International Journal of Computer Applications (0975 - 8887)
Volume 187 - No.61, December 2025

Multi-Objective Clustering and Reinforcement-based Routing in IoT Networks

Moez Elarfaoui
University of Tunis
SMART LAB, ISG
Bardo, Tunisia

Hamdi Ouechtati
University of Tunis
SMART LAB, ISG
Bardo, Tunisia

Nadia Ben Azzouna
University of Tunis
ESSECT
Monfleury, Tunisia

ABSTRACT

The rapid development of devices on the Internet of Things (IoT) and the diversity of their applications have made them ubiquitous. However, deploying these devices in large-scale networks presents several challenges, including limited energy capacity, security concerns, unreliable links, and transmission delays. This paper, proposes a multi-objective optimization approach for wireless IoT networks based on machine learning techniques. Specifically, a clustering scheme is developd by using an improved k-means algorithm. This is combined with a dynamic routing strategy based on multi-objective Q-learning using parallel Q-tables. This approach leads to measurable gains in energy efficiency, transmission latency, and reliability. Compared to existing approaches in similar contexts, such as weighted sum, the proposed solution achieves significant improvements in overall network performance.

General Terms

Clustering , Reinforcement Learning, Multi-Objective Decision

Keywords

Machine learning, clustering, Q-learning, IoT, multi-objective, reliability, energy.

1. INTRODUCTION

The rapid proliferation and diversification of IoT devices have made them an integral part of modern networks [1]. Although this widespread adoption offers great opportunities, it also introduces significant challenges. These include network congestion, security vulnerabilities, increased energy consumption, and the need to satisfy stringent Quality of Service (QoS) requirements [2]. Most IoT devices are battery-powered, which limits their battery life and requires them to operate in an energy-efficient manner.

One technique commonly used to address these challenges is clustering, which groups nodes into clusters. Each cluster is managed by a specific node known as the Cluster Head (CH). The Cluster Head collects data from the member nodes, aggregates it, and transmits it to a base station or server for centralized decision making [3]. The main source of energy consumption in an IoT device is its radio module [4]. Therefore, optimizing the routing process in an IoT network is essential to extend its operational lifetime. However, optimizing the energy consumption of network nodes can come at the cost of other important metrics such as transmission delay and path reliability. Therefore, effective operation of such a network often requires a multi-objective optimization approach.

Artificial intelligence techniques have been widely adopted to address the challenges within IoT networks, including nature-inspired metaheuristics, fuzzy logic, and machine learning methods (supervised, unsupervised, and reinforcement learning). Machine learning, and particularly reinforcement learning, is well suited for IoT network routing problems, as it takes into account the dynamic nature of the environment. This study makes three main contributions:

—Designing of a multi-objective clustering scheme.

—The design of a multi-objective, multi-hop routing protocol.

—Integration of reinforcement learning for Cluster Head election and next-hop selection.

2. PRELIMINARY

Conventional IoT and WSN routing systems face several well-known challenges, as highlighted in numerous studies. Recurring issues include high energy consumption, limited scalability, and poor adaptability to dynamic environments [5], which underlines the need for innovative solutions. Recent research has introduced Machine Learning (ML) as a potentially effective paradigm to address traditional routing problems. Supervised, unsupervised, and reinforcement learning, along with neural networks and other ML approaches, allow routing decisions to be adapted in real-time based on the state of the network [6]. Many IoT-based applications, including collaborative communications, routing, and flow control, use reinforcement learning (RL) to help sensors and nodes operate efficiently in dynamic environments. This learning paradigm, often described as ”trial and error”, enables knowledge acquisition through interaction with the environment, rather than through predefined models. Reinforcement learning is particularly attractive because of its ease of implementation, adaptability, and low resource requirements (in terms of memory and processing power), while offering a wide range of applications. For example, reinforcement learning in WSNs has been applied to clustering, media access control, and routing protocols [5].

2.1 Principle of reinforcement learning

The reinforcement learning paradigm is based on the concept that intelligent systems can learn through trial-and-error interactions with their environment. This idea is especially powerful when applied to the adaptive control of complex and dynamic systems, as illustrated in **Figure 1**, which depicts the fundamental interaction between the agent and the environment through actions, states, and rewards.

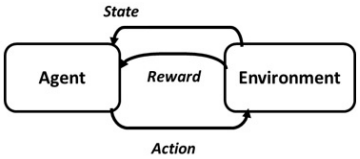

Fig. 1. Principle of reinforcement learning

Formally, reinforcement learning can be defined as a Markov Decision Process (MDP) [7]. This process provides a flexible framework for goal-oriented learning and is described by the tuple: $M = (S, A, P(S_{t+1}, r|s, a), R, \gamma)$, where:

—S: the set of all possible states of the environment;

—A: the set of actions available to the agent;

—P: the transition probability of moving to state s^{t+1} and receiving reward r, given the current state s and action a;

—$R \in \mathbb{R}$: the reward function, R(s, a), representing the expected return for taking action a in state s;

—$\gamma \in [0, 1]$: the discount factor that determines the importance of future rewards.

2.2 Multi-objective problem

Multi-objective decision-making, also known as multi-criteria optimization, has long been a key topic in real-world applications [8]. This is particularly relevant in IoT and WSN environments, where several conflicting objectives must be considered simultaneously, such as energy consumption, transmission delay, throughput, and others. The literature discusses a variety of techniques to address multi-objective optimization problems. Table 1 provides a non-exhaustive list of these methods [9], including a priori and a posteriori approaches.

Table 1. Multi-objective methods

Preferences	**Methods**
Priori	Lexicographic Method, Goal Programming
Posteriori	lignes Weighting Method, ε-Constraint Method, Normal Boundary Intersection, Order of Preference by Similarity to Ideal Solution
Interactive	Probabilistic Trade-Off Development Method, Sequential Multi-objective Problem Solving Method, Interactive Surrogate Worth Trade-Off Method (ISWT), Tchebycheff Method
No preferences	Global Criterion Method

2.3 Multi-objective reinforcement learning

When reinforcement learning is applied to multi-objective contexts, the reward function can be designed to reflect multiple performance criteria. This variant is known as Multi-Objective Reinforcement Learning (MORL). For instance, in a routing process, the selection of the next hop may consider several criteria, such as the reliability of the chosen node, its remaining energy, and the estimated transmission delay.

3. RELATED WORKS

Several recent studies have explored the use of reinforcement learning and its variants in routing and resource allocation for IoT and WSNs. Current surveys indicate the emergence of Federated Reinforcement Learning (FRL) as a privacy-preserving and distributed approach to IoT applications, particularly for energy saving and security applications [10]. In parallel, Q-learning-based and multi-objective routing techniques have been proposed to select the cluster heads and optimize the routes with significant enhancement in energy efficiency and latency by adopting locally learned policies [11], [12]. Besides, multi-agent and deep reinforcement learning techniques have also shown satisfactory performance in cooperative routing as well as handling dynamic network topologies [13], [14]. Explainability of RL-based network control has also become a concern, and recent efforts such as XGate attempt to improve the explainability and transparency of RL-based decisions for sensor networks [15]. In addition, multi-objective reinforcement learning (MORL) and online Pareto-approximation techniques have also been proposed to resolve trade-offs among conflicting objectives under real-time constraints, again demonstrating the importance of Pareto-based decision-making strategies in IoT routing settings [16]. These breakthroughs collectively emphasize the need for integrating multi-objective, explainable, and federated RL paradigms to build scalable, intelligent, and robust routing solutions for next-generation IoT networks. In [17], Chai and Zeng proposed a multi-objective routing protocol based on the Dyna-Q algorithm in wireless mesh networks to improve both delay and energy performance. Each objective is handled independently, and each node maintains two separate Q-tables, one for delay and one for energy. Chebyshev distance is employed to select the best-performing action for each objective. In [18], Prabhu et al. introduced a multi-agent reinforcement learning approach for energy-efficient routing in WSNs. They focused on two objectives: transmission delay and reliability. The authors proposed three reliable reward functions for Q-value computation, based on energy levels along the route, packet delivery ratio (PDR), and packet consistency (PC).

In [19], Xing Su et al. developed a Q-learning-based routing strategy for energy-efficient data transmission in flat WSNs. Their reward function incorporates five components: the distance between two nodes (si, sj), the remaining energy of node sj after transmission, the energy consumed by sj for transmission, and the nature of the transmission action. The final reward is computed as a weighted sum.

In [20], Godfrey et al. proposed an energy-efficient multi-objective routing protocol using a reinforcement learning algorithm with Dynamic Objective Selection (DOS-RL) to optimize energy consumption in IoT networks. The approach considers correlated objectives and uses informative-shaped reward functions to accelerate learning. Three reward functions are defined, corresponding to data forwarding energy, load balancing, and link

quality. DOS selects actions based on the estimated Q-values of the most confident objective.
In [21], Vaishnav et al. proposed a dynamic and distributed routing approach for IoT networks using multi-objective Q-learning. They targeted two objectives: energy consumption and packet delivery ratio (PDR). The proposed method, Dynamic Preference Routing Q-learning, assigns a Q-table to each preference parameter, allowing Q-learning to train all Q-functions in parallel. This approach ensures adaptability and efficient handling of shifting objectives.

4. PROPOSED APPROACH

The current approach employs the Machine Learning (ML) paradigm to optimize the Quality of Service (QoS) in IoT networks. These networks involve multiple, often conflicting objectives. To address this, we first partition the network's nodes into clusters—a technique known to extend network lifetime. Clustering is performed using an improved K-means algorithm that incorporates a multi-objective technique for Cluster Head (CH) election. Subsequently, multi-objective reinforcement learning is used to route packets from source nodes to the base station (BS). In both stages, Pareto optimality is employed to find a trade-off among competing objectives. Assumptions All nodes are homogeneous, meaning they share the same technical characteristics (computing power, storage capacity, and energy). Nodes are stationary (non-mobile). The base station knows the geographic coordinates of all nodes. Each node can adjust its transmission range as needed.

4.1 Energy model

The energy dissipated to send k bits from a node n_i to a node n_j over a distance $d(n_i, n_j)$ is given by the equation (1) [22].

$$E_{TX} = \begin{cases} E_{elec} * k + E_{fs} * k * d^2, if d \leq d0 \\ E_{elec} * k + E_{mp} * k * d^4, if d > d0 \end{cases} \quad (1)$$

where E_{elec} is electronic energy that counts on the filtering, modulation the digital coding and spreading of the signal;
E_{fs} and E_{mp} are the amplification energy in the free space and multi-path fading careers, respectively and d_0 is a threshold distance.

$$d_0 = \sqrt{\frac{E_{fs}}{E_{mp}}}$$

The consumed energy by a node nj to receive a packet of k bits is depicted by the equation (2).

$$E_{RX} = E_{elec} * k \quad (2)$$

4.2 Solution details

4.2.1 Clustering phase.

K-means [23, 24] is one of the unsupervised machine learning algorithms that aims to create a clustering scheme based on similarities between individuals. This algorithm was improved to takes into account the multi-objective approach to select cluster head (CH).
Once clusters are formed, the selection of the CH begins. This procedure takes into account three objectives or (criteria) which are the residual energy, the distance to the BS, and the frequency of the previous election as a CH node. These objectives are defined as:
• E_{Resid} : the residual energy of the node ni; to be maximized;
• $d(n_i, BS)$: distance from a node ni to BS; to be minimized;
• F_{req} : frequence of the previous election of node i; to be minimized.
The election of the CH needs to minimize the three objectives. So, to have synergy between these objectives, we take the inverse of E_{Resid}.
Each node in the cluster is described by a vector, $n_i = [Ci_1, Ci_2, Ci_3]$, with three components which represent the three objectives.
In this work Pareto optimality was adopted to choose the optimal CH node. To achieve this, the Skyline algorithm was used. [25, 26] to efficiently identify the dominant nodes. The pseudo code in Algorithm 1 explains this idea.

Algorithm 1 Skyline SDI calculation

Input: DataMatrix: matrix of candidate nodes
Output: Skyline: set of non-dominated nodes

```
Sort DataMatrix by the first objective column
// Initialize Skyline
Skyline ← first row of DataMatrix
for i ← 2 to length(DataMatrix) do
    row ← DataMatrix[i]
    Dom ← false
    for j ← 1 to length(Skyline) do
        Sky ← Skyline[j]
        if dominates(row, Sky) then
            remove Sky from Skyline
            else if dominates(Sky, row) then
                Dom ← true; break
            end
        end
    end
    if Dom = false then
        Add row to Skyline
    end
end
```

4.2.2 Routing phase.

a) *Path discovery*

Path discovery in a wireless IoT network is the process of finding one or more routes between a source node and the destination (BS). This step is crucial to ensuring the efficient and reliable transmission of data. This phase is initiated by the BS. After the clustering procedure, the BS sends a Hello message to signal the nearest CH nodes. A node, which receives this message, updates the number of hops to the BS and sends a hello message to its direct neighbors by adding its number of hops to the BS, the quantity of residual energy, and the distance that separates it from the BS. This procedure continues until the node at the end of the fields. At the end of this phase, each CH node initializes its neighbors table. It should be noted among the information on neighbors, there are parameters which are static (distance, number of hops) and others which are dynamic (residual energy, state of the queue message). The overall process is illustrated in **Figure 2**.

b) *Routing through Q-Learning*

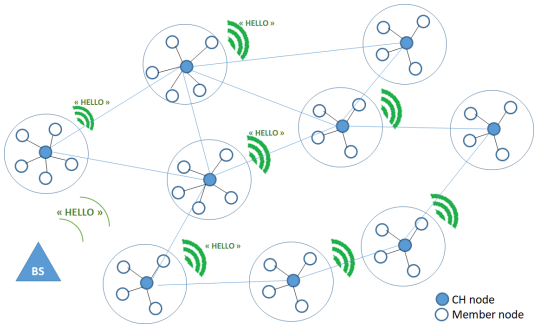

Fig. 2. Path discovery mechanism

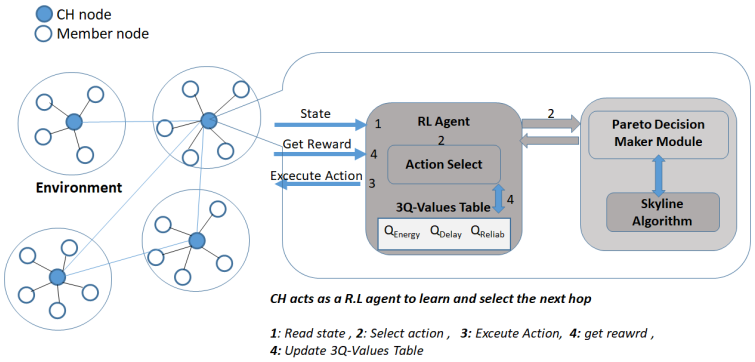

Fig. 3. Overview of the Proposed Multi-Objective Q-Learning Routing Model in Clustered IoT/WSN Networks

In this phase, the study is limited to the inter cluster routing. To do so,all Cluster Heads (CHs)are used to route packets from the CH source to the base station (BS) in a multi-hop manner. To perform this procedure efficiently, Multi-Objective Q-Learning is used to learn an optimal policy that takes into account the dynamic state of the network. It aims to optimize three objectives. These objectives correspond to the energy consumption along a path from a node CH to the BS, the transmission delay from node to the BS and the link reliability. Q-learning is a model-free algorithm that learns to make better decisions by constantly interacting with the environment, selecting actions, observing rewards and updating the Q-Values in the Q-Table according to the action value function [20, 21]. The Q-Table is then updated using equation (3).

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma max_{a'} Q(s',a') - Q(s,a)) \quad (3)$$

where

—s: the current state (s ∈ S), S the set of possible states of the environment;

—a: the selected action (a ∈ A), A the set of possible actions;

—a': next possible action (a ∈ A);

—Q(s, a): the value of action a in state s;

—α: learning rate;

—γ: discount factor, which balances immediate and future reward;

—r: the reward obtained after an action.

As **Figure 3** illustrates, the overall architecture of the proposed Multi-Objective Q-Learning (MOQL) routing model is designed for clustered IoT and Wireless Sensor Networks (WSNs). The model integrates the principles of reinforcement learning and multi-objective optimization to achieve energy efficiency, low latency, and high reliability in data transmission. On the left side, the environment is represented by multiple clusters of sensor nodes, each containing several low-power IoT devices connected to a central node called the Cluster Head (CH). Each CH aggregates data from its local members and participates in inter-cluster communication toward the Base Station (BS) through multi-hop routing. The clustering structure illustrates how the WSN topology supports hierarchical organization and localized decision-making.

At the center, one Cluster Head (CH) acts as the Agent in the reinforcement learning process. The agent interacts dynamically with its environment by:

- Observing network states such as residual energy, neighbor distance, and queue statu;
- Selecting actions corresponding to the choice of the next-hop CH for packet forwarding;
- Receiving rewards that quantify the effectiveness of each action based on multiple performance criteria.

The Q-learning mechanism is implemented through three parallel Q-tables: Each Q-table is updated independently after each learning iteration using the Q-update rule. The Q-update process integrates both the immediate reward and the estimated future reward, allowing the CH to gradually converge toward optimal routing policies. On the right side, the Skyline Algorithm (based on Pareto optimality) evaluates the set of updated Q-values and selects the best trade-off action among all competing objectives. Instead of relying on a weighted-sum method (which depends on predefined preferences), the Skyline-based selection automatically identifies non-dominated solutions, ensuring an adaptive and fair balance between energy consumption, delay, and reliability. Finally, the directed arrows illustrate the logical flow of interactions:

From Environment → Agent (CH) → Q-learning process → Skyline Algorithm → back to Environment through action execution. This continuous feedback loop enables the system to learn optimal multi-hop routes over time, adapting to the dynamic network conditions of IoT/WSN environment. The update adjusts the Q-value by incorporating both the immediate reward and the maximum estimated future reward from the next state. The parameter α is the learning rate, it plays a critical role in determining how quickly or accurately the Q-learning algorithm converges to the optimal Q-values. To better understand the use of Q-learning for routing in IoT networks, we recall the key concepts introduced above.

s represents a CH node which is holding the data packet, a corresponds to selected node (next CH) when we are in state s and a' means the possible next action where we become in the s' state.

Observing network states

When a CH wants to send a data packet, it observes network states according to the Q-table to choose the suitable action. The action is selected through the policy of ε-greedy to balance between exploration and exploitation. Once this is done, we execute the action. Execute action means sending data packet to the next node. After sending the data packet, the sender receives the immediate reward. This reward reflects how good or bad the chosen action is.

Action selection

The action is selected with ε-greedy policy to obtain a trade-off between exploration and exploitation. The principle is given by the pseudo-code in Algorithm 2.

Algorithm 2 Action Selection using ε-Greedy Policy

Input: $QTable$: Q-value table, $currentState$: current node, ε: exploration rate
Output: $action$: selected action

$r \leftarrow \texttt{Random}(0,1)$
if $r < \varepsilon$ **then**
  $action \leftarrow \texttt{RandomAction}(QTable)$
  `// Exploration`
end
else
  $action \leftarrow \texttt{argmax}(QTable[currentState])$
  `// Exploitation`
end
return $action$

Reward computing

The immediate reward r is computed based on three components:
R_{Egy} : Energy consumed by the sender node.
R_{Dly} : Estimated delay to transmit the data packet to the BS through the selected hop.
R_{Rbly}: Link reliability between the sender and the next node.

Energy Component

The consumed energy to transmit bits between tow nodes, n_i and n_j,is given by the following equation.
$R_{Egy} = E_{cons}(n_i) + E_{cons}(n_j) = E_{TXi} + E_{RXj}$
Note that E_{cons} depends only on the distance. In other word, we must choose the closest one among the neighbouring nodes.

Delay Component

The delay depends on the speed, the distance between two nodes and the length of the message's queue.

$$R_{Delay} = \frac{d}{speed} + D_{Queue} \tag{4}$$

where:

—d: Distance between the two nodes;

—D_{Queue} : average delay in the message's queue.

Reliability Component
Link reliability considers both the residual energy and the queue status. A node with low residual energy or a saturated queue is likely to drop packets. Thus, reliability is defined as:

$$R_{Rbly} = exp(\frac{-E_{Resid}}{E_{init}}) * exp(\frac{-L_{Queue}}{Max_{Queue}}) \tag{5}$$

where:
E_{Resid}: The residual energy of the next node.
E_{init}: The initial energy of the next node.
L_{Queue}: The length of the message's queue of the next node.
Max_{Queue}: The maximum length of the message's queue of the next node.
The values of these components should be normalized in [0,1] to have more signification. To compute the value of reward r, the weighted sum is the most used some for its simplicity but its value is not always significant because it depends on the choice of the weights that influenced by the preferences of the Decision Maker (DM).
To achieve a trade-off among the three objectives, we use the standard deviation (STD). The ideal value of standard division is zero that means all objectives values are equal.

$$r = \begin{cases} 2, if Action = BS \\ \frac{1}{STD+1}, if Action = reachablenode \\ -2, if Action = unreachablenode \end{cases} \tag{6}$$

The standard deviation is a meaningful choice because it provides insight into the trade-off of the three objectives.
The penalty of -2 means that the node is far (out of transmission range) or failed. In this case, the packet is lost.

Q-values Updating

Considrering the equation (3), the updating of Q-Value use the $max_{a'}Q(s',a')$ which is the maximum Q-value of the next state s'. Since a multi-objective Q-learning is used, a Q-value is associated for each objective. So the Q-value is not a scalar but a vector of three Q-Values.
$Q(s,a) = [Q(s,a)_{O1}, Q(s,a)_{O2}, Q(s,a)_{O3}]$. where:
$Q(s,a)_{O1}$: Q-value related to the first objective.
$Q(s,a)_{O2}$: Q-value related to the second objective.
$Q(s,a)_{O3}$: Q-value related to the third objective.
This means that all objectives are computed in parallel. In this case, each node should keep Q-table for each objective or a Q-table with three Q-Value. This concept is illustrated in Table 2. Since t the Q-values are vectors representing multiple objectives, the standard scalar max operator cannot be applied directly. Instead, Pareto optimality is used via the Skyline algorithm to determine the best trade-off. This method is better than weighted sum because it does not need the intervention of decision maker.
For instance, when a CH node $s_1$selects s_2 as next hop , the corresponding Q-values for s_1 are updated as:
By reference to equation 3 $max_{a'}Q(s',a')$ corresponds to $max_Q(2,a')$.

$Q(2,a')$ = $\{Q(2,1)$,$Q(2,2)$,..., $Q(2,BS)\}$.

If CH_k has the dominant vector, reward r is calculated as:

$STD(0.7, 0.5, 0.8)$ = 0.151 and

$r = \frac{1}{STD+1}$= 0.86.

The Current Q-Values of s_1 is a vector $[0.4, 0.5, 0.6]$

For α=0.4 and γ =0.9, The update Q-Values of s_1 becomes:

$$Q_1(s1,2) = 0.4 + 0.4 * (0.86 + 0.9 * 0.7 - 0.4) = 0.45$$

$$Q_2(s1,2) = 0.5 + 0.4 * (0.86 + 0.9 * 0.5 - 0.5) = 0.45$$

$$Q_3(s1,2) = 0.6 + 0.4 * (0.86 + 0.9 * 0.8 - 0.6) = 0.60$$

Table 2. Structure of Q-table for Multi-objective Q-learning

Actions	CH1			CH2			CHk			BS
States	Q1	Q2	Q3	Q1	Q2	Q3	Q1	Q2	Q3	..
CH1	0.0	0.0	0.0	0.4	0.5	0.6	0.0	0.0	0.0	..
CH2	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.5	0.8	..
..	..	..	..	..	..	..	..	..	..	..
CHk	..	..	..	..	..	..	..	..	..	..

5. SIMULATION AND DISCUSSION

The simulation is performed on a set of 100 nodes which are randomly deployed on a square area of 100x100m. All nodes are homogeneous. The simulation environment used in this work is MATLAB 2020 and the network parameters are mentioned in Table 3. The proposed Multi-Objective Reinforcement Routing using three parallel Q-values (MORR3Q) is compared with ohter algorithms such as; Multi-Objective Reinforcment Routing Weight Sum (MORRWS) and Reinforcement Energy Based Routing (REBR). The evaluation criteria in this simulation focus on: First dead node (FDN), Last Dead Node (LDN), residual Energy and Packet Delivery Rate (PDR).

Table 3. Parameters of simulation

Network parameters	Values
Area size	100 x 100 m
Number of nodes	100
BS position	(100,100)
Data Packet size	4000 Bytes
Control Packet size	100 Bytes
Initial Energy of node	0.5 j
Transmitter/Receiver electronics	50 nj
Efs	10pj
Emp	0.0013pj
d0	80m

5.1 Results and discussion

5.1.1 First dead nodes (FDN).
The FDN metric gives an idea about the network stability. By adopting a multi-objective approach during the CH election, the network lifetime is further extended. Table 4 shows the contribution of the proposal approch compared to MORRWS and REBR in terms of FDN and LDN. The results are illustrated in **Figure 4**, which depicts both FDN and LDN metrics for the three algorithms.

Table 4. FDN and LDN for the three algorithms

Algorithm	MORR3Q	MORRWS	REBR
FDN	2109	1800	2050
LDN	3370	3328	3380

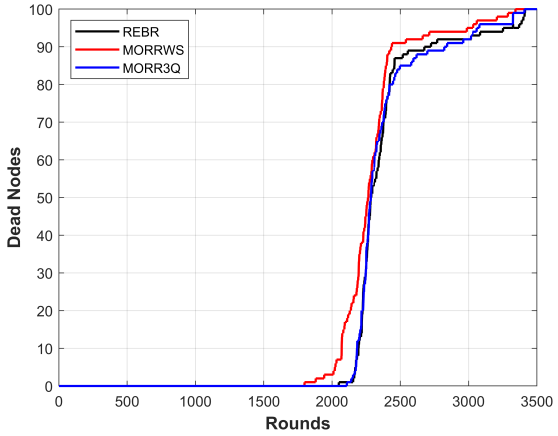

Fig. 4. FDN and LDN

5.1.2 Energy consumption.
Figure 5 shows the energy consumption in the whole network. We note here that our solution optimizes energy consumption and so extends network lifetime which is noticed by the largest value of FDN. Table 5 also reports the average energy consumption per round.

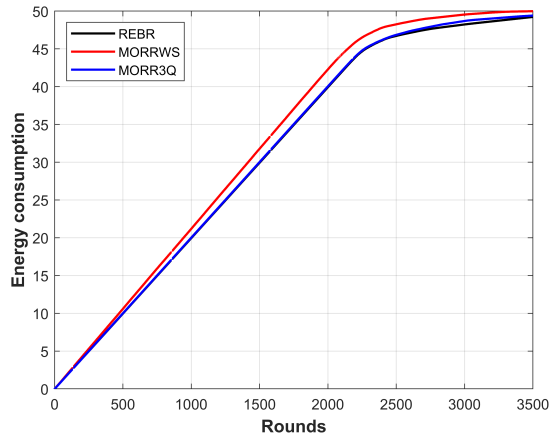

Fig. 5. Energy consumption over simulation rounds

5.1.3 Trade-off between objectives .
The trade-off between the objectives to be optimized is the major concern of any multi-objective problem. The prosed solution guarantees a compromise between delay, energy consumption and reliability. It is showen that the approach based only on the energy objective, REBR, enables low energy consumption, but on the other hand it has shortcomings in terms of packet transfer delay and reliability, which result in a lower PDR. This is expected, since energy-centric routing tends to select the closest nodes, which may compromise reliability. As a result, unreliable nodes can be encountered, often resulting in a loss of packets The weighted

sum multi-objective approach, solves this trade-off of objectives, but suffers from the choice of weights to assign to each objective. In addition, this method often requires the intervention of the decision-maker (DM). Table 5 shows that our solution, based on Pareto optimality, achieves this compromise efficiently. It allows the three objectives to be calculated in parallel on each hop, so that the next node can be selected efficiently.

Table 5. Comparison of performance metrics across approaches

Algorithm	MORR3Q	MORRWS	REBR
PDR	0.9850	0.9250	0.8674
Avrg.Delay	1.4051e-09	1.4928e-09	1.7327e-09
Avrg.EngyCons	0.0138	0.0138	0.0134

6. CONCLUSION AND PERSPECTIVE

In IoT networks, optimization is primarily concerned with energy consumption, in order to extend the network lifetime. However, other parameters also need to be optimized to guarantee the required quality of service. From this point of view, the design of a multi-objective routing solution is necessary. In this context, we have designed a multi-objective clustering and multi-objective multi-hop routing solution based on reinforcement learning using the Q-learning algorithm with three parallel Q-tables. The work focuses on optimizing energy consumption, reliability and end-to-end delay. To ensure a balanced trade-off among the three objectives, the Skyline algorithm is integrated using Pareto optimality.

As a future work, others potential QoS metrics can be considered, such as scalability and network security. Also, the proposed solution can be extended by integrating deep learning. Indeed, with modern IoT networks, which are becoming increasingly complex and address major challenges in terms of scalability, security, and dynamic adaptation, the Q-learning approach becomes ineffective due to the explosion in the size of Q-Tables. Deep learning offers analysis and modeling capabilities that make it ideal for simultaneously optimizing clustering and routing, two closely related functions in IoT networks.

7. REFERENCES

[1] N. D. Tan and V.-H. Nguyen. Machine learning meets iot: developing an energy-efficient wsn routing protocol for enhanced network longevity. *Wireless Networks*, 31(4):3127–3147, 2025.

[2] A. I. Al-Sulaifanie, B. K. Al-Sulaifanie, and S. Biswas. Recent trends in clustering algorithms for wireless sensor networks: A comprehensive review. *Computer Communications*, 191:395–424, 2022.

[3] S. El Khediri. Wireless sensor networks: a survey, categorization, main issues, and future orientations for clustering protocols. *Computing*, 104(8):1775–1837, 2022.

[4] A. Mittal, Z. Xu, and A. Shrivastava. Energy-efficient, secure, and spectrum-aware ultra-low power internet-of-things system infrastructure for precision agriculture. *IEEE Transactions on AgriFood Electronics*, 2(2):198–208, 2024.

[5] R. Priyadarshi. Exploring machine learning solutions for overcoming challenges in iot-based wireless sensor network routing: a comprehensive review. *Wireless Networks*, 30(4):2647–2673, 2024.

[6] P. M. Mwangi. A systematic literature review of routing protocols in wireless sensor networks: Current trends and future directions.

[7] J. Moos, K. Hansel, H. Abdulsamad, S. Stark, D. Clever, and J. Peters. Robust reinforcement learning: A review of foundations and recent advances. *Machine Learning and Knowledge Extraction*, 4(1):276–315, 2022.

[8] S. Sharma and V. Kumar. A comprehensive review on multi-objective optimization techniques: Past, present and future. *Archives of Computational Methods in Engineering*, 29(7):5605–5633, 2022.

[9] J. L. J. Pereira, G. A. Oliver, M. B. Francisco, S. S. Cunha Jr, and G. F. A review of multi-objective optimization: methods and algorithms in mechanical engineering problems. *Archives of Computational Methods in Engineering*, 29(4):2285–2308, 2022.

[10] A. Khan, M. Rehman, S. Zhang, and H. Kim. Federated reinforcement learning for internet-of-things applications: A survey. *IEEE Internet of Things Journal*, 10(5):4123–4142, 2023.

[11] M. Al-Shorman, R. Ahmad, and A. Ezugwu. Q-learning-based energy-efficient clustering and routing in wireless sensor networks. *IEEE Access*, 12:102345–102358, 2024.

[12] S. Ghamry and S. Shukry. Multi-Objective Intelligent Clustering Routing Scheme for IoT-Enabled WSNs Using Deep Reinforcement Learning. *Cluster Computing*,27(4):4941–4961, 2024.

[13] R. Priyadarshi and P. Kumar and N. Singh. Multi-Agent Deep Reinforcement Learning for Cooperative Routing in IoT Networks. *Ad Hoc Networks*,149:103244,2024

[14] Y. Zhou and L. Chen and F. Zhang. Adaptive Multi-Agent Reinforcement Learning Framework for Dynamic IoT Topologies. *Sensors*,25(3):1567,2025

[15] M. Kim, H. Park, and J. Lee. Xgate: Explainable reinforcement learning framework for trustworthy network management. *Sensors*, 24(6):3128, 2025.

[16] P. Li, X. Wang, and R. Gupta. Online pareto-approximation techniques for multi-objective reinforcement learning in iot routing. *Computer Networks*, 242:110112, 2025.

[17] Y. Chai and X.-J. Zeng. A multi-objective dyna-q based routing in wireless mesh network. *Applied Soft Computing*, 108:107486, 2021.

[18] D. Prabhu, R. Alageswaran, and S. Miruna Joe Amali. Multiple agent based reinforcement learning for energy efficient routing in wsn. *Wireless Networks*, 29(4):1787–1797, 2023.

[19] X. Su, Y. Ren, Z. Cai, Y. Liang, and L. Guo. A q-learning-based routing approach for energy efficient information transmission in wireless sensor network. *IEEE Transactions on Network and Service Management*, 20(2):1949–1961, 2022.

[20] D. Godfrey, B. Suh, B. H. Lim, K.-C. Lee, and K.-I. Kim. An energy-efficient routing protocol with reinforcement learning in software-defined wireless sensor networks. *Sensors*, 23(20):8435, 2023.

[21] S. Vaishnav, P. K. Donta, and S. Magnússon. Dynamic and distributed routing in iot networks based on multi-objective q-learning. *arXiv preprint arXiv:2505.00918*, 2025.

[22] W. K. Ghamry and S. Shukry. Multi-objective intelligent clustering routing schema for internet of things enabled wireless sensor networks using deep reinforcement learning. *Cluster Computing*, 27(4):4941–4961, 2024.

[23] A. M. Ikotun, A. E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, 622:178–210, 2023.

[24] E. U. Oti, M. O. Olusola, F. C. Eze, and S. U. Enogwe. Comprehensive review of k-means clustering algorithms. *criterion*, 12(08):22–23, 2021.

[25] M. A. Mohamud, H. Ibrahim, F. Sidi, S. N. M. Rum, Z. B. Dzolkhifli, Z. Xiaowei, and M. M. Lawal. A systematic literature review of skyline query processing over data stream. *IEEE Access*, 11:72813–72835, 2023.

[26] A. N. Fadhilah, T. A. Cahyanto, and I. Saifudin. Sort filter skyline in movie recommendation based on individual preferences: Performance and time complexity analysis. *Scientific Journal of Informatics*, 11(3):789–802, 2024.