

Article

An On-Demand Charging for Connected Target Coverage in WRSNs Using Fuzzy Logic and Q-Learning [†]

Phi Le Nguyen ^{1,*}, Van Quan La ^{1,‡}, Anh Duy Nguyen ^{1,‡}, Thanh Hung Nguyen ¹ and Kien Nguyen ²

¹ The School of Information and Communication Technology, Hanoi University of Science and Technology, Ha Noi 11615, Vietnam; quan.lv133135@sis.hust.edu.vn (V.Q.L.); duy.na184249@sis.hust.edu.vn (A.D.N.); hungnt@soict.hust.edu.vn (T.H.N.)

² The Graduate School of Engineering, Chiba University, 1-33, Yayoi-cho, Inage-ku, Chiba 263-8522, Japan; nguyenn@chiba-u.jp

* Correspondence: lenp@soict.hust.edu.vn

[†] This paper is an extended version of Q-learning-based, Optimized On-demand Charging Algorithm in WRSN. In Proceedings of the 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 24–27 November 2021.

[‡] Co-first authors.



Citation: Nguyen, P.L.; La, V.Q.; Nguyen, A.D.; Nguyen, T.H.; Nguyen, K. An On-Demand Charging for Connected Target Coverage in WRSNs Using Fuzzy Logic and Q-Learning. *Sensors* **2021**, *21*, 5520. <https://doi.org/10.3390/s21165520>

Academic Editors: Giovanni Pau and Nick Harris

Received: 16 June 2021

Accepted: 12 August 2021

Published: 17 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract: In wireless rechargeable sensor networks (WRSNs), a mobile charger (MC) moves around to compensate for sensor nodes' energy via a wireless medium. In such a context, designing a charging strategy that optimally prolongs the network lifetime is challenging. This work aims to solve the challenges by introducing a novel, on-demand charging algorithm for MC that attempts to maximize the network lifetime, where the term “network lifetime” is defined by the interval from when the network starts till the first target is not monitored by any sensor. The algorithm, named Fuzzy Q-charging, optimizes both the time and location in which the MC performs its charging tasks. Fuzzy Q-charging uses Fuzzy logic to determine the optimal charging-energy amounts for sensors. From that, we propose a method to find the optimal charging time at each charging location. Fuzzy Q-charging leverages Q-learning to determine the next charging location for maximizing the network lifetime. To this end, Q-charging prioritizes the sensor nodes following their roles and selects a suitable charging location where MC provides sufficient power for the prioritized sensors. We have extensively evaluated the effectiveness of Fuzzy Q-charging in comparison to the related works. The evaluation results show that Fuzzy Q-charging outperforms the others. First, Fuzzy Q-charging can guarantee an infinite lifetime in the WRSNs, which have a sufficient large sensor number or a commensurate target number. Second, in other cases, Fuzzy Q-charging can extend the time until the first target is not monitored by 6.8 times on average and 33.9 times in the best case, compared to existing algorithms.

Keywords: WRSN; Q-learning; on-demand charging algorithm; target coverage; connectivity

1. Introduction

Wireless Sensor Networks (WSNs) have found various applications, such as air quality monitoring, environmental management, etc., [1,2]. A WSN typically includes many battery-powered sensor nodes, monitoring several targets, and sending sensed data to a base station for further processing. In the WSNs, it is necessary to provide sufficient monitoring quality surrounding the targets (i.e., guaranteeing target coverage). Moreover, the WSNs need to have an adequate capacity for the communication between the sensors and base station (i.e., ensuring connectivity) [3–5]. The target coverage and connectivity are severely affected by the depletion of the battery on sensor nodes. When a node runs out of battery, it becomes a dead node without sensing and communication capability, damaging the whole network in consequence. Wireless Rechargeable Sensor Networks (WRSNs) leverages the advantages of wireless power transferring technology to solve that

critical issue in WSNs. A WRSN uses a mobile charger (MC) to wirelessly compensate for a rechargeable battery's energy consumption on a sensor node, aiming to guarantee both the target coverage and connectivity.

In a normal operation, the MC moves around the networks and performs charging strategies, which can be classified into periodic [6–10] or on-demand charging [11–17]. In the former, the MC, with a predefined trajectory, stops at charging locations to charge the nearby sensors' batteries. In the latter, the MC will move and charge upon receiving requests from the sensors, which have the remaining energy below a threshold. The periodic strategy is limited since it cannot adapt to the sensors' energy consumption rate dynamic. On the contrary, the on-demand charging approach potentially deals with the uncertainty of the energy consumption rate. Since a sensor with a draining battery triggers the on-demand operation, the MC's charging strategy faces a new time constraint challenge. The MC needs to handle two crucial issues: deciding the next charging location and the staying period at the location.

Although there are many, the existing on-demand charging schemes in the literature face two serious problems. The first one is the consideration of the same role for the sensor nodes in WRSNs. That is somewhat unrealistic since, intuitively, several sensors, depending on their locations, significantly impact the target coverage and the connectivity than others. Hence, the existing charging schemes may enrich unnecessary sensors' power while letting necessary ones run out of energy, leading to charging algorithms' inefficiency. It is of great importance to take into account the target coverage and connectivity simultaneously. The second problem is about the MC's charging amount, which is either a full capacity (of sensor battery) or a fixed amount of energy. The former case may cause: (1) a long waiting time of other sensors staying near the charging location; (2) quick exhaustion of the MC's energy. In contrast, charging a too small amount to a node may lead to its lack of power to operate until the next charging round. Therefore, the charging strategy should adjust the transferred energy level dynamically following the network condition.

Motivated by the above, we propose a novel on-demand charging scheme for WRSN that assures the target coverage and connectivity and adjusts the energy level charged to the sensors dynamically. Our proposal, named Fuzzy Q-charging, aims to maximize the network lifetime, which is the period from when the network starts till the first target is not monitored by any sensor. A target is considered to be monitored by a sensor if it is covered by a sensor (i.e., remaining within the sensor's sensing range) and the sensor is connected to the base station (i.e., a routing path exists between the sensor and the base station). To achieve this ultimate goal, Fuzzy Q-charging attempts to extend the lifetime of the sensors, in which, sensors that contribute more to target monitoring will be prioritized more. Fuzzy Q-charging combines two techniques: Fuzzy logic and Q-learning, each of which is designed to accomplish a given task with a certain goal. First, we exploit Fuzzy logic in an optimization algorithm that determines the optimal charging time at each charging location. The Fuzzy logic-based algorithm aims at maximizing the number of alive sensors. Fuzzy logic is used to cope with network dynamics by taking various network parameters into account during the determination process of the optimal charging time. Second, given the optimal charging time at every charging location, we leverage the Q-learning technique to select the next charging location to maximize the network lifetime. The MC maintains a Q-table containing the charging locations' Q-values representing the charging locations' goodness. The Q-values will be updated in a real-time manner whenever there is a new charging request from a sensor. We design the Q-value to prioritize charging locations at which the MC can charge a node depending on its critical role. After finishing tasks in one place, the MC chooses the next one, which has the highest Q-value, and determines an optimal charging time. The main contributions of the paper are as follows.

- We propose a Fuzzy logic-based algorithm that determines the energy level to be charged to the sensors. The energy level is adjusted dynamically following the network condition.

- Based on the above algorithm, we introduce a new method that optimizes the optimal charging time at each charging location. It considers several parameters (i.e., remaining energy, energy consumption rate, sensor-to-charging location's distance) to maximize the number of alive sensors.
- We propose *Fuzzy Q-charging*, which uses Q-learning in its charging scheme to guarantee the target coverage and connectivity. *Fuzzy Q-charging's* reward function is designed to maximize the charged amount to essential sensors and the number of monitored targets.

The remainder of the paper is constructed as follows. Section 3 describes the network model and Q-learning. We briefly review the related work in Section 2. Section 4 introduces our proposed algorithm. Section 5 includes the performance evaluation. Finally, Section 6 concludes the paper and shows our future work.

2. Related Work

Initially, we introduce the existing works related to periodic charging in WRSNs. In [6], the authors leverage PSO and GA to propose a charging path determination algorithm that minimizes the docking time during which the MC recharges itself at the depot. Ref. [7] jointly considers charging path planning and depot positioning to minimize the number of MCs while ensuring no sensor runs out of energy before being recharged. The work in [8] determines a charging path to maximize the MC's accumulative charging utility gain or minimize the MC's energy consumption during traveling. The authors then propose approximation algorithms with constant ratios for the maximization and minimization problems. Arguing that an MC can not fulfill all sensors' demand in dense networks, W. Xu et al. in [9] introduce a multi-chargers approximation model to increase the charging speed. In [10], C. Lin et al. derive a new energy transfer model with distance and angle factors. They also consider the problem of minimizing the total charging delay for all nodes. They use linear programming and obtain the optimal solution. As the charging schedule is always fixed, the periodic scheme fails to adapt to the dynamic of sensors' energy consumption.

Regarding the on-demand charging, the authors in [16] address the node failure problem. They first propose to choose the next charging node based on the charging probability. Second, they introduce a charging node selected method to minimize the number of other requesting nodes suffering from energy depletion. In [12,13], aiming to maximize the charging throughput, they propose a double warning threshold charging scheme. Two dynamic warning thresholds are triggered depending on the residual energy of sensors. The authors in [17] studied how to optimize the serving order of the charging requests waiting in the queue using the gravitational search algorithm. In [15], X. Cao et al. introduce a new metric (i.e., charging reward), which quantifies the charging scheme's quality. The authors then address the problem of maximizing the total reward in each charging tour under the constraint of the MC's energy and sensors' charging time windows. They use a deep reinforcement learning-based on-demand charging algorithm to solve the addressed problem.

The existing charging algorithms have two serious problems. First, the charging time problem has not been thoroughly considered. Most of the charging schemes leverage either the fully charging approach [6–8,11–13,16] or the partial charging one [18]. We want to emphasize that the charging time is an essential factor that decides how much the charging algorithm can prolong the network lifetime. Moreover, there is no existing work considering the target coverage and connectivity constraints concurrently. Most previous works treat all sensors in WRSNs evenly; hence, the MC may charge unnecessary sensors while necessary ones may run out of energy. Unlike them, this work addresses the target coverage and connectivity constraints in charging schedule optimization. We uniquely consider the optimization of charging time and charging location simultaneously. We use Fuzzy logic and Q-learning in our proposal.

Fuzzy logic has been applied in many fields, such as signal processing [19,20], robotics [21], and embedded controllers [22]. In WSNs, Fuzzy logic is a promising technique in dealing with various problems, including localization, routing [23,24], clustering [25], and data aggregation [26,27]. R. M. Al-Kiyumi et al. in [23] propose a Fuzzy logic-based routing for lifetime enhancement in WSNs, which maps the network status into corresponding cost values to calculate the shortest path. In [28], the authors also leverage Fuzzy logic and Q-learning but in a cooperative multi-agent system for controlling the energy of a microgrid. In [29], Fuzzy and Q-learning are combined to address the problem of thermal unit commitment. Specifically, each input state vector is mapped with the Fuzzy rules to determine all the possible actions with the corresponding Q-values. The main idea is exploiting Fuzzy logic to map the network status into corresponding cost values to calculate the shortest path. Recently, the authors in [14] use Fuzzy logic in an algorithm for adaptively determining the charging threshold and deciding the charging schedule. Different from the others, we use Fuzzy logic and Q-learning in our unique Fuzzy Q-charging proposal. The earlier version of this work has been published in [30], which considers only Q-charging.

3. Network Model, Q-Learning, and Fuzzy Logic

3.1. Network Model and Problem Definition

Figure 1 shows the considered network model, in which a WRSN monitors several targets. The network has three main components: an MC, sensor nodes, and a base station. The MC is a robot that can move and carry a wireless power charger. The sensor nodes can receive charged energy from the MC via a wireless medium. The base station is static and responsible for gathering sensing information. We assume that there are n sensors S_j ($j = 1, \dots, n$) and m targets T_k ($k = 1, \dots, m$). We call a sensor a *target-covering* sensor if it covers at least one target. Moreover, if there exists an alive routing path between a sensor and the base station, it is *connected* to the base station. The target is defined as to be *monitored* when at least one sensor connected to the base station covers it.

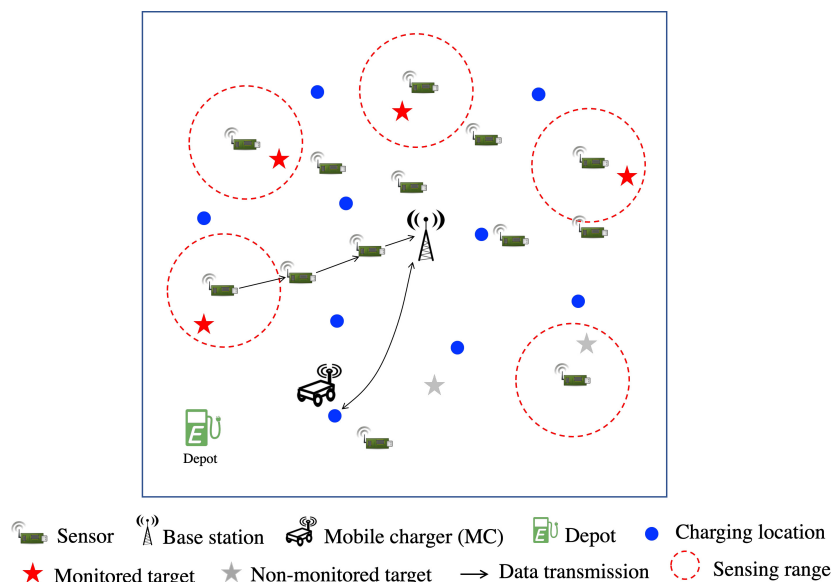


Figure 1. The network model.

A sensor node that has its remaining energy below E_{th} (i.e., a predefined threshold) will send a charging request to the base station. The base station then uses one-hop routing to transfer the request to the MC. We assume that the MC can interact with the base station over a long-range communication. We target a non-preemptive charging schedule, in which charging requests from sensors are queued at the MC. We assume that there are k charging locations denoted by D_1, \dots, D_k in the network. When the MC completes its tasks

at a charging location, it runs our proposed algorithm to select the next optimal charging location from D_1, \dots, D_k . Moreover, the MC also determines the optimal charging time at that charging location. When the energy of the MC goes below a threshold, it returns to the depot to recharge itself. Besides gathering the sensing information, the base station is also responsible for collecting information about the remaining energy sensors. Based on that, the MC estimates every sensor's energy consumption rate using the weighted averaging method. Given all sensors and the targets' locations, our on-demand charging algorithm aims to maximize the network lifetime.

3.2. Q-Learning

Q-learning is a reinforcement learning technique that is widely used in making a decision. The main idea is to achieve a specific goal based on experience learning from the past. The standard Q-learning framework consists of four components: an environment, one or more agents, a state space, and an action space, as shown in Figure 2. The Q-value represents the approximate goodness of the action concerning the agent's goal. An agent chooses actions according to the policy and the Q-value. After performing an action, the agent modifies its policy to attain its goal. The Q-value is updated using the Bellman equation as follows:

$$Q(S_t, A_t) \leftarrow (1 - \alpha)Q(S_t, A_t) + \alpha[R_t + \gamma \max_a Q(S_{t+1}, a)], \quad (1)$$

where $Q(S_t, A_t)$ is the Q-value of action A_t at a given state S_t . R_t is the reward obtained if performing action A_t in the state S_t . Moreover, $\max_a Q(S_{t+1}, a)$ is the maximum possible Q-value in the next state S_{t+1} for all possible actions a . α and γ are the learning rate and the future reward discount factor. Their values are set between 0 and 1.

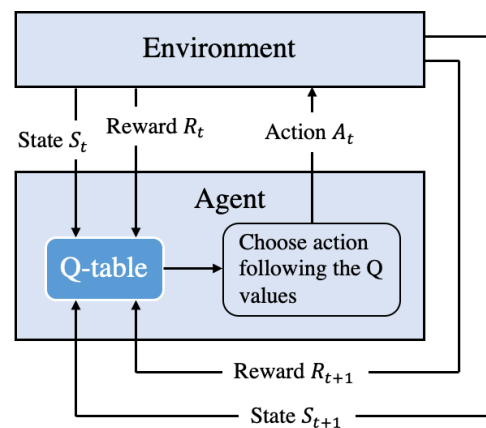


Figure 2. Q-learning overview.

3.3. Fuzzy Logic

A fuzzy logic system consists of three components: *fuzzification*, *fuzzy logic controller*, and *defuzzification*. The first component converts the crisp values of the variable into their fuzzy form using some membership functions. The second one is responsible for simulating the human reasoning process by making *fuzzy inference* based on inputs and a set of defined **IF-THEN** rules. The module itself can be separated into two subcomponents, namely *Knowledge Base* and *Inference Engine*. *Knowledge Base* is a set of specifically designed rules so that together with the input states of variables, they will produce consistent results. Each rule's form is "**IF** {set of input} **THEN** {output}". More explicitly, a fuzzy rule R_i with k -inputs and 1-output has the following form.

$$R_i : \mathbf{IF} (I_1 \text{ is } A_{i1}) \Theta (I_2 \text{ is } A_{i2}) \Theta \dots \Theta (I_k \text{ is } A_{ik}) \\ \mathbf{THEN} (O \text{ is } B_i), \quad (2)$$

where $\{I_1, \dots, I_k\}$ represents the crisp inputs to the rule. $\{A_{i1}, \dots, A_{ik}\}$ and B_i are linguistic variables. The operator Θ can be **AND**, **OR**, or **NOT**. The *Inference Engine* is in charge of the estimation of the Fuzzy output set. It calculates the membership degree (μ) of the output for all linguistic variables by applying the rule set described in the *Knowledge Base*. For Fuzzy rules with lots of inputs, the output calculation depends on the operators used inside it, i.e., **AND**, **OR**, or **NOT**. The calculation for each type of operator is described as follows:

$$\begin{aligned} (I_i \text{ is } A_i \text{ AND } I_j \text{ is } A_j) : \\ \mu_{A_i \cap A_j}(I_{ij}) &= \min(\mu_{A_i}(I_i), \mu_{A_j}(I_j)), \\ (I_i \text{ is } A_i \text{ OR } I_j \text{ is } A_j) : \\ \mu_{A_i \cup A_j}(I_{ij}) &= \max(\mu_{A_i}(I_i), \mu_{A_j}(I_j)), \\ (\text{NOT } I_i \text{ is } A_i) : \\ \mu_{\bar{A}_i}(I_i) &= 1 - \mu_{A_i}(I_i). \end{aligned}$$

The last component helps to convert the fuzzy output set from the linguistic variables into a crisp value. The most popular fuzzy solution is a methodology called the centroid technique, described as follows:

$$\text{Center of Gravity of } B \text{ (CoG}_B) = \frac{\int_{-\infty}^{+\infty} \mu_B(z)zdz}{\int_{-\infty}^{+\infty} \mu_B(z)dz}, \quad (3)$$

where $\mu_B(z)$ is the output membership function of the linguistic variable B .

4. Fuzzy Q-Charging Algorithm

4.1. Overview

We follow the on-demand charging strategy, in which a sensor sends a charging request to the base station when its energy is below a predefined threshold E_{th} . The base station then uses one-hop routing to transfer the request to the MC. The request is inserted into the waiting list at the MC. The MC then performs the following procedures to update the Q-table:

- The MC leverages Fuzzy logic to calculate a so-called *safe energy level* (denoted as E_{sf}), which is sufficiently higher than E_{th} . The MC then uses the algorithm described in Section 4.3 to determine the charging time at each charging location. The charging time is optimized to maximize the number of sensors that guarantee the safe energy level.
- The MC calculates the reward of every charging location using Equation (12) and updates the Q-table using Equation (4).

After finishing charging at a charging location, the MC selects the next charging location as the one with the highest Q-value. Finally, the MC moves to the next charging location and charges for the determined charging time. When the energy of the MC goes below a threshold, it returns to the depot to recharge itself. Figure 3 presents the overview of our charging algorithm. To facilitate the reading, we summarize all the used notations in Table 1.

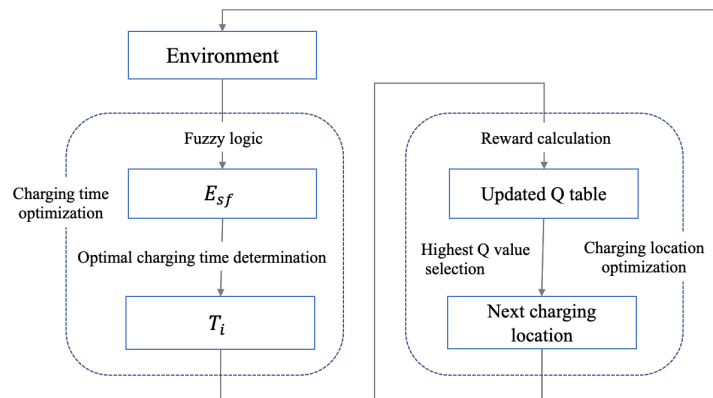


Figure 3. The flow of Fuzzy Q-learning-based charging algorithm.

Table 1. List of notations.

Notation	Definition
D_i	The i -th charging location
$Q(D_i, D_j)$	Action value of the action moving from D_i to D_j
$r(D_i)$	Reward obtained after the MC moves to D_i
α	The learning rate of the Q-learning algorithm
γ	The discount factor of the Q-learning algorithm
E_{th}	The threshold for sending a charging request
E_{sf}	The safe charging level
θ	The safe energy factor
E_{max}	The maximum energy capacity of the sensors
p_j^i	The per second energy that a sensor S_j is charged when the MC stays at D_i
e_j	Energy consumption rate of sensor S_j
E_j	Remaining energy of sensor S_j
T_i	The optimal charging time at D_i
L_r, E_{min}	Fuzzy input variables
w_j	The priority index of S_j
ξ_j	The energy severity index of S_j , $\xi_j = \frac{e_j}{E_j}$.

4.2. State Space, Action Space and Q Table

In our Q-learning-based model, the network is considered the environment while the MC is the agent. A state is defined by the current charging location of the MC, and an action is a move to the next charging location. Each MC maintains its own Q-table, which is a two-dimensional array. Each row represents a state, and each column represents an action. An item $Q(D_j, D_i)$ in the j -th row and i -th column represents the Q-value corresponding to the action when the MC moves from the current charging square D_j to the next charging location D_i . Figure 4 shows an illustration of our Q-table. In the figure, the gray row represents the Q-values concerning all possible actions when the MC stays at the charging location D_c . The green cell depicts the maximum Q-value regarding the next charging location.

Let D_c be the current charging location and D_i be an arbitrary charging location, then the Q-value of action moving from D_c to D_i is iteratively updated by using the Bellman equation as follows:

$$Q(D_c, D_i) \leftarrow Q(D_c, D_i) + \alpha(r(D_i) + \gamma \max_{1 \leq j \leq l} Q(D_i, D_j) - Q(D_c, D_i)). \quad (4)$$

The equation's right side consists of two elements, including the current Q-value and the temporal difference. The temporal difference measures the gap between the estimated target, i.e., $r(D_i) + \gamma \max_{1 \leq j \leq l} Q(D_i, D_j)$, and the old Q-value, i.e., $Q(D_c, D_i)$. α and γ are two hyper-parameters whose names are learning rate and discount factor, respectively. $r(D_i)$ is our proposed reward function, which will be detailed in Section 4.5.

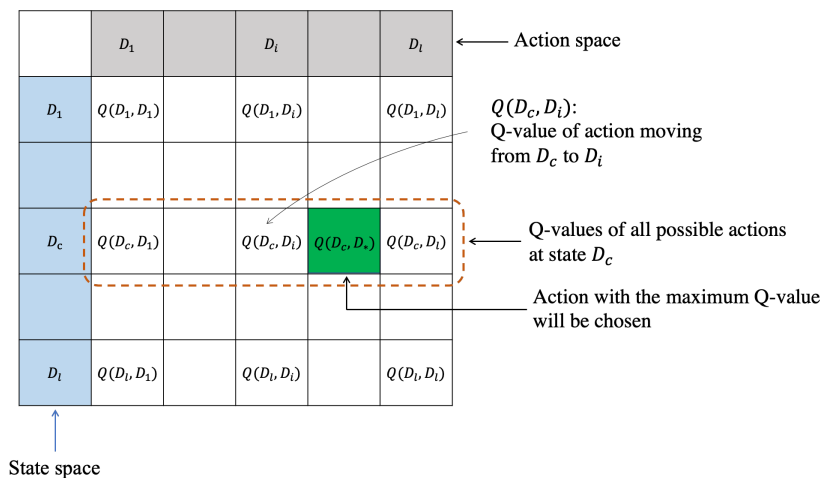


Figure 4. An illustration of the Q-table.

In the following, we first describe our algorithms to determine the optimal charging time and the safety energy level in Sections 4.3 and 4.4. Then, we present the details of the reward function and the mechanism for updating the Q-table in Sections 4.5 and 4.6.

4.3. Charging Time Determination

We aim to design a charging strategy so that the number of sensors reaching a safe energy level is as big as possible after each charging round. Here, the safe energy level means the energy amount that is sufficiently greater than E_{th} . We define the safe energy level, E_{sf} , as

$$E_{sf} = E_{th} + \theta E_{max}, \quad (5)$$

where E_{max} is the maximum energy capacity of the sensors. θ is an adaptive parameter, named safe energy factor, which is determined by Fuzzy logic. The algorithm determining θ algorithm will be described in Section 4.4.

A sensor node has the *critical status* if its remaining energy is smaller than to E_{sf} . The sensor with a critical status is named as *critical sensor*. Otherwise, a sensor node is called a *normal sensor*. For each charging location D_i ($1 \leq i \leq l$), we want to determine the optimal charging time \mathcal{T}_i to minimize the number of critical sensors.

We adopt the multi-nodes charging model, in which the MC can simultaneously charge to all sensors. We leverage the charging model proposed in [31], which has been widely used in most research in the WRSN-related studies. This model follows the RF-power harvesting. The authors in [31] have performed experiments to verify the model in the real environment. Following the experiment results shown in the paper, the charging

energy is about 12×10^{-4} W when the distance is 1.6 m. According to [31], the per second energy that a sensor S_j is charged when the MC stays at D_i is given by

$$p_j^i = \frac{\lambda}{(d(S_j, D_i) + \beta)^2}, \quad (6)$$

where λ and β are known constants decided by the hardware of the charger and receiver. $d(S_j, D_i)$ is the Euclidean distance between S_j and D_i . We denote e_j as the energy consumption rate of S_j , which is estimated by the MC. Suppose that the MC charges S_j at D_i , we denote the remaining energy of S_j when the charging process starts and finishes as E_j and E'_j , then $E'_j = E_j + (p_j^i - e_j) \times \mathcal{T}_i$. At the charging location D_i , we call $p_j^i - e_j$ the *energy gain* of S_j . The remaining energy of S_j will increase if its energy gain is positive and decreases otherwise. Note that the energy of S_j equals the safety energy level, if the charging time equals to $\frac{E_{sf} - E_{a_j}}{p_{a_j}^i - e_{a_j}}$, which is named as the *safety charging time* of S_j with respect

to the charging location D_i and denoted as Δ_j^i . The sensors can be classified into four groups. The first and second ones contain normal sensors with positive energy gain and critical sensors with negative energy gain, respectively. The third and fourth groups contain normal sensors with negative energy gain and critical sensors with positive energy gain, respectively. Obviously, the first and second groups' sensors do not change their status no matter how long the MC charges at D_i . In contrast, a sensor S_j in the third group will fall into the critical status, and a sensor in the fourth group can alleviate the critical status, if the charging time \mathcal{T}_i is greater than or equals to its safety charging time, i.e., $\mathcal{T}_i > \Delta_j^i$. Hereafter, we call the sensors in the third group *negative normal sensors* and the sensors in the fourth group *positive critical sensors*. Let ϵ_1, ϵ_2 denote the number of sensors belonging to the third and fourth groups whose status changes after being charged (from critical to normal and vice versa). It is worth noting that the greater the value of \mathcal{T}_i , the greater ϵ_1 , and the greater ϵ_2 , also. Our objective is to determine the optimal value of \mathcal{T}_i to maximize $\epsilon_1 - \epsilon_2$. This goal can be achieved by using the following algorithm.

- First, we calculate the safety energy charging time of all negative normal sensors (denoted as $\Delta_{a_1}^i, \dots, \Delta_{a_u}^i$) and positive critical sensors (denoted as $\Delta_{b_1}^i, \dots, \Delta_{b_v}^i$).
- We then combine the values of $\Delta_{a_1}^i, \dots, \Delta_{a_u}^i$ and $\Delta_{b_1}^i, \dots, \Delta_{b_v}^i$ into an array denoted as $\Delta_{c_1}^i, \dots, \Delta_{c_{u+v}}^i$, where $\Delta_{c_1}^i, \dots, \Delta_{c_{u+v}}^i$ have been sorted by decreasing order (i.e., $\Delta_{c_1}^i \geq \Delta_{c_2}^i \geq \dots \geq \Delta_{c_{u+v}}^i$). We have an important observation that the value of $(\epsilon_1 - \epsilon_2)$ does not change when \mathcal{T}_i varies in the range from $\Delta_{c_p}^i$ to $\Delta_{c_{p+1}}^i$ ($1 \leq p \leq u + v$). Therefore, the optimal value of \mathcal{T}_i can be easily determined by brute force search over $\Delta_{c_1}^i, \dots, \Delta_{c_{u+v}}^i$.

4.4. Fuzzy Logic-Based Safe Energy Level Determination

4.4.1. Motivation

We observe that E_{sf} contributes significantly to the algorithm's performance. When E_{sf} 's value is small, the MC's maximum energy amount, charging to the sensors, is also small. Accordingly, after being charged, the sensor's battery may quickly go below the threshold. On the contrary, if E_{sf} is too large, the MC needs to spend a long time at every charging point. Consequently, the sensors far from the charging points may run out of energy while waiting for their turn. To this end, we leverage Fuzzy logic to adjust E_{sf} 's value adaptively. In the following, we first analyze factors that affect E_{sf} . We then describe the proposed algorithm. Remember that according to Equation (5), E_{sf} is determined by $E_{sf} = E_{th} + \theta E_{max}$. Therefore, below, we will show how to adjust the value of θ adaptively.

When the MC stays at a charging point, the nearby sensors receive a more significant energy amount than the faraway sensors. Therefore, each near sensor's charging amount is likely more significant than its energy consumption for sensing and communication. Consequently, the nearby ones tend to increase their battery level gradually. On the

contrary, the faraway sensors tend to decrease due to the energy consumption for sensing and communication tasks. It is expected that the MC should spend a longer time at charging points where sensors far from it do not encounter critical situations (i.e., having low residual energy or high energy consumption rate). Based on this observation, our algorithm is designed so that E_{sf} tends to receive a small value in the following cases.

- The residual energy of all sensors is small.
- Many sensors need to be charged.

We propose a Fuzzy logic-based E_{sf} determination algorithm, which utilizes the following two variables. The first one is the minimum residual energy of all sensors E_{min} . The second is the number of charging requests, denoted as L_r .

The details of the algorithm is presented in Algorithm 1.

4.4.2. Fuzzification

With the two variables L_r and E_{min} , we denote the output as the value of θ . Each input is mapped into three linguistic variables that are *low*, *medium*, and *high*. Meanwhile, the output is mapped into five ones, namely *very low*, *low*, *medium*, and *high*. We leverage the triangular and trapezoidal fuzzy numbers, whose formulas are given below:

$$\text{Triangular}(x, a, b, c) = \begin{cases} 0 & , \quad x \leq a \\ \frac{x-a}{b-a} & , \quad a \leq x \leq b \\ \frac{c-x}{c-b} & , \quad b \leq x \leq c \\ 0 & , \quad c \leq x \end{cases} \quad (7)$$

$$\text{Trapezoidal}(x, a, b, c, d) = \begin{cases} 0 & , \quad x \leq a \\ \frac{x-a}{b-a} & , \quad a \leq x \leq b \\ 1 & , \quad b \leq x \leq c, \\ \frac{d-x}{d-c} & , \quad c \leq x \leq d \\ 0 & , \quad d \leq x \end{cases} \quad (8)$$

where x is the crisp input, and a, b, c, d are membership function ranges of the fuzzy variables. The values of a, b, c, d are represented in Tables 2 and 3. Figures 5 and 6 depict the Fuzzy membership functions of the input and the output variables, respectively.

Table 2. Input variables with their linguistic values and corresponding membership function.

Input Variable	Linguistic Value	Membership Function
L_r	<i>L</i>	$[0, 0, 2, 6]$
	<i>M</i>	$[2, 6, 10]$
	<i>H</i>	$[6, 10, \infty, \infty]$
E_{min}	<i>L</i>	$[0, 0, \frac{1}{4}E_{th}, \frac{2}{4}E_{th}]$
	<i>M</i>	$[\frac{1}{4}E_{th}, \frac{2}{4}E_{th}, \frac{3}{4}E_{th}]$
	<i>H</i>	$[\frac{2}{4}E_{th}, \frac{3}{4}E_{th}, E_{th}, E_{th}]$

Table 3. Output variable with its linguistic values and membership function, $\delta = (0.1 - 2 \frac{E_{th}}{E_{max}})$.

Output Variable	Linguistic Value	Membership Function
θ	<i>VL</i>	$[0, 0, \frac{1}{3}\delta]$
	<i>L</i>	$[0, \frac{1}{3}\delta, \frac{2}{3}\delta]$
	<i>M</i>	$[\frac{1}{3}\delta, \frac{2}{3}\delta, \delta]$
	<i>H</i>	$[\frac{2}{3}\delta, \delta, \delta]$

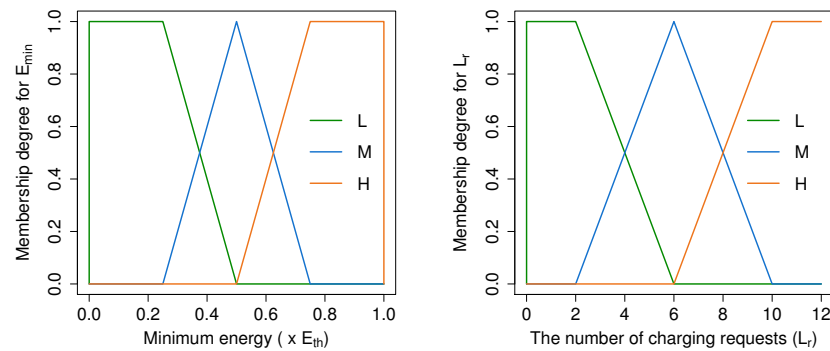


Figure 5. Fuzzy input membership functions.

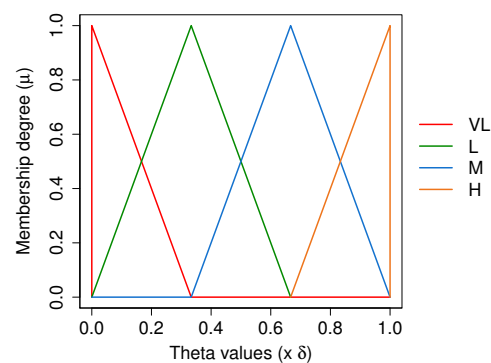


Figure 6. Fuzzy output membership functions.

4.4.3. Fuzzy Controller

There are two input variables; each is converted to three fuzzy sets, so we have a total of $3^2 = 9$ rules in the *Knowledge Base*, which are listed in Table 4. The rules are designed to reflect the observation described in Section 4.4.1. Our rules have the form of “IF (L_r is A) AND (E_{min} is B) THEN (θ is D)”, in which A, B obtain the values of *low*, *medium*, or *high*, and D is either *very low*, *low*, *medium*, or *high*. For the ease of presentation, we use the following notations: $VL = \text{very low}$, $L = \text{low}$, $M = \text{medium}$, $H = \text{high}$. As the Fuzzy rules are based on the AND operator, the output membership degree is defined by

$$\mu_{R_i} = \min\{\mu_A(L_r), \mu_B(E_{min})\}, \forall i = 1, \dots, 9. \quad (9)$$

Table 4. Fuzzy rules for safe energy level determination.

$R_{\#}$	Input		Output
	L_r	E_{min}	θ
1	L	L	H
2	L	M	M
3	L	H	L
4	M	L	M
5	M	M	L
6	M	H	VL
7	H	L	L
8	H	M	VL
9	H	H	VL

4.4.4. Defuzzification

After the *Inference Engine* determines the output's membership degree on fuzzy sets (by using Equation (9)), the fuzzy set with the highest membership degree is considered the output variable. Finally, we utilize the CoG function to calculate the crisp value of the output's fuzzy set.

Let us consider an example where $L_r = 3$, $E_{min} = 3$, and $E_{th} = 4$. First, the fuzzy inputs of the two variables L_r and E_{min} are calculated using Equations (7) and (8), as presented in Table 5. Then, we use Equation (9) to calculate the output membership degree of each fuzzy rule. For example, the fuzzy output for rule R_1 is given by

$$\begin{aligned}\mu_{R_1} &= \min(\mu_L(L_r), \mu_L(E_{min})) \\ &= \min(0, 0) = 0.\end{aligned}$$

The values of all the fuzzy rules are shown in Table 6. From Table 6, it can be seen that rule 3 achieves the highest membership degree. Therefore, the output variable θ obtains the linguistic value of $low(L)$. Finally, we use the CoG function to convert the output's linguistic variable into a crisp value. According to Equation (3), the CoG of the output linguistic variable $low(L)$ is defined by

$$CoG_L = \frac{\int_0^\delta \theta \mu_L(\theta) d\theta}{\int_0^\delta \mu_L(\theta) d\theta}, \quad (10)$$

where μ_L is the membership function of $low(L)$. Following the definition of the output membership function represented in Table 3, we have

$$\mu_L(\theta) = \begin{cases} \frac{3}{\delta}\theta & , \quad 0 \leq \theta \leq \frac{1}{3}\delta \\ 2 - \frac{3}{\delta}\theta & , \quad \frac{1}{3}\delta \leq \theta \leq \frac{2}{3}\delta \\ 0 & , \quad \frac{2}{3}\delta \leq \theta \leq \delta. \end{cases} \quad (11)$$

where $\delta = \left(0.1 - 2\frac{E_{th}}{E_{max}}\right)$. By substituting Equation (10) into Equation (11), we obtain the value of CoG_M as $\frac{1}{3}\delta$. It means that with $L_r = 3$, $E_{min} = 3$, and $E_{th} = 4$, the value of θ is $\frac{1}{3}\delta$, thus the *safe energy level* E_{sf} is given by $E_{sf} = E_{th} + \frac{1}{3}\delta E_{max}$.

Table 5. Inputs of linguistic variables.

Input Variable	Membership Function	Value
L_r	μ_L	0.75
	μ_M	0.25
	μ_H	0.00
E_{min}	μ_L	0.00
	μ_M	0.00
	μ_H	1.00

Table 6. Fuzzy rules evaluation.

$R_\#$	$\mu_{R_\#}$	$R_\#$	$\mu_{R_\#}$	$R_\#$	$\mu_{R_\#}$
1	0.00	4	0.00	7	0.00
2	0.00	5	0.00	8	0.00
3	0.75	6	0.25	9	0.00

Algorithm 1: Fuzzy Logic-based θ determination

```

Input   :  $E_j, E_{th}, L_r, \forall j = \overline{1, n}$ ;
            $E_j$ : Remaining energy of sensor  $S_j$ ;
            $E_{th}$ : Energy threshold sensors request charging;
            $L_r$ : The number of charging request;

Output  :  $\theta$ 
 $\theta \leftarrow 0$ ;
for  $i = 1$  to  $m$  do
   $E_{min} = \min_{1 \leq j \leq n} E_j$ ;
  // Fuzzification
   $\mu_L(E_{min}) = \text{Trapezoidal}(E_{min}, 0, 0, \frac{1}{4}E_{th}, \frac{2}{4}E_{th})$ ;
   $\mu_M(E_{min}) = \text{Triangular}(E_{min}, \frac{1}{4}E_{th}, \frac{2}{4}E_{th}, \frac{3}{4}E_{th})$ ;
   $\mu_H(E_{min}) = \text{Trapezoidal}(E_{min}, \frac{2}{4}E_{th}, \frac{3}{4}E_{th}, E_{th}, E_{th})$ ;
   $\mu_L(L_r) = \text{Trapezoidal}(L_r, 0, 0, 2, 6)$ ;
   $\mu_M(L_r) = \text{Triangular}(L_r, 2, 6, 10)$ ;
   $\mu_H(L_r) = \text{Trapezoidal}(L_r, 10, 10, \infty, \infty)$ ;
  // Fuzzy controller
   $\mathcal{M} \leftarrow \text{null}$ ;
  for  $A \in \{L, M, H\}$  do
    for  $B \in \{L, M, H\}$  do
       $\mu = \min(\mu_A(L_r), \mu_B(L_r))$ ;
       $\mathcal{M}.\text{add}(\mu)$ ;
    end
  end
  // Defuzzification
   $l = \arg \max_{\mu} \mathcal{M}$ ;
   $D \leftarrow$  the output of rule  $R_i$ ;
   $\theta \leftarrow$  the value of  $\theta$  by CoG function;
end
return  $\theta$ 

```

4.5. Reward Function

Our objective is to maximize the network lifetime. To achieve this goal, we need to guarantee that every target is monitored by at least one sensor. Note that a target is considered to be monitored by a sensor if it is covered by a sensor (i.e., remaining within the sensor's sensing range) and the sensor is connected to the base station (i.e., a routing path exists between the sensor and the base station). Hence, the next charging location of MC should be selected to prioritize the following sensors:

- The sensors with either high energy consumption rate or low level of remaining energy;
- The sensors either cover many targets or participate in many routing paths from target-covering sensors to the base station.

We emphasize that our design can be applied to any routing protocol. However, in the experiments in Section 5, we adopt the geographic greedy routing protocol [32]. The geographic greedy routing protocol is widely accepted in WSNs due to its simplicity and efficiency. In this routing protocol, each node chooses the next hop to be the neighboring node closest to the destination. It is worth noting that in geographic routing, it is usually assumed that every node knows its location, and the location of its one-hop neighbors. This assumption can be realized by using positioning services [33] and the neighbor notification packets, respectively. Besides, the source node knows the position of the destination node. This assumption is legitimate in geographic routing [34–42]. To determine the next node, the current node only needs to look up in its neighbor table and find the one with the smallest distance to the destination. The computational complexity for determining such the next node is only $O(m)$, where m is the number of the current node's one-hop neighbors. For each sensor S_j , we define an *energy severity index* (denoted as ζ_j) which is calculated by the ratio of S_j 's energy consumption rate e_j to its remaining energy E_j ($\zeta_j = \frac{e_j}{E_j}$). We can see that the higher the energy severity index, the more critical the sensor is. Hence, the MC should charge a larger amount of energy to the sensor. We name a *priority index* for

each sensor, which indicates the sensor's importance in covering targets and transferring sensory data to the base station. Specifically, a sensor S_j has the priority index w_j , defined as a sum of its covered target number and the routing-path number. The routing path is from a target-covering sensor to the base station through S_j . Similarly, the priority index is proportional to the significant effects on the target coverage and connectivity. The sensor with a higher priority index needs more energy from the MC than a lower-index one does.

We reflect the observation mentioned above in the design of the reward function, which considers three factors *energy factor*, *sensor priority factor*, and *target monitoring factor*. The first two factors depict the relationship between the sensors' energy severity indexes and priority indexes with the energy they will be charged. The last factor estimates the number of monitored targets. We denote a charging location D_i 's energy factor, sensor priority factor, and target monitoring factor as $\mathbf{E}(D_i)$, $\mathbf{P}(D_i)$, and $\mathbf{T}(D_i)$, respectively. The first two factors are designed based on a well-known fact that given two sorted number sets $a_1 \geq a_2 \geq \dots \geq a_n$ and $b_1 \geq b_2 \geq \dots \geq b_n$. Let π_1 and π_2 be two permutations of $\{1, 2, \dots, n\}$, then $\sum_{i=1}^n a_{\pi_1(i)} b_{\pi_2(i)}$ attains the greatest value when $\pi_1 = \pi_2 = \{1, \dots, n\}$.

Intuitively, $\sum_{i=1}^n a_{\pi_1(i)} b_{\pi_2(i)}$ is maximized when $a_{\pi_1(i)}$ is proportional with $b_{\pi_2(i)}$.

First, the energy factor $\mathbf{E}(D_i)$ is defined by $\sum_{j=1}^n p_j^i \times \xi_j$, which sums up the products of energy charged to the sensors and their severity indexes. According to the fact mentioned above, $\mathbf{E}(D_i)$ tends to achieve the maximal value when p_j^i is proportional with ξ_j . It means that $\mathbf{E}(D_i)$ will increase if we increase the value of p_j^i for sensors with a large value of e_j and a small value of E_j . In other words, our algorithm encourages charging more to sensors with a large energy consumption rate and small remaining energy.

The priority factor $\mathbf{P}(D_i)$ is calculated by $\sum_{j=1}^n w_j p_j^i$, where w_j is the priority of sensor S_j . Obviously, the second factor is maximized if w_j is proportional with p_j^i . Therefore, our algorithm will prioritize the sensors that play important roles in guaranteeing the network's target coverage and connectivity.

Third, the target monitoring factor $\mathbf{T}(D_i) = \frac{t_i}{m}$ depicts the ratio of the number of targets that are monitored (i.e., t_i) to the total number of the targets (i.e., m). By including this factor, our algorithm will choose the next charging location that maximizes the number of monitored targets.

The reward of a charging location D_i (denoted by $r(D_i)$) is the normalized sum of the three factors. $r(D_i)$ can be calculated as follows.

$$r(D_i) = \frac{\sum_{j=1}^n \frac{1}{E_j} p_j^i e_j}{\sum_{k=1}^l \sum_{j=1}^n \frac{1}{E_j} p_j^k e_j} + \frac{\sum_{j=1}^n w_j p_j^i}{\sum_{k=1}^l \sum_{j=1}^n w_j p_j^k} + \frac{t_i}{\sum_{k=1}^l t_k}. \quad (12)$$

4.6. Q Table Update

To determine the optimal charging time and calculate the charging locations' Q-value, the MC needs information about the sensors' remaining energy and energy consumption rate.

Since the base station gathers and transfers all sensors' remaining energy information to the MC periodically, the MC can estimate every sensor's energy consumption rate based on the received information. In this work, we leverage a simple weighted averaging method to estimate the energy consumption rate. The energy consumption rate of sensor S_j is defined by the average of its energy consumption rate at L timings in the past weighted by the corresponding time,

$$e_j = \frac{\sum_{k=1}^L e_j^k \times t_k}{\sum_{k=1}^L t_k},$$

where e_j^k is the energy consumption rate of e_j at timing t_k in the past.

The more frequently the sensors update the information to the base station, the more accurately the MC can estimate the sensors' energy consumption rate. However, sending such information too often may consume significant energy of sensors. Therefore, in our algorithm, the sensors only update their remaining energy in the following scenarios.

5. Performance Evaluation

We compare the performance of Fuzzy Q-charging with the most relevant four existing algorithms. The first one is INMA [16], in which the MC determines the next sensor to charge based on factors, including the residual energy of sensors and the distance from sensors to the MC. The next charging sensor is chosen to minimize the number of other requesting nodes that may suffer from energy depletion. The second one is GSA [17]. At each charging round in GSA, the MC uses the gravitational search algorithm to determine a near-optimal charging order to fulfill all charging requests. In both INMA and GSA, the MC always charges to the maximum battery capacity of the sensor. The third comparison benchmark is RMP-RL [15]. RMP-RL uses the Deep Q-learning technique to determine the charging path of the MC. The objective is to minimize the number of dead sensor nodes and the moving distance of MC. The last comparison benchmark is our previous work, namely Q-charging [30]. Q-charging leverages Q-learning to determine the next charging location. However, different from Fuzzy Q-charging, Q-learning tries to maximize the number of sensors being charged to a predefined energy level. Besides, we also measure the network lifetime when there is no charging scheme is applied. Hereafter, we call this option a "no-charging" scheme.

We conduct two experiments, among which the first complements the other. The first experiment investigates the impact of parameters γ and α on the performance of our proposal. Based on the first experiment results, we determine the optimal values of γ and α . They are used in the second experiment, which compares Fuzzy Q-charging performance to the existing works. The metrics of interest include the network lifetime and the number of non-monitored targets over time in the evaluation. In all experiments, the network area is fixed at the size of 100 m \times 100 m. The sensors and targets were randomly scattered in the simulated region. The charging locations are positioned in the same place as the sensors. Each value plotted on the curves is the average obtained from 20 runs.

Regarding the charging model, we adopted the parameters proposed in [31,43]. The parameters have been verified by the experiments in [31,43]. More specifically, we set $\lambda = 4.32 \times 10^{-4}$, $\beta = 0.2316$, $e_{MC} = 10$ J/s, $e_{move} = 0.01$ J/s. Moreover, the initial energy of sensors and MC are 10J and 100J, respectively. Each sensor has a battery capacity of 10J. In this simulation, we assume that sensors follow the Zigbee communication standard. We set the transmission range of sensors to 15 m. The reason is that based on the real experiment results reported in [44], the transmission becomes unreliable (i.e., the drop ratio becomes greater than 0) beyond 15 m. The velocity of the MC is 5 m/s. The average energy consumption rate of the sensors is estimated by the base station, as mentioned in Section 3. The parameters are summarized in Table 7.

5.1. Impacts of Parameters

This section studies the impacts of parameters α and γ on our proposed algorithm's performance. Although we have conducted experiments with various settings, the results show similar trends. Therefore, we only present the results in a scenario with 300 sensors and 200 targets.

5.1.1. Impacts of α

We vary the value of α from 0.3 to 0.8 and measure the network lifetime's variation. The results are shown in Table 8. We can see the network lifetime enlarges significantly when α increases from 0.3 to 0.5. It dramatically drops when α reaches 0.6 and becomes stable. This phenomenon can be explained as follows. As shown in Equation (4), the new

Q-value is calculated from the current Q-value, the reward, and the estimated maximal Q-value. α is the weight of the last two components, while $1 - \alpha$ is the weight of the first one. Intuitively, the current Q-value reflects the experience the agent has learned so far. Meanwhile, the reward and the estimated maximal Q-value can be seen as the knowledge the agent has just attained through the current action and the future prediction, respectively. When α is relatively small, e.g., less than 0.5, increasing α helps exploit the experience and the future forecast in making the decision, thus improving the goodness of the actions. However, when α is significantly large, the current experience and the future prediction dominate the Q-value. It means that the agent makes a decision primarily based on the current reward and future forecast and ignore all the experiences the agent has learned so far. The Q-learning now converges to the greedy approach. That is why the performance drops severely when α increases from 0.5 to 0.6 and becomes stable beyond that. From the experiment results, α should be moderate values around 0.4 and 0.5.

Table 7. System parameters.

Factor	Value
λ	4.32×10^{-4}
β	0.2316
Initial energy of the MC	100 J
Battery capacity of MC	500 J
The velocity of the MC	5 m/s
Initial energy of sensors	10 J
Battery capacity of sensors	10 J
E_{th}	4 J
Sensing range	7.5 m
Transmission range	15 m
Number of sensors	200~400
Number of targets	100~300
Per second packet generation probability	0.05~0.25

Table 8. The impact of α on the network lifetime.

α	0.3	0.4	0.5	0.6	0.7	0.8
Network lifetime (10^3 s)	235.042	246.601	246.268	243.493	243.719	244.183

5.1.2. Impacts of γ

Similarly, the impacts of γ is shown in Table 9. In this experiment, we set the value of α to 0.5. As can be observed, the network lifetime gradually decreases when γ increases. This is because γ is the weight of the predicted maximal Q-value in the future. The greater the γ , the more importantly the future prediction information contributes to the agent's action. When γ is significantly small, the role of the future prediction (i.e., Q_{max}) in the Q-value is minor. Increasing γ helps agents exploit more future information in making action decisions, thus improving the decision's goodness, thereby extending the network lifetime. However, when γ is significantly large, e.g., more than 0.6, increasing γ will eliminate the impacts of the current Q-value in making a decision. In other words, the agent tends to ignore all experiences learned so far and relies primarily on the future prediction. As the future prediction does not entirely correct, the performance of the Fuzzy Q-learning downgrades severely. From the experiment results, the optimal value of γ is from 0.4 to 0.5.

Table 9. The impact of γ on the network lifetime.

γ	0.4	0.5	0.6	0.7	0.8
Network lifetime (10^3 s)	∞	∞	302.876	302.876	126.89

5.2. Comparison with Existing Algorithms

This section presents the comparison of our proposal to the existing ones. Following the previous observation, we set the values of α and γ to 0.5 and 0.4, respectively.

5.2.1. Impacts of the Number of Sensors

Figure 7 depicts the network lifetime when the number of sensors varies from 200 to 400. In this experiment, the packets are generated randomly, with the probability of 0.2 packets in 1s; the target number is 150. The targets are randomly located in the network area. We can see that the network lifetime increases along with the increasing number of sensors in all algorithms due to each sensor's traffic load has been reduced. However, Fuzzy Q-charging consistently outperforms the others. Ours can extend the network lifetime by at least 19.3 times. Moreover, the performance gaps between Fuzzy Q-charging and the others are proportional to the sensor number. When the number increases from 200 to 250, the gaps are small. However, the gaps dramatically change when reaching 300 sensors. Notably, when the number of sensors is 300, Fuzzy Q-charging extends the network lifetime infinitely, while Q-charging, INMA, and GSA can only attain a limited network lifetime. The reason is that when the number of sensors is small, the traffic imposed on each sensor is large. Therefore, the energy consumption rate of all sensors becomes immensely high. In all charging algorithms, the MC cannot charge to all sensors in time. That explains why the performance gap between the algorithms is insignificant with a small number of sensors. When the sensor number becomes sufficiently large, the energy consumption rate is slower. Fuzzy Q-charging favors the sensors with more essential roles in covering targets and transferring data to the base station. It can hence maintain the essential sensors' lifetime and ensure all monitored targets. Other algorithms do not concurrently consider the target coverage and connectivity constraints. Therefore, the essential sensors may not be charged in time, causing some targets to be unmonitored.

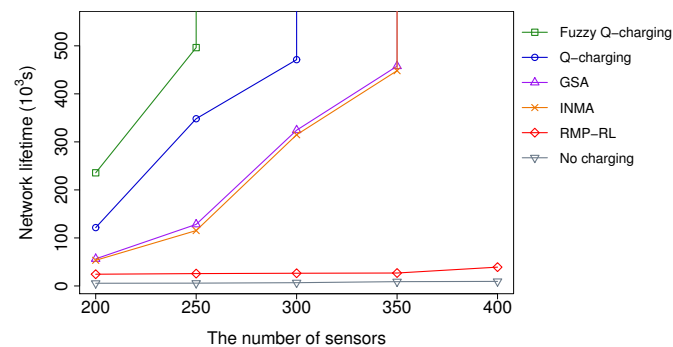


Figure 7. Network lifetime vs. the number of sensors.

Compared to Q-charging, i.e., the second-best charging algorithm, at the condition of fewer than 300 nodes, Fuzzy Q-learning's network lifetime is 1.4 times better. In the case of 300 sensors, Fuzzy Q-learning's network lifetime is infinite, while Q-learning's one is only prolonged to about 500×10^3 s. This results proves the effectiveness of our algorithm, which uses Fuzzy logic to automatically adjust the charging energy level. Concerning the two other algorithms, GSA, INMA, and RMP-RL, Fuzzy Q-learning improves the network lifetime to more than 4.3 times, at the condition of fewer than 300 nodes. Moreover, when the number of sensors reaches 300 nodes, GSA, INMA, and RMP-RL only prolong the network lifetime to less than about 448×10^3 seconds, while that of Fuzzy Q-learning is infinite.

Among all the algorithms, RMP-RL shows the worst performance. The reason is that RMP-RL relies on the Deep reinforcement learning technique, which necessitates the training of a deep learning model. Unfortunately, a Deep reinforcement model typically takes a long time to converge. As a result, its early-stage performance is often poor, resulting in many dead nodes. This behavior is clearly demonstrated in Section 5.2.4.

5.2.2. Impacts of the Number of Targets

We evaluate the target number's impact in a scenario with 300 nodes and the packet generation probability of 0.3. We investigate the network lifetime variation when the number of targets increases from 100 to 300. To do so, we first generate 300 targets randomly in the network region. After that, the number of targets is adjusted by subtracting 50, 100, 150, and 200 targets at random from the initial set. The results are presented in Figure 8. As shown, Fuzzy Q-charging performs much better than the other algorithms. As expected, as the number of targets increases, the network lifetime achieved by all algorithms drops. The reason for this is because when the target number increases, the volume of traffic on the sensors increases as well. When the number of targets is significantly large (i.e., more than 250), the energy consumption rate of sensors becomes too high. As a result, the MC is unable to charge the sensors in time. As a consequence, no algorithm can considerably increase the network lifetime.

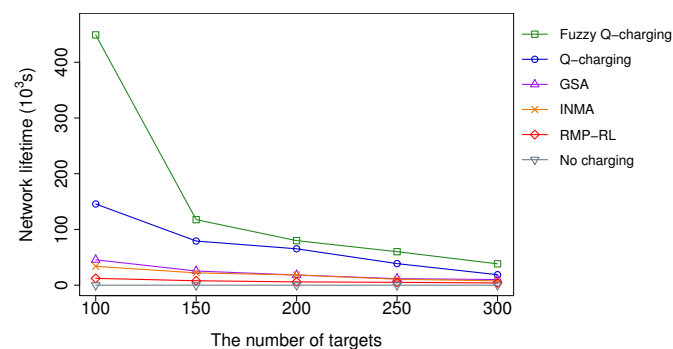


Figure 8. Network lifetime vs. the number of targets.

When compared to the no-charge approach, Fuzzy Q-charging extends the network lifetime by more than 9.8 in all scenarios. Fuzzy Q-charging has average performance gaps of 6.3, 6.0, and 16.4 when compared to GSA, INMA, and RMP-RL, respectively. In the best case, the performance gaps of Fuzzy Q-charging to GSA, INMA, and RMP-RL are 12.6, 11.5, and 33.9, respectively. Compared to Q-charging, Fuzzy Q improves the network lifetime by 1.9 times in average and 3.1 times in the best case. The improvement of Fuzzy Q-charging is because Fuzzy Q-charging favors the sensors with more essential roles in covering targets and transferring data to the base station. It can hence maintain the essential sensors' lifetime and ensure all monitored targets. Other algorithms do not concurrently consider the target coverage and connectivity constraints. Therefore, the essential sensors may not be charged in time, causing some targets to be unmonitored.

5.2.3. Impacts of the Packet Generation Frequency

Figure 9 shows the resulting impact of the packet generation probability on the network lifetime. In this experiment, the number of sensors and targets is set to 300 and 150, respectively. The location of the targets are generated randomly. In all algorithms, the network lifetime tends to decrease when the packet generation probability increases. When the probability is too large (i.e., being more than 0.25), all sensors' energy consumption rate (especially sensors in the base station's vicinity) becomes fast. Therefore, the sensors' batteries exhaust quickly. In such a critical case, the difference between the algorithms is minor. We can see the improvement of Fuzzy Q-charging over the existing algorithms clearly under the condition of small packet generation probability. Notably, when the probability is 0.1, Fuzzy Q-charging prolongs the network lifetime infinitely, while the others cannot. When the probability is greater than 0.1, Fuzzy Q-charging's network lifetime is 1.4 times more than Q-learning, 5.0 times more than INMA's, 5.1 times more than GSA's, and 21.1 times more than RMP-RL, on average. The performance gaps between Fuzzy Q-charging and the other algorithms decrease when the packet generation probability increases. Even when the probability is 0.25, the network lifetime's ratio achieved by

Fuzzy Q-charging is 1.4, 4.5, 4.5, and 10.8 times better than Q-charging, INMA, GSA, and RMP-RL, respectively.

In summary, we can conclude Fuzzy Q-charging outperforms the existing algorithms. Moreover, the performance gaps between Fuzzy Q-charging and the others increase when the number of sensors increases, the number of targets decreases, or the packet generation probability decreases.

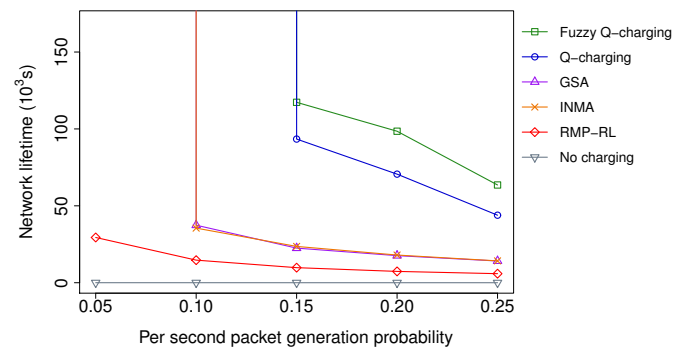


Figure 9. Network lifetime vs. the packet generation frequency.

5.2.4. Non-Monitored Targets and Dead Sensors over Time

We present the number of non-monitored targets and the number of sensors over time caused by different algorithms in Figure 10a,b, respectively. In Figure 10b, when the time elapses, the number of sensors exhausting energy and becoming dead nodes increases. Accordingly, more targets become non-monitored, as shown in Figure 10a. Fuzzy Q-learning outperforms the other algorithms concerning both metrics. There is a huge gap between the performance of Fuzzy Q-learning and the others in Figure 10a. Fuzzy Q-charging with better charging strategies slows down the increase of non-monitored targets over time. Another interesting observation is that while the gaps between the number of dead sensors caused by using Fuzzy Q and that of INMA and GSA are relatively small (Figure 10b), the gaps concerning the number of non-monitored targets are huge (Figure 10a). The reason is INMA and GSA do not consider target coverage and connectivity constraints. Therefore, the next charging location is not optimized to prioritize the sensors with an essential role. Those sensors may be dead in INMA and GSA, leading to the targets being non-monitored. Meanwhile, in Fuzzy Q-charging, the charging location determination algorithm can identify the sensor nodes with a specific priority. Therefore, the dead sensors caused by Fuzzy Q-charging are the less important ones. In many cases, the death nodes may not affect or have minor impacts on the monitored targets.

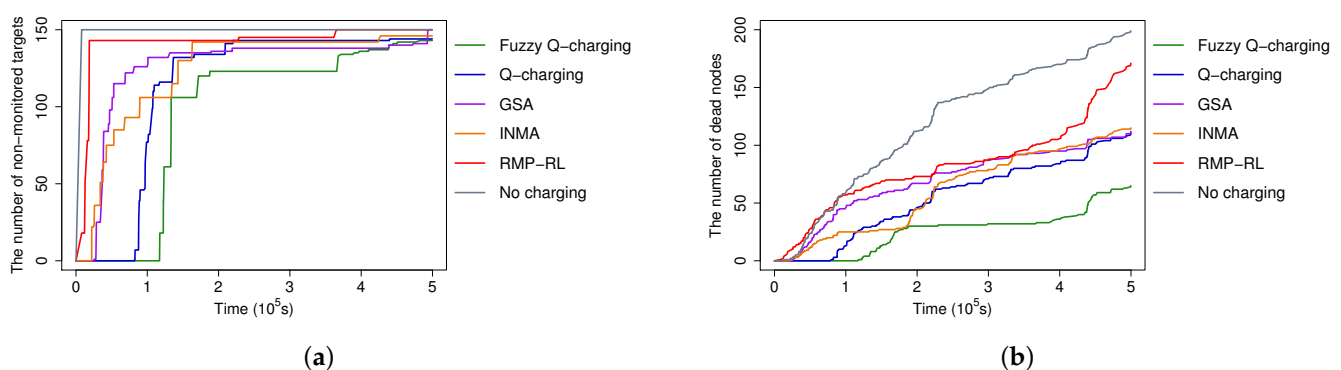


Figure 10. A comparison of non-monitored targets and dead sensors over time. (a) Non-monitored targets over time. (b) Dead sensors over time.

6. Conclusions and Future Work

This paper addresses optimizing the MC's charging schedule in WRSNs, which considers target coverage and connectivity constraints. Unlike the existing approaches, ours took into account the charging location and the charging time in the newly proposed Fuzzy Q-charging. Fuzzy Q-charging has an optimal charging time determination algorithm that relies on Fuzzy logic to adjust the energy charging level dynamically. The algorithm has been utilized at every charging point to maximize the number of alive sensors. Moreover, Fuzzy Q-charging uses Q-learning in an optimal charging scheme to maximize the target number. We have extensively evaluated Fuzzy Q-charging in comparison to the previous charging schemes in WRSNs. The evaluation results show that Fuzzy Q-charging outperforms the others. Specifically, Fuzzy Q-charging prolongs the network lifetime infinitely in certain conditions of the target and sensor numbers, while the other algorithms cannot. In other cases, Fuzzy Q-charging extends the network lifetime by 6.8 times on average and 33.9 times in the best case, compared to the existing algorithms. In the future, we plan to extend this work to handle the WRSNs with multiple mobile chargers.

Author Contributions: Conceptualization: V.Q.L., P.L.N.; Methodology: P.L.N., K.N.; Software, validation: V.Q.L., A.D.N., T.H.N.; Formal analysis, investigation: P.L.N., V.Q.L., A.D.N.; Writing—original draft preparation, Visualization: A.D.N., V.Q.L.; Writing—review and editing: P.L.N., K.N.; Project administration: K.N., T.H.N.; Resources: T.H.N.; Funding acquisition: K.N., P.L.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by Ministry of Education and Training of Vietnam under grant number B2020-BKA-13. This research was partly supported by ROIS NII Open Collaborative Research 2021-21S1201, JSPS KAKENHI Grant Number 19K20251, 20H04174. La Van Quan was funded by Vingroup Joint Stock Company and supported by the Domestic Master/Ph.D. Scholarship Programme of Vingroup Innovation Foundation (VINIF), Vingroup Big Data Institute (VINBIGDATA), code VINIF.2020.ThS.BK.03.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Han, G.; Yang, X.; Liu, L.; Guizani, M.; Zhang, W. A Disaster Management-Oriented Path Planning for Mobile Anchor Node-Based Localization in Wireless Sensor Networks. *IEEE Trans. Emerg. Top. Comput.* **2017**, *8*, 115–125. [[CrossRef](#)]
2. Ojha, T.; Misra, S.; Raghuvanshi, N.S. Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges. *Comput. Electron. Agric.* **2015**, *118*, 66–84. [[CrossRef](#)]
3. Le Nguyen, P.; Nguyen, K.; Vu, H.; Ji, Y. TELPAC: A time and energy efficient protocol for locating and patching coverage holes in WSNs. *J. Netw. Comput. Appl.* **2019**, *147*, 102439. [[CrossRef](#)]
4. Le Nguyen, P.; Ji, Y.; Le, K.; Nguyen, T. Load balanced and constant stretch routing in the vicinity of holes in WSNs. In Proceedings of the 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 12–15 January 2018; pp. 1–6.
5. Hanh, N.T.; Le Nguyen, P.; Tuyen, P.T.; Binh, H.T.T.; Kurniawan, E.; Ji, Y. Node placement for target coverage and network connectivity in WSNs with multiple sinks. In Proceedings of the 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 12–15 January 2018; pp. 1–6.
6. Lyu, Z.; Wei, Z.; Pan, J.; Chen, H.; Xia, C.; Han, J.; Shi, L. Periodic charging planning for a mobile WCE in wireless rechargeable sensor networks based on hybrid PSO and GA algorithm. *Appl. Soft Comput.* **2019**, *75*, 388–403. [[CrossRef](#)]
7. Jiang, G.; Lam, S.; Sun, Y.; Tu, L.; Wu, J. Joint Charging Tour Planning and Depot Positioning for Wireless Sensor Networks Using Mobile Chargers. *IEEE/ACM Trans. Netw.* **2017**, *25*, 2250–2266. [[CrossRef](#)]
8. Ma, Y.; Liang, W.; Xu, W. Charging Utility Maximization in Wireless Rechargeable Sensor Networks by Charging Multiple Sensors Simultaneously. *IEEE/ACM Trans. Netw.* **2018**, *26*, 1591–1604. [[CrossRef](#)]
9. Xu, W.; Liang, W.; Kan, H.; Xu, Y.; Zhang, X. Minimizing the Longest Charge Delay of Multiple Mobile Chargers for Wireless Rechargeable Sensor Networks by Charging Multiple Sensors Simultaneously. In Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 7–10 July 2019; pp. 881–890.

10. Lin, C.; Zhou, Y.; Ma, F.; Deng, J.; Wang, L.; Wu, G. Minimizing Charging Delay for Directional Charging in Wireless Rechargeable Sensor Networks. In Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 1819–1827.
11. Feng, Y.; Liu, N.; Wang, F.; Qian, Q.; Li, X. Starvation avoidance mobile energy replenishment for wireless rechargeable sensor networks. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 22–27 May 2016; pp. 1–6.
12. Lin, C.; Zhou, J.; Guo, C.; Song, H.; Wu, G.; Obaidat, M.S. TSCA: A Temporal-Spatial Real-Time Charging Scheduling Algorithm for On-Demand Architecture in Wireless Rechargeable Sensor Networks. *IEEE Trans. Mobile Comput.* **2018**, *17*, 211–224. [[CrossRef](#)]
13. Lin, C.; Sun, Y.; Wang, K.; Chen, Z.; Xu, B.; Wu, G. Double warning thresholds for preemptive charging scheduling in Wireless Rechargeable Sensor Networks. *Comput. Netw.* **2019**, *148*, 72–87. [[CrossRef](#)]
14. Tomar, A.; Muduli, L.; Jana, P.K. A Fuzzy Logic-based On-demand Charging Algorithm for Wireless Rechargeable Sensor Networks with Multiple Chargers. *IEEE Trans. Mobile Comput.* **2020**, *1233*, 2715–2727. [[CrossRef](#)]
15. Cao, X.; Xu, W.; Liu, X.; Peng, J.; Liu, T. A Deep Reinforcement Learning-Based On-Demand Charging Algorithm for Wireless Rechargeable Sensor Networks. *Ad Hoc Netw.* **2020**, *110*, 102278. [[CrossRef](#)]
16. Zhu, J.; Feng, Y.; Liu, M.; Chen, G.; Huang, Y. Adaptive online mobile charging for node failure avoidance in wireless rechargeable sensor networks. *Comput. Netw.* **2018**, *126*, 28–37. [[CrossRef](#)]
17. Kaswan, A.; Tomar, A.; Jana, P.K. An efficient scheduling scheme for mobile charger in on-demand wireless rechargeable sensor networks. *J. Netw. Comput. Appl.* **2018**, *114*, 123–134. [[CrossRef](#)]
18. Xu, W.; Liang, W.; Jia, X.; Xu, Z. Maximizing Sensor Lifetime in a Rechargeable Sensor Network via Partial Energy Charging on Sensors. In Proceedings of the 2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), London, UK, 27–30 June 2016; pp. 1–9.
19. Krishnamurthi, R.; Goyal, M. Hybrid Neuro-fuzzy Method for Data Analysis of Brain Activity Using EEG Signals. *Soft Comput. Signal Process.* **2019**, *900*, 165–173.
20. Behera, S.K.; Jena, L.; Rath, A.K.; Sethy, P.K. Disease Classification and Grading of Orange Using Machine Learning and Fuzzy Logic. In Proceedings of the 2018 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 3–5 April 2018; pp. 678–682.
21. Yang, C.; Jiang, Y.; Na, J.; Li, Z.; Cheng, L.; Su, C. Finite-Time Convergence Adaptive Fuzzy Control for Dual-Arm Robot with Unknown Kinematics and Dynamics. *IEEE Trans. Fuzzy Syst.* **2019**, *27*, 574–588. [[CrossRef](#)]
22. Castillo, O.; Amador-Angulo, L. A generalized type-2 fuzzy logic approach for dynamic parameter adaptation in bee colony optimization applied to fuzzy controller design. *Inf. Sci.* **2018**, *460–461*, 476–496. [[CrossRef](#)]
23. Yang, C.S.; Kim, C.K.; Moon, J.; Park, S.; Kang, C.G. Channel Access Scheme With Alignment Reference Interval Adaptation (ARIA) for Frequency Reuse in Unlicensed Band LTE: Fuzzy Q-Learning Approach. *IEEE Access* **2018**, *6*, 26438–26451. [[CrossRef](#)]
24. Jain, A.; Goel, A. Energy Efficient Fuzzy Routing Protocol for Wireless Sensor Networks. *Wirel. Pers. Commun.* **2020**, *110*, 1459–1474. [[CrossRef](#)]
25. Krishnaswamy, V.; Manvi, S. Fuzzy and PSO Based Clustering Scheme in Underwater Acoustic Sensor Networks Using Energy and Distance Parameters. *Wirel. Pers. Commun.* **2019**, *108*, 1529–1546. [[CrossRef](#)]
26. Ghosh, N.; Banerjee, I.; Sherratt, R. On-demand fuzzy clustering and ant-colony optimisation based mobile data collection in wireless sensor network. *Wirel. Netw.* **2019**, *25*, 1829–1845. [[CrossRef](#)]
27. Wan, R.; Xiong, N.; Hu, Q.; Wang, H.; Shang, J. Similarity-aware data aggregation using fuzzy c-means approach for wireless sensor networks. *EURASIP J. Wirel. Commun. Netw.* **2019**, *2019*, 59. [[CrossRef](#)]
28. Kofinas, P.; Dounis, A.; Vouros, G. Fuzzy Q-Learning for multi-agent decentralized energy management in microgrids. *Appl. Energy* **2018**, *219*, 53–67. [[CrossRef](#)]
29. Avin, N.; Sharma, R. A fuzzy reinforcement learning approach to thermal unit commitment problem. *Neural Comput. Appl.* **2019**, *31*, 737–750.
30. Van Quan, L.; Nguyen, P.L.; Nguyen, T.H.; Nguyen, K. Q-learning-based, Optimized On-demand Charging Algorithm in WRSN. In Proceedings of the 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 24–27 November 2020; pp. 1–8.
31. He, S.; Chen, J.; Jiang, F.; Yau, D.K.Y.; Xing, G.; Sun, Y. Energy Provisioning in Wireless Rechargeable Sensor Networks. *IEEE Trans. Mob. Comput.* **2013**, *12*, 1931–1942. [[CrossRef](#)]
32. Karp, B.; Kung, H.T. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, Boston, MA, USA, 6–11 August 2000; pp. 243–254.
33. Bulusu, N.; Heidemann, J.; Estrin, D. GPS-less low-cost outdoor localization for very small devices. *IEEE Pers. Commun.* **2000**, *7*, 28–34. [[CrossRef](#)]
34. Fang, Q.; Gao, J.; Guibas, L.J. Locating and bypassing routing holes in sensor networks. In Proceedings of the IEEE INFOCOM, Hong Kong, China, 7–11 March 2004; pp. 2458–2468.
35. Yu, F.; Park, S.; Tian, Y.; Jin, M.; Kim, S. Efficient Hole Detour Scheme for Geographic Routing in Wireless Sensor Networks. In Proceedings of the VTC Spring 2008-IEEE Vehicular Technology Conference, Marina Bay, Singapore, 11–14 May 2008; pp. 153–157.

36. Kim, S.; Kim, C.; Cho, H.; Yim, Y.; Kim, S. Void Avoidance Scheme for Real-Time Data Dissemination in Irregular Wireless Sensor Networks. In Proceedings of the 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA), Crans-Montana, Switzerland, 23–25 March 2016; pp. 438–443.
37. Tian, Y.; Yu, F.; Choi, Y.; Park, S.; Lee, E.; Jin, M.; Kim, S.-H. Energy-Efficient Data Dissemination Protocol for Detouring Routing Holes in Wireless Sensor Networks. In Proceedings of the 2008 IEEE International Conference on Communications, Beijing, China, 19–23 May 2008; pp. 2322–2326.
38. Li, F.; Zhang, B.; Zheng, J. Geographic hole-bypassing forwarding protocol for wireless sensor networks. *IET Commun.* **2011**, *5*, 737–744. [[CrossRef](#)]
39. Won, M.; Stoleru, R. A Low-Stretch-Guaranteed and Lightweight Geographic Routing Protocol for Large-Scale Wireless Sensor Networks. *ACM Trans. Sens. Netw. (TOSN)* **2014**, *11*, 18:1–18:22. [[CrossRef](#)]
40. Nguyen, P.L.; Ji, Y.; Liu, Z.; Nguyen, K.V. Distributed Hole-Bypassing Protocol in WSNs with Constant Stretch and Load Balancing. *Comput. Netw.* **2017**, *129*, 232–250. [[CrossRef](#)]
41. Nguyen, P.; Ji, Y.; Trung, N.T.; Hung, N.T. Constant stretch and load balanced routing protocol for bypassing multiple holes in wireless sensor networks. In Proceedings of the 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 30 October–1 November 2017; pp. 1–9.
42. Huang, H.; Yin, H.; Min, G.; Zhang, J.; Wu, Y.; Zhang, X. Energy-Aware Dual-Path Geographic Routing to Bypass Routing Holes in Wireless Sensor Networks. *IEEE Trans. Mob. Comput.* **2018**, *17*, 1339–1352. [[CrossRef](#)]
43. Fu, L.; Cheng, P.; Gu, Y.; Chen, J.; He, T. Optimal Charging in Wireless Rechargeable Sensor Networks. *IEEE Trans. Veh. Technol.* **2016**, *65*, 278–291. [[CrossRef](#)]
44. Mohammed, S.L. Distance Estimation Based on RSSI and Log-Normal Shadowing Models for ZigBee Wireless Sensor Network. *Eng. Technol. J.* **2016**, *34*, 2950–2959.