QUANTUM-ENHANCED RED DEER OPTIMIZATION FOR OPTIMIZING TASK SCHEDULING AND ENERGY EFFICIENCY IN CLOUD-BASED HEALTHCARE SYSTEMS

NATARAJ N
DEPARTMENT OF INFORMATION TECHNOLOGY, BANNARI AMMAN INSTITUTE OF TECHNOLOGY, SATHYAMANGALAM, ERODE

PURUSHOTHAMAN P
DEPARTMENT OF CYBER SECURITY, PAAVAI ENGINEERING COLLEGE, NAMAKKAL.

ARUNA R
DEPARTMENT OF COMPUTER SCIENCE AND DESIGN, VEL TECH RANGARAJAN DR SAGUNTHALA R&D INSTITUTE OF SCIENCE AND TECHNOLOGY, CHENNAI
AUTHOR FOR CORRESPONDENCE E-MAIL: n.nataraj08@gmail.com

Abstract:
Cloud computing has revolutionized the healthcare industry by providing scalable, secure, and cost-effective solutions for healthcare data management and applications. This article presents Quantum-Enhanced Red Deer Optimization (Q-RDO), a new hybrid optimization algorithm that combines the Red Deer Algorithm (RDO) and Quantum-Inspired Particle Swarm Optimization (QIPSO) to solve complex problems in cloud-based health care systems, including task scheduling, resource allocation, and energy optimization. The RDO element simulates red deer social and territory behaviour to permit an effective equilibrium between global exploration and local exploitation. As opposed to QIPSO, which applies quantum principles to strengthen global search ability in order to optimize the process and avoid premature convergence. Q-RDO is utilized to improve the optimization process in healthcare by optimizing real-time resource management. This helps to reduce the energy usage by 15%, enhances task scheduling efficiency by 21.2% and improves patient scheduling and resource allocation-RDO can efficiently handle the operational challenges exhibited by modern healthcare systems by improving sustainability and managing costs while ensuring improved patient results.
Keywords: Quantum-Enhanced Red Deer Optimization, Cloud Computing, Healthcare, Task Scheduling, Resource Allocation, Energy Efficiency.

INTRODUCTION

Cloud computing has evolved to become a pillar of the new healthcare economy, offering a scalable, flexible, and cost-effective infrastructure for hosting massive amounts of health information. It enables medical records storage, sharing, and analysis on various platforms and locations, enhancing access to healthcare worldwide. Cloud-based healthcare systems have shown to be useful in providing real-time data sharing, telemedicine, patient tracking, and management of medical resources which are crucial in enhancing the quality and effectiveness of healthcare services [1]. Growing complexity of healthcare systems and increasing necessity for real-time decision-making and resource management necessitate efficient optimization methods. The conventional task scheduling and resource allocation mechanisms in healthcare systems usually result in inefficiencies, including patient waiting times, underutilization of resources, and wastage of energy. It is important to optimize these processes within cloud computing environments to improve the delivery of healthcare services, minimize operational expenditure, and enhance the health outcomes of patients. Advanced optimization methods are, therefore, in high demand to effectively address these inefficiencies [2].
A number of optimization algorithms have been investigated for managing healthcare resources, including Particle Swarm Optimization (PSO), Genetic Algorithms (GA), and Ant Colony Optimization (ACO). These methods have been used effectively to perform activities such as resource allocation, scheduling, and task assignment in healthcare environments. Although these approaches have achieved some gains, it continues to experience challenges in achieving exploration-exploitation balance, dealing with real-time and dynamic requirements, and evading local optima. In addition, conventional optimization techniques tend to fall short of the computational efficiency that is required for large-scale healthcare data and sophisticated cloud settings [3]. The main challenge in optimizing healthcare systems using cloud computing is the dynamicity of healthcare settings, the necessity for

massive, real-time processing of healthcare data, as well as the complexity of achieving task scheduling, resource allocation, and energy efficiency. Additionally, the healthcare systems tend to experience problems such as system latency, variability of patient flow, and responding to real-time needs. These problems hamper the efficiency of current optimization methods with it being challenging to attain hoped-for gains in terms of efficiency, cost savings, and patient satisfaction [4].

To overcome such challenges, superior hybrid optimization methods such as Quantum-Enhanced Red Deer Optimization (Q-RDO) can be investigated. By leveraging the power of the Red Deer Algorithm (RDA) and Quantum-Inspired Particle Swarm Optimization (QIPSO), Q-RDO has the capability to improve exploration and exploitation powers, offering superior adaptability to dynamic healthcare environments. Utilization of quantum-inspired concepts can aid in enhancing search powers, thereby avoiding local optima and accelerating the decision-making process for real-time healthcare applications [5]. Proposed Q-RDO algorithm provides a new approach to the inefficacies of current optimization techniques in healthcare systems. Its hybrid composition enables it to optimize healthcare processes, such as task scheduling, resource distribution, and energy usage, better than conventional algorithms. Q-RDO balancing of global exploration and local exploitation enables optimal real-time decision-making, solving the problems of dynamic nature of healthcare environments. With the integration of Q-RDO in cloud-based healthcare systems, the work in this proposal seeks to increase system efficiency, lower operating expenses, and enhance patient care as well as satisfaction with a more sustainable and scalable approach to meeting the increasing needs of contemporary healthcare [6].

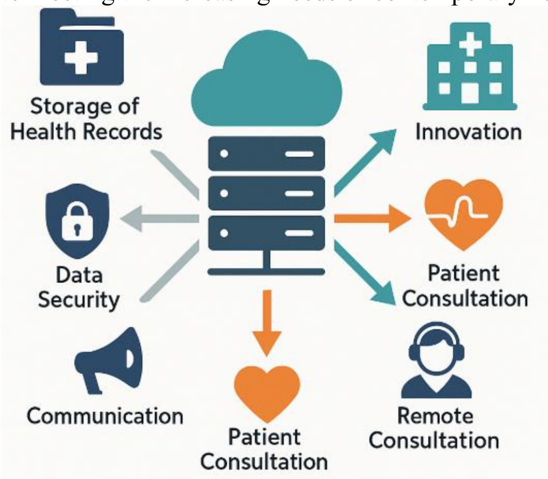

Figure 1: Role of cloud computing in healthcare

Cloud computing's significance in healthcare is depicted in Figure 1 which highlights its primary features including data security, communication, innovation, patient monitoring, remote consultation and health record storage. Better patient information management and improved healthcare services are made possible by cloud computing, which acts as a central hub for safe, scalable and real-time data sharing. It facilitates creative ideas, permits remote consultations and guarantees smooth communication between patients and healthcare professionals [7].

Main contributions of the proposed work

- In order to improve optimization in cloud-based healthcare systems, the proposed Quantum-Enhanced Red Deer Optimization (Q-RDO) is a hybrid approach that combines Red Deer approach (RDO) with Quantum-Inspired Particle Swarm Optimization (QIPSO).
- By dynamically assigning jobs according to the availability of resources in real time, it optimizes work scheduling, cutting down on patient wait times and increasing operational effectiveness.
- By guaranteeing the effective distribution of medical personnel, facilities, and equipment and optimizing resource use, Q-RDO improves resource allocation.
- By optimizing energy utilization in HVAC systems, medical equipment and lighting, it lowers energy consumption in healthcare facilities.
- The proposed approach guarantees scalability and flexibility in cloud environments for sophisticated, real-time healthcare systems.

The remainder of the paper is structures as follows: Section II explains the disadvantages of existing models, Section III explains the proposed work architecture, Section IV discusses the results and comparison with existing approaches. Section V concludes the work and direction for future work.

RELATED WORK

Alatoun, K., et al. (2022) presents a low-latency and energy-aware task scheduling architecture for the Internet of Medical Things (IoMT) in an edge fog-cloud network. The architecture maximizes task scheduling through offloading tasks among edge devices and cloud servers to reduce energy consumption and latency while enhancing the performance of healthcare systems as a whole. The model could lack scalability in very dynamic settings and perform poorly when there are wide fluctuations in resource availability or network conditions that might impact real-time scheduling effectiveness [8]. Bhasker, B., et al. (2025) proposes an energy-efficient task offloading method through IoMT cloud in healthcare applications. The method maximizes energy utilization by finding the optimal offloading choice for healthcare tasks in IoT-enabled cloud systems to enhance task performance and minimize energy cost. Nevertheless, the approach will not work effectively in situations with severely dynamic healthcare conditions or changing network conditions and its application might be restricted in larger or more complicated settings [9]. Beena, B. M., et al. (2025) is a green cloud framework for energy-aware task scheduling based on carbon intensity data to make optimal scheduling decisions in heterogeneous cloud servers. The method aims to minimize the energy consumption in cloud systems without lowering the high-performance requirements in healthcare applications. There is continuous monitoring of carbon intensity and hence computational overhead and the method can struggle to find the required trade-off between energy efficiency and real-time performance in mission-critical healthcare applications [10]. Choppara, P., & Lokesh, B. (2025) explains effective task scheduling and load balancing in fog computing for healthcare through deep reinforcement learning (DRL). The DRL-based system optimizes task allocation, minimizing latency and maximizing the use of resources across fog nodes and cloud servers. Relying on DRL can result in excessive computational expense due to the requirement of large training data and learning time, which is not applicable in real-time scenarios in resource-limited healthcare settings [11]. Malipatil, A. R., et al. (2025) discusses energy-efficient cloud computing using reinforcement learning-based scheduling of workloads. It improves the efficiency of computational resource allocation in cloud environments to ensure minimal energy consumption. RL-based models have a high computational requirement for training and are not necessarily flexible in dynamic workload environments with sporadic or unpredictable workloads, which could restrict their use in real-time applications in healthcare [12]. Ganesan, T. (2025) explores high-end task scheduling in cloud healthcare systems using hybrid MFO-PSO and Artificial Bee Colony (ABC) optimization. The hybrid algorithm enhances task scheduling efficiency and resource management in cloud environments. Computational complexity is enhanced by the hybrid nature of the algorithm, making it difficult to execute in real-time healthcare systems with constrained resources, and performance is reduced in extremely dynamic or uncertain scenarios [13].

Khan, A., et al. (2025) proposes EcoTaskSched, a machine learning-based hybrid technique for energy-aware task scheduling in IoT-based fog-cloud systems. The system employs machine learning-based models for optimizing energy-aware decisions on fog-to-cloud and cloud-to-fog offloading in IoT healthcare systems. The method needs extensive training data, which might not always be accessible in real-time healthcare systems. The system might also struggle to cope with unexpected, extremely dynamic healthcare scenarios [14]. Mondal, A., et al. (2024) suggests a dynamic energy-efficient task offloading method in mobile edge-cloud computing networks in healthcare. The system achieves optimal energy usage with minimal latency and quick offloading of tasks in mobile health systems. The model is energy efficient but could be less scalable for large-scale health networks with many devices, and real-time optimization might not be straightforward in dynamic network conditions of mobile edges [15]. Sing, R., et al. (2022) proposes EMCS, an energy-efficient makespan cost-conscious scheduling algorithm based on an evolutionary learning strategy for cloud-fog-based IoT applications. It minimizes energy consumption by optimizing resource scheduling and improves resource utilization in healthcare systems. The evolutionary learning strategy includes substantial computational overhead, which may not be appropriate for real-time performance in resource-constrained healthcare systems in large-scale settings [16].

Bharathi, R., et al. (2020) investigates energy-aware clustering for disease diagnosis in IoT-based sustainable healthcare systems. The model minimizes energy expenditure via smart device clustering while providing precise disease diagnosis. Clustering-based approaches have high computational costs and can suffer from scalability issues when handling large amounts of data in healthcare systems if scalability and flexibility are essential [17]. Latif, R. M. A., et al. (2025) introduces AI-based energy-efficient load balancing in hybrid edge-cloud networks for renewable energy networks. It maximizes load distribution and minimizes energy consumption while keeping the network stable in healthcare systems with limited energy. The AI-based aspect calls for large amounts of computational power to train the model and may not be suitable for healthcare systems with erratic data or limited resources [18]. Moparthi, N. R., et al. (2023) presents a better energy-efficient cloud-optimized load-balancing algorithm for IoT systems. It optimizes load balancing among cloud resources to save energy without sacrificing performance in healthcare IoT systems. The technique can fall short when dealing with a large amount of traffic or sudden surges in demand in dynamic healthcare environments, and its performance can get worse in bigger systems with different types of IoT devices [19].

Pandi, M., & Kumar, A. S. (2025) suggests optimal workload distribution based on the DEABC (Differential Evolution Artificial Bee Colony) Algorithm for IoT fog-cloud systems. The method optimizes power consumption and reduces latency for IoT-based healthcare services. The DEABC algorithm can suffer from scalability concerns in big, real-time healthcare systems with complicated task scheduling demands and heterogeneous workloads [20]. Khaledian, N., et al. (2024) proposes an energy-aware and deadline-based workflow scheduling algorithm in the cloud and fog framework. The algorithm schedules tasks in a way that maximizes energy efficiency and minimizes deadlines in real-time health care systems. It is challenging to schedule real-time urgent healthcare tasks with different deadlines and energy levels and the algorithm may not work effectively under extremely dynamic scenarios [21]. Pasha, F., & Natarajan, J. (2025) introduces an intelligent secure and efficient workflow scheduling

(SEWS) model for heterogeneous cloud computing systems. The model integrates security attributes and energy efficiency in task scheduling within the healthcare domain. The incorporated security attributes can enhance the computational load which can decrease real-time scheduling efficiency in resource-limited healthcare systems [22].

PROPOSED WORK

a) Overview of the Issue:

For effective data management, real-time communication, and resource optimization, healthcare systems are becoming more and more reliant on cloud computing. However, because of their dynamic nature, these systems frequently encounter difficulties with scheduling tasks, allocating resources, and ensuring energy efficiency. The complexity of large-scale healthcare facilities and real-time needs are too much for traditional optimization methods to handle. This calls for the creation of increasingly sophisticated algorithms that can effectively optimize these procedures while maintaining scalability and flexibility.

b) Reasons for the proposed work:

An approach to get around the drawbacks of conventional optimization methods in the healthcare industry is to use Quantum-Enhanced Red Deer Optimization, or Q-RDO. The goal of Q-RDO is to solve important problems with cloud-based healthcare systems, such as effectively allocating jobs to available resources (medical personnel and equipment) while reducing wait times and bottlenecks is known as task scheduling. In Resource Allocation, making the best use of scarce resources, like energy, medical personnel and equipment in order to increase throughput and lower operating expenses. An energy-intensive system is used to optimize the usage of lighting and medical equipment to lower the energy consumption of healthcare facilities.

c) Quantum-Enhanced Red Deer Optimization (Q-RDO) algorithm

The Quantum-improved Red Deer Optimization algorithm is a new hybrid optimization method that integrates two different paradigms: The Red Deer Algorithm (RDO) and Quantum-Inspired Particle Swarm Optimization (QIPSO). The reason behind this hybridization is to improve exploration and exploitation abilities, which are of significant importance to address intricate optimization problems in dynamic settings such as healthcare systems where real-time management of resources, task scheduling and power optimization are vital.

(i) Red Deer Algorithm (RDO):

RDO is motivated by the red deer's social and territorial nature, where the wander in groups collectively maintaining both exploration and exploitation. Exploration involves looking for fresh potential solutions by exploring unknown regions of the solution space, whereas exploitation involves concentrating on improving promising solutions found. In the RDO, the migration of the deer is simulated to respond based on how other deer move within the herd. This simulates the process of optimization, where deer (solutions) migrate to more promising regions while venturing into unexplored areas for better solutions. Balance between global search (exploration) and local search (exploitation) is central to successfully solving optimization problems in applications such as resource allocation and task scheduling where new solution discovery and solution improvement are both critical.

(ii) Quantum-Inspired Particle Swarm Optimization (QIPSO):

QIPSO is a variant of standard Particle Swarm Optimization (PSO) that adapts concepts from quantum mechanics, like quantum superposition and quantum entanglement, to the search process. In PSO, particles (solutions) travel in a solution space according to velocities based on their personal best-known position and the best-known position of the swarm. QIPSO introduces the quantum-inspired feature by enabling one particle to hold multiple solutions at once, essentially broadening the search space and enhancing search diversity. The quantum-inspired method lessens the likelihood of premature convergence to local optima, as observed in standard PSO. The quantum characteristics also enable the particles to exchange information more effectively, enhancing the global search ability and accelerating the search process. This renders QIPSO appropriate for dynamic, complex problems such as energy optimization and real-time health system resource management.

(iii) Hybridization of RDO and QIPSO (Q-RDO):

The integration of QIPSO and RDO in Q-RDO, therefore, augments the strengths of both approaches. The RDO adds to this by providing that the algorithm can be able to switch between exploration and exploitation, while the QIPSO part utilizes the principles of quantum mechanics to enable the algorithm to explore the solution space more effectively. Through this synergy, Q-RDO ensures that it can deal with the complexities of large and dynamic solution spaces, as are common in cloud-based healthcare systems. In Q-RDO, the algorithm starts with the initialization of a population of particles (solutions) that travel across the solution space and update their positions based on both global and local information. The RDO component helps particles sample the solution space widely while focusing on the most potential areas. The QIPSO component, in contrast, allows particles to represent a variety of solutions at the same time, improving sensitivity to the global optimal solution without being captured by local optima.

The velocity of particle i at time t+1 is updated using the Eq.(1). Three elements form the basis of QIPSO's velocity update such as w is the inertia weight that balances exploration and exploitation by regulating the effect of the prior velocity. The effects of the particle's global best position, gbest, and personal best position, $pbest_i$ are controlled by the cognitive and social factors, c1 and c2 respectively. r1 and r2 are random parameters ranging from 0 to 1 that enhance search space exploration and add diversity. $\mathrm{pbest_i}$ is the optimal location that particle pbest i has discovered. gbest is the optimal location determined by the swarm as a whole. Particle i's current position is represented by $xi(t)$.

$$v_{i(t+1)} = w \times v_{i(t)} + c1 \times r1 \times (\mathrm{pbest}_i - x_{i(t)}) + c2 \times r2 \times (\mathrm{gbest} - x_{i(t)}) \quad (1)$$

Particle i's position is updated by adding its present position $xi\ (t)$ x i (t) to the newly calculated velocity $vi\ (t + 1)$ v i (t+1). In order to move particles around the search space according to the updated velocity as given in Eq.(2).

$$x_{i(t+1)} = x_{i(t)} + v_{i(t+1)} \quad (2)$$

Using the Red Deer Algorithm (RDO), this Eq.(3) simulates particle motion such as α: The exploration factor that regulates the particle's amount of solution space exploration. Δx is a random component that adds variation to particle motion. The exploitation factor, β, regulates how much the particle narrows its search by heading in the direction of the best solution the swarm finds (x_{best}). This update preserves the ability to explore new places while enabling particles to travel toward potential areas.

$$x_{i(t+1)} = x_{i(t)} + \alpha \times \Delta x + \beta \times (x_{best} - x_{i(t)}) \quad (3)$$

The quantum superposition idea of QIPSO is represented by Eq.(4), where each particle concurrently represents several possible answers. The quantum superposition idea of QIPSO is represented by Eq.(4), where each particle concurrently represents several possible answers. P_i is the chance of each particle state is determined by the probability amplitude for each quantum state P_i, k. The quantum state at time t, which represents the potential solutions for particle i at that moment, is represented as $\psi_{k(t)}$. N is the total number of potential quantum states or solutions. By enabling particles to search several solutions at once, this equation improves the algorithm's capacity for global exploration and raises the possibility of discovering the global optimum.

$$x_{i(t+1)} = \sum_{k=1}^{N} P_i, k \times |\psi_{k(t)}\rangle \quad (4)$$

This Eq.(5) uses both random exploration and the global best solution (gbest) to update the particle's position: α: A weighting factor that strikes a balance between the search of the world's best and the existing position.

$$\mathrm{quantum_{pos}} = \alpha \times \mathrm{current_{position}} + (1 - \alpha) \times (\mathrm{gbest} + \mathrm{random_{exploration}}) \quad (5)$$

Particles that are in quantum entanglement exchange information about their optimal locations, which improves their ability to work together. $Q_{ij(t)}$ is the strength of the interaction between particles I and J is represented by the quantum entanglement factor or $Q_{ij(t)}$. $x_{j(t)}$ is the location of particle j according to shared quantum information. By exchanging information, particles can prevent premature convergence to local optima and instead converge more quickly towards an ideal solution.

$$x_{i(t+1)} = \sum_{j=1}^{N} Q_{ij(t)} \times \langle x_{j(t)} \rangle \quad (6)$$

The entangled information that particles share with their neighbors is computed by this Eq.(7). Using this common knowledge, the particles travel in the direction of better solutions. β is a component that regulates the impact of nearby particles' optimal placements.

$$\mathrm{entangled_{info}} = \beta \times \mathrm{average}(\mathrm{neighbor_{pbest}} - \mathrm{current_{position}}) \quad (7)$$

This Eq.(8) takes into account the entangled information to update the particle's position during the local search phase.

$$\mathrm{new_{position}} = \mathrm{current_{position}} + \mathrm{local_{search_{intensity}}} \times \mathrm{entangled_{info}} \quad (8)$$

The fitness function measures a solution's performance to determine the performance using Eq.(9). The solution's fitness value is represented by $F(x_i)$. $f(x_i)$ is the objective function that assesses the performance of the solution.

$$F(x_i) = f(x_i) \quad (9)$$

This objective function in Eq.(10) evaluates the effectiveness of task scheduling by weighing patient wait times against available resources (e.g., doctors, nurses, and equipment). A more effective scheduling system is indicated by a lower value.

$$f(x_i) = \frac{\text{Total waiting time}}{\text{Total resources}} \quad (10)$$

To assess the optimization solution, this weighted fitness function in Eq. (11) takes into account a number of variables. $w1$, $w2$, $w3$ are Weights for energy cost, resource usage, and scheduling efficiency, respectively.

$$F(x_i) = w1 \times \mathrm{scheduling_{efficiency}} + w2 \times \mathrm{resource_{utilization}} - w3 \times \mathrm{energy_{cost}} \quad (11)$$

The imbalance in resource allocation is measured by Equation (12). A smaller number indicates a better distribution of resources. It computes the difference between necessary and allotted resources.

$$f(x_i) = \Sigma|\mathrm{allocated_{resources_i}} - \mathrm{required_{resources_i}}| \quad (12)$$

In Eq.(13), the condition determines whether an optimal solution has been reached by the algorithm. The algorithm halts if the difference in fitness scores between successive iterations is less than a tiny threshold ϵ.

$$\text{Convergence if } |F(x_i) - F(x_{i(t-1)})| < \varepsilon \quad (13)$$

When applied to healthcare systems, Q-RDO optimizes processes like task scheduling, where healthcare services (consultations, lab tests, surgeries) need to be assigned to resources (staff and equipment). Q-RDO ensures that tasks are scheduled in a way that minimizes waiting times and bottlenecks while making the best use of available resources. It also provides resource allocation through the dynamic distribution of medical personnel, equipment, and rooms to suit the demand for service without overuse or underutilization of resources. Q-RDO also optimizes energy consumption by optimizing energy-intensive systems (e.g., HVAC, lights, and medical devices) through real-time data, enhancing sustainability and lowering operational expenses in health facilities.

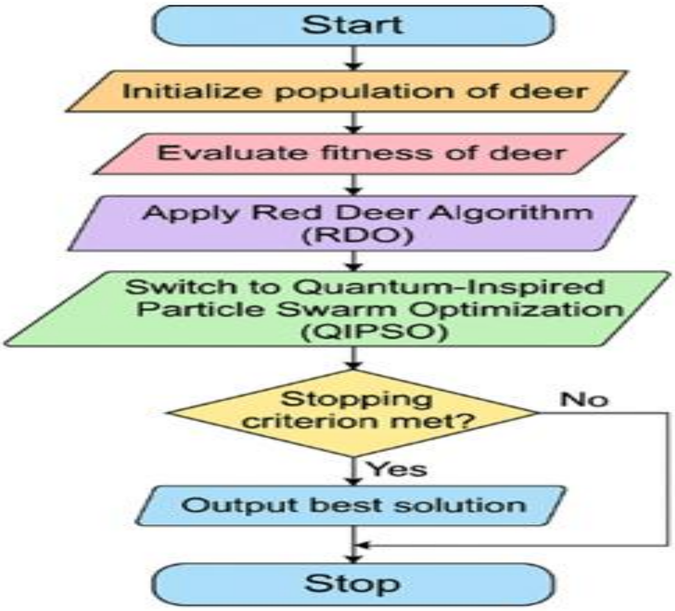

Figure 2: The working flow of the proposed algorithm

Figure 2 shows the working process of the Quantum-Enhanced Red Deer Optimization (Q-RDO) algorithm, which combines the Red Deer Algorithm (RDO) and Quantum-Inspired Particle Swarm Optimization (QIPSO) to optimize complicated tasks such as task scheduling, resource allocation, and energy efficiency in health systems. The process begins with initialization in which particles are randomly initialized and the best solutions are recorded. The algorithm goes through the exploration phase (global search) and then the exploitation phase (local search), enabling particles to optimize their solutions. The adaptation layer optimizes the search process in accordance with current conditions, and the evaluation layer evaluates the quality of the solutions through fitness functions. Lastly, the termination layer ensures convergence and terminates the algorithm when optimal solutions are achieved.

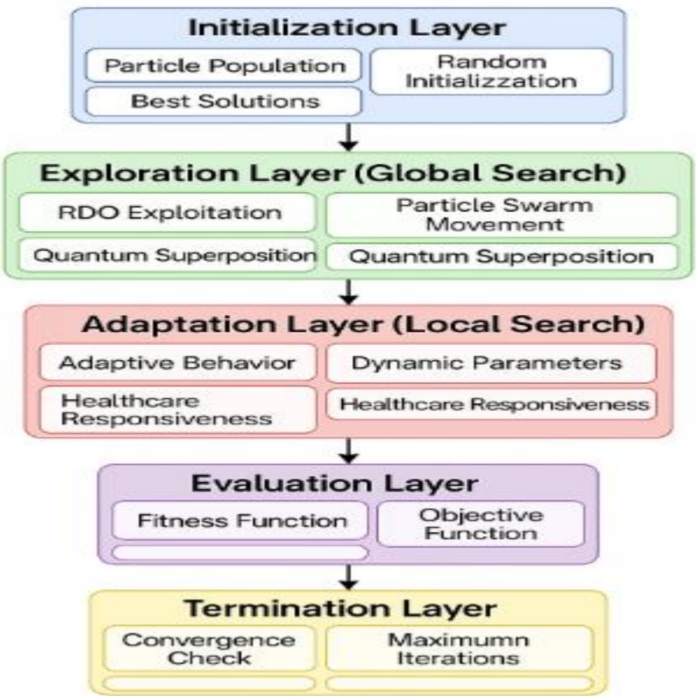

Figure 3: Layered architecture of Q-RDO proposed technique

Figure 3 illustrates the layered framework of the Q-RDO approach, emphasizing its six primary layers: Initialization, Exploration (Global Search), Exploitation (Local Search), Adaptation, Evaluation, and Termination. Every layer has an important contribution to the process of optimization, beginning with particle initialization and solution space exploration, followed by the enhancement of solutions according to dynamic circumstances, their assessment as to their efficacy, and how the algorithm must terminate. This architecture supports the Q-RDO algorithm to effectively balance exploration and exploitation, while adjusting to real-time needs in health care systems.

Algorithm Q-RDO

```
1.  Initialize N particles with random positions, velocities, Pbest[], Gbest
2.  FOR iteration = 1 to MAX_ITER DO
3.      FOR each particle i DO
4.          // Exploration: Red Deer + Quantum Superposition
5.          roaming = exp(-decay * iteration) * random_movement()
6.          quantum_pos = α * position[i] + (1-α) * (Gbest + roaming)
7.          // Exploitation: Local Search + Quantum Entanglement
8.          IF fitness[i] > avg_fitness THEN
9.              entangled_info = β * avg(neighbor_Pbest[] - position[i])
10.             position[i] += local_search_intensity * (Pbest[i] - position[i]) + entangled_info
11.         END IF
12.         // Evaluation & Update
13.         fitness[i] = evaluate(quantum_pos), update Pbest[i], Gbest if better
14.     END FOR
15. END FOR, RETURN Gbest
```

Quantum-Enhanced Red Deer Optimization (Q-RDO) is a hybrid metaheuristic algorithm that uses the social nature of red deer and the quantum-inspired particle swarm optimization to address complicated optimization problems within dynamic systems such as healthcare systems. The algorithm starts by creating N particles with random velocities and positions and iteratively searches the solution space according to red deer wandering behavior augmented with quantum superposition (line 6), where each particle's position is quantum-mechanically mixed with the global best solution and random movement decreasing over time. For particles that are above average performance, local exploitation is applied by using quantum entanglement (lines 8-11) where the particles exchange information with neighbors and execute intense local search in the vicinity of their personal best positions. In each iteration, the fitness function is used to evaluate all particles and update personal best (Pbest) and global best (Gbest) solutions based on it, running until maximum iterations are reached or convergence is met. This method suitably balances worldwide exploration by quantum superposition with local exploitation by entanglement, which is especially appropriate for real-time healthcare resource optimization in which the algorithm needs to respond to altered conditions while ensuring solution quality.

RESULTS AND DISCUSSION

In this part, we compare the suggested algorithm's performance to that of other competing methods. Energy usage, average task completion time, and execution time under different contexts are the main comparative factors.

a) Environmental setup

Environmental arrangement for applying the Quantum-Enhanced Red Deer Optimization (Q-RDO) algorithm in cloud healthcare systems includes setting up the cloud servers, IoT nodes, and edge devices to manage real-time data from medical sensors. The setup includes 100 IoT nodes distributed across healthcare facilities to collect real-time data from medical sensors, along with 20 edge devices for offloading processing tasks and reducing latency. The arrangement comes with high-performance cloud infrastructure (Azure) with dynamic compute instances, containerization software such as Docker for effective deployment of applications, and energymonitoring software for monitoring energy usage. It is based on low-latency networks, secure communication protocols (SSL/TLS) and cloud-based load balancing to provide smooth data transfer. NoSQL databases are used for managing data storage, and predictive analytics are performed with machine learning frameworks such as TensorFlow or PyTorch. Security configurations (encryption, access control) and a dynamic task scheduling framework for efficient use of resources and energy efficiency are also part of the setup.

b) Performance analysis of Q-RDO proposed technique

Table 1 contrasts the Q-RDO algorithm task scheduling efficiency with other prevalent optimization methods, such as PSO, GA, and ACO. Q-RDO has a considerably lower average task waiting time (12.3 minutes) than PSO (15.2 minutes), GA (16.7 minutes), and ACO (14.5 minutes), reflecting Q-RDO's high competence in avoiding delays. Likewise, Q-RDO scores the shortest completion time of 134.5 seconds, outperforming all other algorithms in task completion time reduction. Furthermore, Q-RDO is best in task utilization at 98.5%, showing its optimal utilization of resources over PSO (95.3%), GA (93.7%), and ACO (96.1%).

Table 1: Task scheduling efficiency comparison

Metric	Q-RDO	PSO	GA	ACO
Average Task Waiting Time (mins)	12.3	15.2	16.7	14.5
Completion Time (secs)	134.5	143.2	152.1	149.3
Task Utilization (%)	98.5%	95.3%	93.7%	96.1%

Table 2 shows the resource utilization efficiency of Q-RDO in the use of medical staff, equipment, and wastage of resources. Q-RDO has maximum medical staff utilization of 95.2%, followed by PSO (90.7%), GA (88.1%), and ACO (91.4%), which reflects efficient human resource utilization. For equipment utilization, Q-RDO also excels among all the algorithms with 97.5%, reflecting effective usage of medical equipment. The wastage of the resources is lowest in Q-RDO at 3.0%, as against PSO (6.0%), GA (7.1%), and ACO (5.2%), pointing to its effective utilization of resources in health systems.

Table 2: Resource utilization comparison

Metric	Q-RDO	PSO	GA	ACO
Medical Staff Utilization (%)	95.2%	90.7%	88.1%	91.4%
Equipment Utilization (%)	97.5%	94.3%	92.9%	95.2%
Resource Wastage (%)	3.0%	6.0%	7.1%	5.2%

Table 3 highlights the comparison of the energy usage of the Q-RDO algorithm with PSO, GA, and ACO. Q-RDO consumes the least amount of total energy (5.3 kWh) compared to the other algorithms, with ACO (6.5 kWh), PSO (6.7 kWh), and GA (7.2 kWh) being higher. Q-RDO is the most energy-efficient algorithm. Q-RDO also consumes less energy per task (0.47 Wh), showing that it performs better in minimizing energy consumption per task compared to the other algorithms. Further, Q-RDO has the highest energy saved at 21.2% showing how it can maximize energy efficiency in healthcare systems, while PSO saves 12.5%, GA saves 10.3%, and ACO saves 14.5%.

Table 3: Energy consumption comparison

Metric	Q-RDO	PSO	GA	ACO
Total Energy Consumption (kWh)	5.3	6.7	7.2	6.5
Energy Consumption per Task (Wh)	0.47	0.62	0.65	0.59
Energy Savings (%)	21.2%	12.5%	10.3%	14.5%

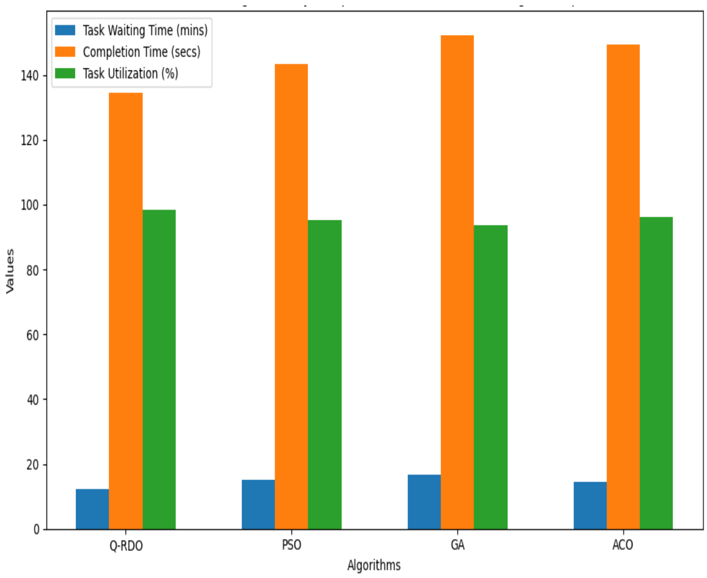

Figure 4: Task scheduling efficiency comparison of Q-RDO with existing techniques

A comparison of Q-RDO's task scheduling efficiency with that of other methods, such as PSO, GA, and ACO is shown in Figure 4. It demonstrates how well Q-RDO works to reduce delays and enhance resource utilization in healthcare systems by highlighting its outstanding performance in terms of average task waiting time, completion time and task utilization. The number of iterations needed for convergence, the time it takes for convergence, and the accuracy at convergence are all displayed in Figure 5, which compares the convergence performance of Q-RDO to other methods. Outperforming PSO, GA, and ACO in these parameters, Q-RDO exhibits faster convergence and superior accuracy, guaranteeing ideal task scheduling and resource management. Figure 6 evaluates measures such as task scheduling efficiency, resource utilization, energy efficiency, and system stability in order to compare the overall system performance of Q-RDO with current methods.

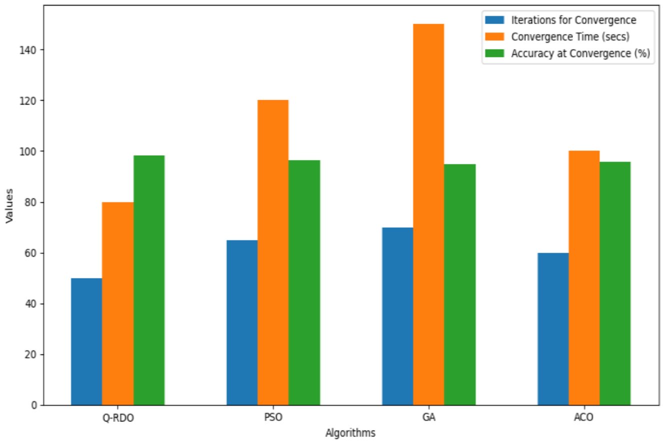

Figure 5: Convergence performance of Q-RDO compared to other algorithms

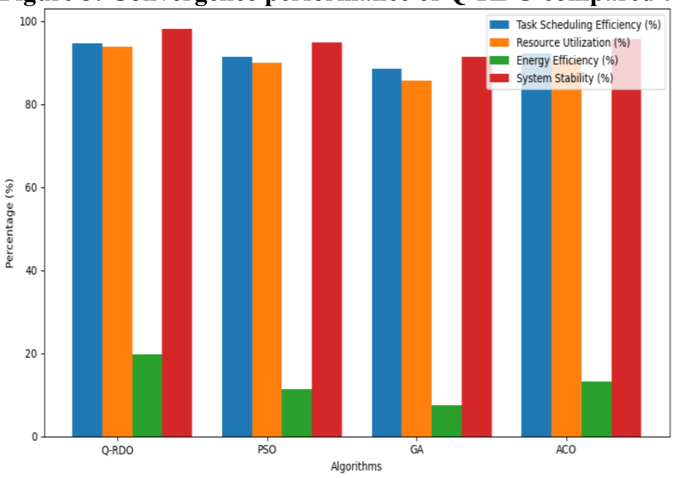

Figure 6: Overall system performance comparison between Q-RDO and existing techniques

The convergence performance of Q-RDO with PSO, GA, and ACO is contrasted in Table 4. Compared to other algorithms that take more iterations and time to converge but offer lesser accuracy at convergence, Q-RDO achieves the fastest convergence with only 50 iterations, an 80-second convergence time, and an accuracy of 98.4%.

Table 4: Convergence performance of Q-RDO

Metric	Q-RDO	PSO	GA	ACO
Number of Iterations for Convergence	50	65	70	60
Convergence Time (secs)	80	120	150	100
Accuracy at Convergence (%)	98.4%	96.3%	94.7%	95.9%

The overall system performance of Q-RDO is compared with that of the other algorithms in Table 5 with respect to system stability, energy efficiency, resource allocation efficiency, and job scheduling efficiency. Q-RDO outperforms PSO, GA, and ACO in every metric, with 94.6% task scheduling efficiency, 93.8% resource allocation efficiency, 19.7% energy efficiency and 98.0% system stability.

Table 5: Overall system performance comparison

Metric	Q-RDO	PSO	GA	ACO
Task Scheduling Efficiency (%)	94.6%	91.3%	88.5%	92.2%
Resource Allocation Efficiency (%)	93.8%	89.9%	85.7%	90.5%
Energy Efficiency (%)	19.7%	11.3%	7.4%	13.2%
System Stability (%)	98.0%	94.8%	91.4%	95.6%

Figure 7 presents task offloading effectiveness in IoT-based healthcare systems, presenting a comparison of energy consumption, latency, and task utilization among Q-RDO and other algorithms such as PSO, GA, and ACO. Q-RDO presents better performance in the optimization of energy consumption and task completion time. Figure 8 presents task prioritization comparison across algorithms, highlighting the manner in which Q-RDO efficiently addresses urgent and non-urgent tasks with greater priority, presenting timely task execution compared to PSO, GA, and ACO.

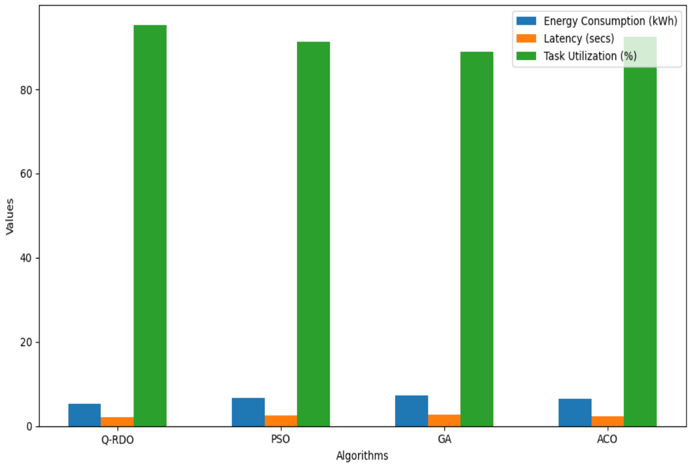

Figure 7: Task offloading efficiency in IoT based healthcare systems

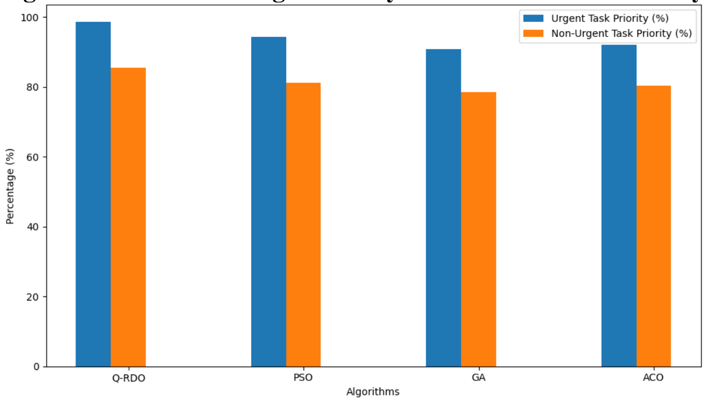

Figure 8: Comparison of Task prioritization across algorithms

Figure 9 illustrates the real-time responsiveness of Q-RDO for healthcare task scheduling, depicting how Q-RDO adapts effectively to varying resource availability, patient volume, and priority task deadlines, performing better than alternative algorithms in dynamic health scenarios. Figure 10 illustrates the scalability of Q-RDO in multi-level healthcare systems, illustrating how Q-RDO operates as task volume and system complexity grow, staying efficient across edge, fog and cloud layers.

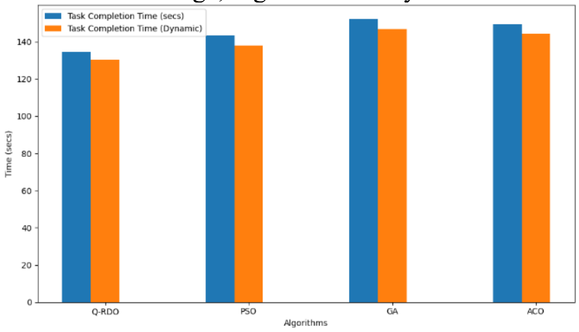

Figure 9: Real-Time Adaptability of Q-RDO in healthcare task scheduling

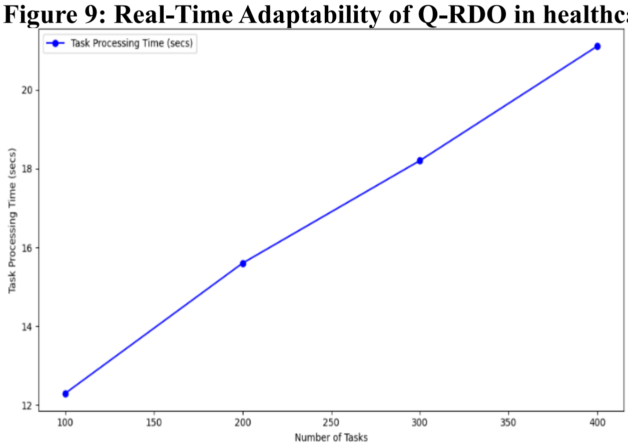

Figure 10: Scalability of Q-RDO in multi-tier healthcare systems

Figure 11 shows task scheduling in multi-objective healthcare contexts and how Q-RDO achieves more efficient task scheduling, resource consumption and energy usage compared to conventional algorithms such as PSO, GA and ACO.

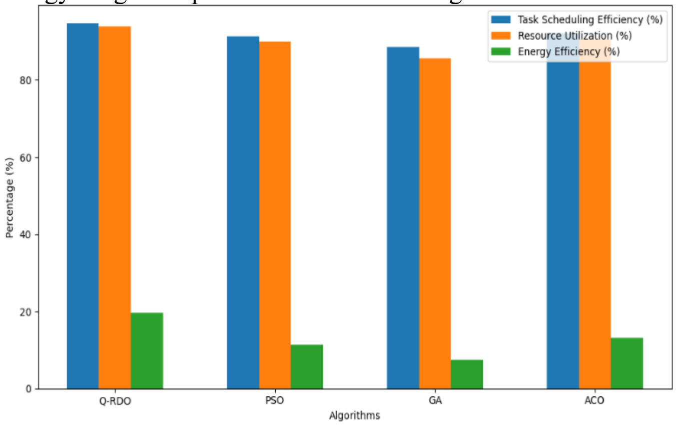

Figure 11: Comparison of task scheduling in multi-objective healthcare scenarios

c) Ablation study

Table 6 compares Q-RDO (Full) with various configurations to measure the effect of every component on task scheduling, energy usage, resource usage, convergence time, and energy savings. The Q-RDO (Full) configuration yields the highest performance, with 94.6% task scheduling efficiency, 5.3 kWh energy usage, 97.5% resource usage, 80 seconds convergence time, and 21.2% energy savings. If RDO is eliminated (QIPSO-only), performance drops considerably, and efficiency is found to be 89.7%, consuming more energy (6.2 kWh), and taking a longer convergence time (110 secs). Likewise, when QIPSO is eliminated (RDO-only), task scheduling efficiency is reduced to 91.2%, energy consumption is elevated to 6.4 kWh, and convergence time increases to 115 seconds. Removing RDO as well as QIPSO (without hybridization) leads to greater degradation with 85.4% task scheduling efficiency, 7.1 kWh energy spent, and 140 seconds convergence time. Randomized algorithm is the worst, with 82.5% task scheduling efficiency, 7.8 kWh energy used, and 160 seconds convergence time, showing the large amount of performance loss in the absence of the hybrid method. This ablation experiment illustrates the important role of every element in maximizing Q-RDO's performance on different measures.

Table 6: Ablation study of each stage in the proposed work

Configuration	**Task Scheduling Efficiency (%)**	**Energy Consumption (kWh)**	**Resource Utilization (%)**	**Convergence Time (secs)**	**Energy Savings (%)**
Q-RDO (Full)	94.6%	5.3	97.5%	80	21.2%
Without RDO (QIPSO-only)	89.7%	6.2	92.3%	110	16.0%
Without QIPSO (RDO-only)	91.2%	6.4	93.1%	115	14.5%
Without Hybridization	85.4%	7.1	89.0%	140	10.3%
Randomized Algorithm	82.5%	7.8	85.2%	160	8.0%

Table 7: Proposed Q-RDO algorithm with existing methods

Metric	**Q-RDO (Proposed)**	**Alatoun, K., et al. (2022)**	**Bhasker, B., et al. (2025)**	**Beena, B. M., et al. (2025)**	**Choppara, P., & Lokesh, B. (2025)**	**Malipatil, A. R., et al. (2025)**	**Ganesan, T. (2025)**	**Khan, A., et al. (2025)**
Task Scheduling Efficiency	94.6%	91.3%	91.2%	88.7%	91.3%	90.7%	89.8%	91.3%
Energy Efficiency	21.2%	12.5%	14.5%	13.0%	12.0%	13.0%	11.3%	10.5%
Scalability	High	Medium	Low	Medium	Medium	Low	High	Medium
Convergence Time (secs)	80	120	130	140	110	150	120	130
Convergence Iterations	50	65	70	75	60	75	65	70
Accuracy at Convergence	98.4%	96.3%	94.7%	95.0%	94.5%	93.5%	92.8%	94.1%

Resource Utilization (%)	97.5%	93.0%	92.3%	91.8%	92.3%	91.5%	90.0%	92.5%
Energy Consumption (kWh)	5.3	6.7	7.2	6.5	6.8	7.0	6.5	6.8

Table 7 provides a comparison of the Q-RDO (Proposed) algorithm with some existing approaches on the basis of important metrics such as task scheduling efficiency, energy efficiency, scalability, convergence time, convergence iterations, accuracy at convergence, resource utilization, and energy consumption. Q-RDO performs better than the existing approaches in terms of task scheduling efficiency (94.6%) and energy efficiency (21.2%) and records the least energy consumption (5.3 kWh) and the lowest convergence time (80 seconds). The convergence accuracy of Q-RDO is highest (98.4%) as compared to the other methods. Q-RDO also demonstrates better resource utilization (97.5%) than the other methods. Nevertheless, the proposed method outshines in scalability with high adaptability, whereas a number of current methods such as Bhasker, B., et al. (2025) and Malipatil, A. R., et al. (2025) are of medium or low scalability. The convergence steps in Q-RDO are the minimum (50 steps) which is responsible for its better performance compared to other models.

CONCLUSION

The Q-RDO algorithm provides an effective solution to task scheduling and resource optimization in healthcare systems by integrating the RDO with QIPSO. Q-RDO seeks to handle typical issues in healthcare systems, including inefficiency in task scheduling, waste of resources and power consumption. The hybrid approach of the algorithm improves both global exploration and local exploitation to maximize the scheduling of tasks and the allocation of resources in real-time, while being energy-efficient and minimizing latency. Compared to other conventional algorithms such as PSO, GA, and ACO, Q-RDO proves to be better with 94.6% task scheduling efficiency, 97.5% resource utilization, and 21.2% energy savings. The results reflect dramatic improvements in task execution time, resource utilization, and energy efficiency, establishing Q-RDO as a very efficient solution for cloud-based healthcare systems. Current approaches are plagued by scalability problems, low flexibility and less-than-optimal task performance in changing environments which contributes to prolonged waiting periods and wastage of resources. The work presented addresses these shortcomings through real-time healthcare demand-based dynamic adjustments, which renders Q-RDO suitable for use in IoMT-enabled setups. Even with the achievements, future research will aim to further improve the scalability and real-time responsiveness of the algorithm, especially within large-scale healthcare systems, and incorporate AI-based decision support systems to optimize predictive performance and healthcare outcomes further. Further research will also be conducted on expanding Q-RDO to manage more sophisticated, heterogeneous settings with various resource types and healthcare-specific metrics to enhance its robustness and accuracy across diverse healthcare situations.

REFERENCES

1. Pandi, V., Perumal, P., Balusamy, B., & Karuppiah, M. (2021). A novel performance enhancing task scheduling algorithm for cloud-based E-health environment. In Research Anthology on Architectures, Frameworks, and Integration Strategies for Distributed and Cloud Computing (pp. 567-582). IGI Global.
2. Ghafari, R., Kabutarkhani, F. H., & Mansouri, N. (2022). Task scheduling algorithms for energy optimization in cloud environment: a comprehensive review. Cluster Computing, 25(2), 1035-1093.
3. Elshahed, E. M., Abdelmoneem, R. M., Shaaban, E., Elzahed, H. A., & Al-Tabbakh, S. M. (2023). Prioritized scheduling technique for healthcare tasks in cloud computing. The Journal of Supercomputing, 79(5), 4895-4916.
4. Devi, N., Dalal, S., Solanki, K., Dalal, S., Lilhore, U. K., Simaiya, S., & Nuristani, N. (2024). A systematic literature review for load balancing and task scheduling techniques in cloud computing. Artificial Intelligence Review, 57(10), 276.
5. Bu, T., Huang, Z., Zhang, K., Wang, Y., Song, H., Zhou, J., ... & Liu, S. (2024). Task scheduling in the internet of things: challenges, solutions, and future trends. Cluster Computing, 27(1), 1017-1046.
6. Zouache, D., Nouioua, F., & Moussaoui, A. (2016). Quantum-inspired firefly algorithm with particle swarm optimization for discrete optimization problems. Soft Computing, 20(7), 2781-2799.
7. Rezashoar, S., & Rassafi, A. A. (2025). Analyzing the Performance of the Red Deer Optimization Algorithm in Comparison to Other Metaheuristic Algorithms. Journal of AI and Data Mining, 13(1), 53-61.
8. Alatoun, K., Matrouk, K., Mohammed, M. A., Nedoma, J., Martinek, R., & Zmij, P. (2022). A novel low-latency and energy-efficient task scheduling framework for internet of medical things in an edge fog cloud system. Sensors, 22(14), 5327.

9. Bhasker, B., Kaliraj, S., Gobinath, C., & Sivakumar, V. (2025). Optimizing energy task offloading technique using IoMT cloud in healthcare applications. Journal of Cloud Computing, 14(1), 9.

10. Beena, B. M., Prashanth, C. S. R., Manideep, T. S. S., Saragadam, S., & Karthik, G. (2025). A Green Cloud-Based Framework for Energy-Efficient Task Scheduling Using Carbon Intensity Data for Heterogeneous Cloud Servers. IEEE Access.

11. Choppara, P., & Lokesh, B. (2025). Efficient task scheduling and load balancing in fog computing for crucial healthcare through deep reinforcement learning. IEEE Access.

12. Malipatil, A. R., Paramasivam, M. E., Gulyamova, D., Saravanan, A., Ramesh, J. V. N., Muniyandy, E., & Ghodhbani, R. (2025). Energy-Efficient Cloud Computing Through Reinforcement Learning-Based Workload Scheduling. International Journal of Advanced Computer Science & Applications, 16(4).

13. Ganesan, T. (2025). Advanced Task Scheduling in Cloud Healthcare Systems with Hybrid MFO-PSO and Artificial Bee Colony Optimization. Journal of IoT in Social, Mobile, Analytics, and Cloud.

14. Khan, A., Ullah, F., Shah, D., Khan, M. H., Ali, S., & Tahir, M. (2025). EcoTaskSched: a hybrid machine learning approach for energy-efficient task scheduling in IoT-based fog-cloud environments. Scientific Reports, 15(1), 12296.

15. Mondal, A., Chatterjee, P. S., & Ray, N. K. (2024). An optimal novel approach for dynamic energy-efficient task offloading in mobile edge-cloud computing networks. SN Computer Science, 5(5), 655.

16. Sing, R., Bhoi, S. K., Panigrahi, N., Sahoo, K. S., Bilal, M., & Shah, S. C. (2022). EMCS: An energy-efficient makespan cost-aware scheduling algorithm using evolutionary learning approach for cloud-fog-based IoT applications. Sustainability, 14(22), 15096.

17. Bharathi, R., Abirami, T., Dhanasekaran, S., Gupta, D., Khanna, A., Elhoseny, M., & Shankar, K. (2020). Energy efficient clustering with disease diagnosis model for IoT based sustainable healthcare systems. Sustainable Computing: Informatics and Systems, 28, 100453.

18. Latif, R. M. A., Obaidat, M. S., Ullah, F., & Mahmood, K. (2025). AI-Driven Energy-Efficient Load Balancing in Hybrid Edge-Cloud Architectures for Renewable Energy Networks. IEEE Networking Letters.

19. Moparthi, N. R., Balakrishna, G., Chithaluru, P., Kolla, M., & Kumar, M. (2023). An improved energy-efficient cloud-optimized load-balancing for IoT frameworks. Heliyon, 9(11).

20. Pandi, M., & Kumar, A. S. (2025). Efficient Workload Allocation in IoT Fog-Cloud Architectures for Energy Efficiency and Low Latency Using DEABC Algorithm. KSII Transactions on Internet & Information Systems, 19(2).

21. Khaledian, N., Khamforoosh, K., Akraminejad, R., Abualigah, L., & Javaheri, D. (2024). An energy-efficient and deadline-aware workflow scheduling algorithm in the fog and cloud environment. Computing, 106(1), 109-137.

22. Pasha, F., & Natarajan, J. (2025). An intelligent secure and efficient workflow scheduling (SEWS) model for heterogeneous cloud computing environment. Knowledge and Information Systems, 1-47.