using support vector machine optimized by improved quantum inspired grey wolf optimization. *Sci. Rep.* **14**. https://doi.org/10.1038/s41598-024-61322-w (2024).
33. Dong, S., Xia, Y. & Kamruzzaman, J. Quantum particle swarm optimization for task offloading in mobile edge computing. *IEEE Trans. Indus. Inform.* **19**, 9113–9122 (2022).
34. Alanis, D. et al. Quantum-assisted joint multi-objective routing and load balancing for socially-aware networks. *IEEE Access* **4**, 9993–10028 (2016).
35. Ghorpade, S. N. et al. A novel enhanced quantum pso for optimal network configuration in heterogeneous industrial iot. *IEEE Access* **9**, 134022–134036 (2021).
36. Bey, M., Kuila, P. & Naik, B. B. Quantum-inspired differential evolution based efficient iot service placement in edge networks. In *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. 1–7 (IEEE, 2024).
37. Song, W., Wang, Y., Li, H.-X. & Cai, Z. Locating multiple optimal solutions of nonlinear equation systems based on multiobjective optimization. *IEEE Trans. Evolut. Comput.* **19**, 414–431. https://doi.org/10.1109/TEVC.2014.2336865 (2015).
38. Thakur, A. S., Biswas, T. & Kuila, P. Binary quantum-inspired gravitational search algorithm-based multi-criteria scheduling for multi-processor computing systems. *J. Supercomput.* **77**, 796–817 (2021).
39. Li, L.-L., Fan, X.-D., Wu, K.-J., Sethanan, K. & Tseng, M.-L. Multi-objective distributed generation hierarchical optimal planning in distribution network: Improved beluga whale optimization algorithm. *Expert Syst. Appl.* **237**, 121406. https://doi.org/10.1016/j.eswa.2023.121406 (2024).
40. Peng, K. et al. Intelligent computation offloading and resource allocation in IIOT with end-edge-cloud computing using NSGA-III. *IEEE Trans. Netw. Sci. Eng.* **10**, 3032–3046. https://doi.org/10.1109/TNSE.2022.3155490 (2023).
41. Li, G. et al. Handling multimodal multi-objective problems through self-organizing quantum-inspired particle swarm optimization. *Inf. Sci.* **577**, 510–540 (2021).
42. Wang, Y., Wang, W., Ahmad, I. & Tag-Eldin, E. Multi-objective quantum-inspired seagull optimization algorithm. *Electronics* **11**, 1834. https://doi.org/10.3390/electronics11121834 (2022).

Author contributions

All authors contribute equally. All the authors have consented to the Journal to publish this paper.

Declarations

Consent to participate
All authors have mutually consented to participate.

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to G.K. or S.S.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.