mathematics

MDPI

Article

A Q-Learning Based Target Coverage Algorithm for Wireless Sensor Networks

Peng Xiong [1], Dan He [2,*] and Tiankun Lu [3]

[1] Kaiserslautern Institute for Intelligent Manufacturing, Shanghai Dianji University, Shanghai 201308, China; xiongp@sdju.edu.cn
[2] School of Information Engineering, Nanchang Hangkong University, Nanchang 330063, China
[3] Industrial Technology Center, Shanghai Dianji University, Shanghai 201308, China; lutk@sdju.edu.cn
* Correspondence: hdnchu@163.com

Abstract: To address the problems of unclear node activation strategy and redundant feasible solutions in solving the target coverage of wireless sensor networks, a target coverage algorithm based on deep Q-learning is proposed to learn the scheduling strategy of nodes for wireless sensor networks. First, the algorithm abstracts the construction of feasible solutions into a Markov decision process, and the smart body selects the activated sensor nodes as discrete actions according to the network environment. Second, the reward function evaluates the merit of the smart body's choice of actions in terms of the coverage capacity of the activated nodes and their residual energy. The simulation results show that the proposed algorithm intelligences are able to stabilize their gains after 2500 rounds of learning and training under the specific designed states, actions and reward mechanisms, corresponding to the convergence of the proposed algorithm. It can also be seen that the proposed algorithm is effective under different network sizes, and its network lifetime outperforms the three greedy algorithms, the maximum lifetime coverage algorithm and the self-adaptive learning automata algorithm. Moreover, this advantage becomes more and more obvious with the increase in network size, node sensing radius and carrying initial energy.

Keywords: wireless sensor network; Q-learning; target coverage

MSC: 37M99

check for updates

Academic Editor: Ioannis Tsoulos

Received: 26 November 2024
Revised: 23 January 2025
Accepted: 3 February 2025
Published: 5 February 2025

Citation: Xiong, P.; He, D.; Lu, T. A Q-Learning Based Target Coverage Algorithm for Wireless Sensor Networks. *Mathematics* **2025**, *13*, 532. https://doi.org/10.3390/math13030532

1. Introduction

With the development of new technology in the field of electronics and communications, a micro-wireless sensor with sense, wireless communication and data processing capability has been used widely. Because of the low-cost and low-power features, these sensors can be organized into the wireless sensor networks (WSN) to monitor and sense a variety of environmental information (such as temperature, humidity, etc.). Today, WSN, with many micro-sensor nodes, has become an important calculating platform [1].

Because of the features of fast deployment, large scale, self-organization and robustness, WSN has been widely used in many fields. Here, coverage quality and network lifetime are obviously the most basic and important problems of WSN. How to extend the network lifetime under the coverage requirements is defined as the coverage problem of WSN [2]. It can also be described as the problem of node deployment with guaranteed sensor network connectivity in a defined monitoring area. Coverage optimization is one of the key challenges in optimizing network performance and resource utilization, which is directly related to network robustness, fault tolerance and survival time.

According to different application scenarios, constraints and target features, the coverage problem can be divided into target coverage, area coverage and fence coverage. The target coverage, also known as the point coverage, focuses on a set of spatial nodes that are discretely distributed in a designed space. The area coverage requires that the sensors arranged in a designed area cover the entire area, i.e., the entire monitoring area where the sensors are located. The fence coverage, on the other hand, mainly examines the relationship between the mobile target and the monitoring area composed of wireless sensor nodes, and this type of problem no longer focuses on the relationship between the sensor nodes and the monitoring area.

In this paper, we propose the use of Deep Q-learning in reinforcement learning to solve the target coverage problem based on the Markovian nature of the problem. The corresponding algorithms and the reward and loss functions adapted to them are designed. The network model, energy model and sensor model are constructed. Finally, by comparing the proposed algorithms with the traditional algorithms for target coverage in several scenarios, it is confirmed that the proposed algorithms are effective in improving the lifetime of energy-limited wireless sensor networks.

2. Related Works

The target coverage has a rich application, and its solution methods are mainly categorized into integer planning methods, game theory-based methods and smart algorithms. Reference [3] has researched the target coverage problem of WSN with multiple sensing radii and solved it by modeling with the column generation method. Reference [4] modeled the coverage connectivity of WSN with the column generation method, and designed a genetic algorithm to replace the slow solution speed of the column generation method, which effectively increased the solution speed. Reference [5] proposed the concept of key target, made the strategy of prioritizing the targets covered by fewer sensors in the network and designed a hybrid column generation method for solving the problem, which extended the lifetime of network. Reference [6] proposed a solution to the target coverage problem of directed sensor networks based on the column generation method for the features of video sensors with limited sensing range. In Reference [7], Shahrokhzadeh et al. considered sensor nodes as nodes participating in the game, modeled the target coverage using a non-cooperative game model and proved that the game necessarily exists a Nash equilibrium. Reference [8] modeled the target coverage of directed sensor networks with the game model, proposed a strategy for node sensing direction change and designed a distributed algorithm to reduce the coverage loophole to solve the problem. Cardei et al. [9] were the first to study the target coverage using heuristic algorithms, and proposed a sleep scheduling strategy, where only some sensor nodes are activated to ensure the quality of the target coverage, and the other sensor nodes save energy by sleep. The activated sensor nodes are selected by a greedy strategy, taking into account the number of coverage targets and the remaining energy of the sensor nodes. Reference [10] proposed an active redundancy strategy for the target coverage of directed sensor networks, which requires that each target needs to be covered by multiple sensor nodes to ensure that the information will not be lost, and adjusts the sensing angle of the directed sensor nodes by using a priori evolutionary algorithms and a posteriori evolutionary algorithms, respectively, so that the number of targets covered by redundancy is improved as much as possible under the premise of ensuring that all targets are covered. Reference [11] proposed a distributed algorithm-based way to solve the target coverage, assuming that each sensor node has access to the operating state of its neighboring nodes, which helps to change the probability of a sensor node to be selected, reduce the occurrence of redundancy and extend the network lifetime. Reference [12] proposed the concept of region of interest (ROI), transformed the target coverage problem into a matching problem by a Bipartite Graph, proposed a heuristic algorithm and three different approximate optimization algorithms and completed the target coverage task by selecting the sensor node with the largest interest value. Reference [13] proposed an improved sparrow search algorithm to optimize the coverage of WSN, which improves the initialization strategy by introducing the set of good points to avoid the premature failure of the algorithm, and optimizes the algorithm's optimization mode by adding a dynamic learning strategy in the sparrow search process. Reference [14] proposes a node selection method based on learning automata, which increases the activation probability of sensor nodes that meet the energy constraints, reduces the selection probability of redundant sensor nodes and extends the network lifetime by constructing more feasible solutions. Reference [15] proposed a genetic algorithm to solve the K-target coverage problem for directed sensor nodes, where each chromosome is used to represent the set of all solutions that can be constructed in the network, and the population is gradually optimized by crossover, mutation and so on to construct more feasible solutions.

Integer planning ways and the ways based on game theory ignore the inner features of the target coverage during the solution process, and search for the optimal solution by designing a large number of constraints, which could lead to slow convergence. Heuristic algorithms are closely related to the features of the target coverage, and the algorithms activate the most suitable sensor nodes to cover the target according to the current state of network. These algorithms are fast, but the causes affecting the lifetime of WSN are analyzed in a one-sided manner, the self-learning ability is insufficient and the quality of the solution needs to be improved. Evolutionary algorithms are easier to construct solutions, and the evolutionary process can implicitly reveal the inner features of the coverage; but, the construction of an initial solution often depends on greedy algorithms, which are prone to problems such as slow solution speed and falling into local optimization.

To address the above problems, this paper proposes a sensor scheduling algorithm based on Deep Q-learning. First, the current network state is described according to three features: the energy of sensors in the network, whether they are activated or not and the coverage of targets. Second, the smart body selects the activated sensor nodes by Q-learning. Again, a reward function is designed from the relationship between the number of targets covered by the sensor nodes and its own energy consuming, which guides the environment to give the smart body an instantaneous reward. And lastly, the smart body enters into a new network state. The process is iterated until no new feasible solutions can be constructed. With the goal of obtaining the maximum cumulative gain, the smart body continuously learns the sensor scheduling strategy, constructs more feasible solutions and extends the network lifetime.

3. Q-Learning Principles

3.1. Markov Decision Process

As one of the mathematical models of sequential decision making, the Markov decision process (MDPs) is applied to reinforcement learning problems and is the mathematical basis of reinforcement learning. MDPs has factors such as action space, state space, reward function, state transfer matrix and discount factor [16]. MDPs is shown in Figure 1.

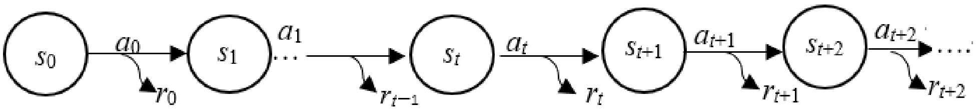

Figure 1. Diagram of the Markov decision process.

Observing the state s_t at the moment t, the smart body chooses to execute the action a_t in the action set A. At this time, the smart body receives an instantaneous reward r_t from the environment, and since the smart body executes the action a_t, the state is transformed to s_{t+1} according to the state transfer probability $P_s^{a_t}(s_t, s_{t+1})$. Next, the smart body chooses the next action a_{t+1} in the set of actions A according to the current state s_{t+1}, and after executing this action, the smart body again receives the instantaneous reward r_{t+1} from the environment, and the state changes to s_{t+2} again according to the transfer probability. By continuing this process, the loop repeats until the final state is reached.

3.2. Q-Learning Based on MDPs

Reinforcement learning is an important branch of machine learning. A smart body (Agent) learns an action strategy π to maximize the cumulative rewards in interaction with the environment. Given a moment t, the smart body observes the current state s_t in the state space S and selects the action a_t in the action space A to interact with the environment [17]. The environment gives the smart body a reward r_t and migrates to a new state s_{t+1}. The smart body gradually learns to refine π according to r_t. Figure 2 shows the framework of the principle of the reinforcement learning.

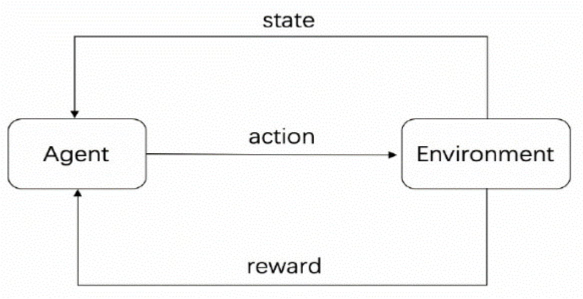

Figure 2. Framework of Reinforcement learning.

Q-learning is a value-based model-free reinforcement learning method. The expected value of the cumulative rewards U_t that can be obtained by the smart body after choosing a_t according to π in s_t is evaluated by the action-value function Q_π (s_t, a_{t}), which is calculated as in Formula (1).

$$Q_\pi(s_t, a_t) = E_{S_{t+1}, A_{t+1}, \ldots, S_n, A_n}[U_t | S_t = s_t, A_t = a_t] \tag{1}$$

where U_t is the discount reward; γ ∈[0, 1] is the discount factor, denoted by Formula (2).

$$U_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots + \gamma^{n-t} r_n \tag{2}$$

From Formula (1), Q_π (s_t, a_{t}) depends on π. Assuming that π* is the optimal strategy, Q* (s_t, a_{t}) is the optimal action-value function, at which point Q* (s_t, a_{t}) depends only on s_t and a_{t} and is no longer correlated with π, i.e.,:

$$\pi^* = \underset{\pi}{\mathrm{argmax}} Q_\pi(s_t, a_t) \tag{3}$$

$$Q^*(s_t, a_t) = \max Q_{\pi^*}(s_t, a_t) \tag{4}$$

Initially, Q-learning appeared based on tables for environments with low dimensionality. Reference [18] proposed Deep Q-network (DQN) to estimate the optimal action-value function in complex environments using a neural network Q (s_t, a_t; ω), where ω denotes the parameters of the neural network, r_t after a smart body performs an action, a_t is a deterministic value, so $r_t + \gamma \max_a Q(s_{t+1}, a_{t+1}; \omega)$ is used as the training target of DQN. DQN is trained using the temporal difference (TD) algorithm [19] with the following loss function.

$$L(\omega) = \frac{1}{2}\left(Q(s_t, a_t; \omega) - \left(r_t + \gamma \max_a Q(s_{t+1}, a_{t+1}; \omega)\right)\right)^2 \tag{5}$$

4. Modeling and Problem Description

4.1. Network Model

We randomly deployed n sensor nodes to cover m static targets in a two-dimensional space of $X \times Y$. The set of sensor nodes is $P = \{p_1, p_2, \ldots, p_n\}$, and $T = \{t_1, t_2, \ldots, t_m\}$ is the set of targets. In this paper, it is assumed that the sensor nodes in the network have the same coverage capability and initial energy, and the nodes' locations do not change after deployment. Figure 3 shows a network of five sensor nodes (represented by the black dot) and three targets (represented by the star symbol).

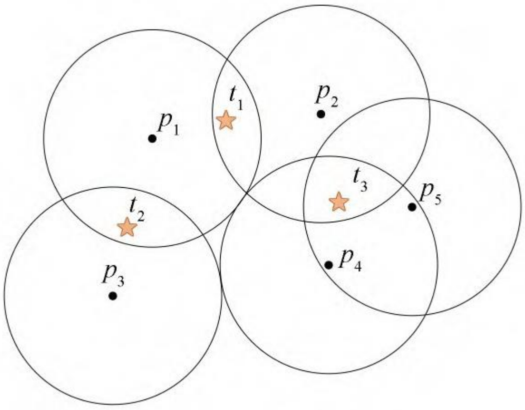

Figure 3. A sensor network containing five sensor nodes and three targets to be covered.

4.2. Sensor Model

The sensing range of a sensor node is a circular area centered on the sensor node with radius k. The coverage relationship between the sensor node p_i and the target t_j is modeled using the deterministic coverage model [20], i.e., the coverage relationship between p_i and t_j is only related to the distance d between them. The locations of p_i and t_j are denoted by two-dimensional coordinates as (x_i, y_i) and (x_j, y_j), respectively, and the Euclidean distance d between them is calculated as follows.

$$d = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{6}$$

When $d \le k$, this paper notates that p_i covers t_j. c_i records the set of targets that can be covered by sensor node p_i. From Figure 3, $c_1 = \{t_1, t_2\}$, $c_2 = \{t_1, t_3\}$, $c_3 = \{t_2\}$, $c_4 = \{t_3\}$, $c_5 = \{t_3\}$.

4.3. Energy Model

All sensor nodes in the network have an initial energy with capacity of E units, where $E > 1$. Sensor nodes have two states, sleepy and active [21]. Sensor nodes in the sleeping state do not consume energy and cannot cover the targets in the network; sensor nodes in the activated state consume one unit of energy per unit of time and continuously sense the targets in their coverage area. In this paper, it is assumed that the sensor nodes consume negligible energy during the two state transitions. The sensor nodes are considered dead after energy depletion and cannot be reactivated.

The meanings of the variables in this paper are as follows (Table 1).

Table 1. Variable Declaration.

Variables	Meaning
n	the number of sensors
P	the set of sensor nodes
p_i	the sensor node i
m	the number of targets
T	the set of targets
t_j	the target j
c_i	the set of targets that can be covered by the sensor node p_i
CS	the set of feasible solutions
cs_i	the ith feasible solution
act_x	the time at which the feasible solution is activated
L	the total number of feasible solutions
E	the initial energy carried by the sensor node
e_{ix}	the energy consumed by p_i node per unit time

4.4. Related Definitions and Theorems

In order to facilitate the discussion and analysis of the proposed scheme, the following definitions and theorems are made.

Definition 1. *The feasible solutions are a subset of P that covers all the targets in the network.*

The set of feasible solutions is denoted as $CS = \{cs_1, cs_2, \ldots, cs_l\}$, and any feasible solution cs_i should satisfy the following formula.

$$\bigcup_{p_n \in cs_i} c_n = T \tag{7}$$

Thus, the feasible solutions in Figure 3 are $cs_1 = \{p_1, p_4\}$, $cs_2 = \{p_1, p_2, p_3\}$, etc.

Definition 2. *A sensor node that covers zero number of targets individually in the feasible solution is called a redundant sensor node.*

All the targets covered by the redundant sensor nodes can be covered by other sensor nodes in the feasible solution. As shown in Figure 2, there exists a feasible solution $\{p_1, p_3, p_5\}$, and the target t_2 covered by sensor node p_3 is also covered by sensor p_1. Therefore, sensor p_3 is a redundant sensor node.

Definition 3. *The feasible solution in the presence of redundant sensor nodes is called a redundant feasible solution.*

Redundant feasible solutions consume additional sensor energy, which seriously affects the network lifetime.

Definition 4. *The feasible solution without redundant sensor nodes is called a non-redundant feasible solution.*

Definition 5. *The time that a WSN continues to cover all targets is called the network lifetime.*

Definition 6. *Constructing and scheduling the feasible solution in a WSN such that the network lifetime is maximized is called the target coverage problem.*

Target Coverage Modeling:

$$\max \sum_{x=1}^{L} act_x \tag{8}$$

$$\max \sum_{x=1}^{L} act_x \cdot e_{ix} \leq E, \quad \forall i \in 1, 2, \ldots, n \tag{9}$$

$$act_x \geq 0 \tag{10}$$

where L denotes the total number of feasible solutions, act_x denotes the length of time that a feasible solution is activated to work, and Formula (9) denotes that the sum of the energy consumed by any sensor p_i in each feasible solution does not exceed its own initial energy E, e_{ix} denotes the energy consumed by p_i per unit of time, and Formula (10) denotes that any feasible solution has non-negative operating time.

Theorem 1. *When an optimal solution is obtained for the target coverage problem of WSN, the activated feasible solutions can be all non-redundant feasible solutions.*

Proof of Theorem 1. Let CS = {cs$_1$, cs$_2$, …, cs_l} be the set of feasible solutions activated at the maximum of the network lifetime. For any feasible solution cs_i in CS, if cs_i is a redundant feasible solution, there must exist a non-redundant feasible solution cs'_i corresponding to it; if cs_i is a non-redundant feasible solution, then cs'_i = cs_i. Therefore, there exists a non-redundant feasible solution set CS' = {cs'_1, cs'_2, …, cs'_l}, which is the same as CS = {cs_1, cs_2, …, cs_l}, which has the same network lifetime as CS = {cs_1, cs_2, …, cs_l}. □

5. Algorithm Description

In this paper, we assume that each sensor node consumes the same amount of energy per unit of time and each feasible solution has the same working time. The core of solving the target coverage becomes a construction of a larger number of feasible solutions. In the process of constructing feasible solutions, the current network state is only related to the previous state of the network and the activated sensor nodes, which is Markov in nature. The state space is not exhaustive, and the sequential activation of sleeping sensors is in line with the features of the discrete problem. Therefore, the Q-learning algorithm is used to model and solve the target coverage problem.

5.1. Modeling

Q-learning algorithms require the definition of the state space, the action space of the smart body and the reward function given by the environment.

5.1.1. States

The state of WSN is described by the state of the sensor nodes and the state of the target coverage, and the state of WSN at the moment t can be denoted by Formula (11).

$$S_t = \{state_1, state_2, \ldots, state_n, \{tc_1, tc_2, \ldots, tc_m\}\} \tag{11}$$

where $\{tc_1, tc_2, \ldots, tc_m\}$ records the number of times each target has been covered by the activated sensor nodes in the current network, and $state_i$ = {e_i, $isActive_i$} records the state of the sensor node p_i, e_i is the residual energy of p_i, and $isActive_i$ denotes whether p_i is activated or not, and takes the following values.

$$isActive_i = \begin{cases} 0, & p_i \text{ } is\ sleeping. \\ 1, & p_i \text{ } is\ actived. \end{cases} \tag{12}$$

In the initial state, each sensor node in the network is sleeping. The state S_t of WSN is input into Q-network to estimate the value of action corresponding to activating each sensor node, and based on this, the activated sensor node p_i is selected. The transfer of state can be categorized into the following four types according to the situation of p_i at this time.

(1) p_i is in the sleeping state, at this time, p_i is transferred from the sleeping state to the active state, and the activated sensor nodes in the network do not constitute a feasible solution, i.e., $\exists\ tc_j = 0, j \in \{1, 2, \ldots, m\}$. $state_i$ and the target coverage are updated in S_{t+1}, i.e., $state_i = \{e_i - 1, 1\}$, and the target coverage update formula is as follows.

$$tc_j = tc_j + 1, \qquad \forall\ t_j \in c_i \tag{13}$$

(2) p_i is active, when the environment state $S_{t+1} = S_t$.

(3) p_i is dead, at which point the environmental state $S_{t+1} = S_t$.

(4) p_i is sleeping, at this time p_i from the sleeping to the activated, the sensor nodes that are active in the network can constitute a feasible solution; S_{t+1} is first updated according to type (1), and then all the sensor states as well as the target coverage are updated.

The sensor state update formula is as follows.

$$state_i = \{e_i, 0\ \}, \qquad \forall\ i \in \{1,2,\ldots,n\} \tag{14}$$

The target coverage update formula is shown in Formula (15).

$$tc_j = 0, \qquad \forall\ j \in \{1,2,\ldots,m\} \tag{15}$$

5.1.2. Action

The action space of the smart body is denoted as $A = \{p_1, p_2, \ldots, p_n\}$. According to the current state of WSN, the action of the smart body is to select a sensor node and try to activate it. In this paper, the current action a_t is selected according to the ε-greedy policy, which is expressed by Formula (16).

$$a_t = \begin{cases} \underset{\pi}{\operatorname{argmax}}\, Q(s_t, a_t; \omega), & with\ probability\ (1-\varepsilon) \\ Uniformly\ extracting\ an\ action\ in\ A, & with\ probability\ \varepsilon \end{cases} \tag{16}$$

5.1.3. Regard

In this paper, we design the reward function in terms of the number of targets covered by sensor nodes and their own energy consuming. From Theorem 1, the smart body should construct the non-redundant feasible solution. Therefore, the environment should reward the smart body for actions that help to construct non-redundant feasible solutions and penalize the actions that lead to redundancy. For each action, the greater the number of targets covered individually by a sensor node, the greater the corresponding rewards. The more residual energy in the case of the same number of targets covered individually, the more the corresponding rewards. Redundant sensor nodes should be penalized and the less energy they have, the greater the penalty should be. Activated and dead sensor nodes do not contribute to the composition of the feasible solution and do not consume energy, so the reward is set to 0. As the training progresses, the smart body will gradually learn the strategy to construct more non-redundant feasible solutions and extend the network lifetime. Therefore, the reward function of the smart body after choosing to activate the sensor node s_i at moment t can be expressed by Formula (17).

$$r_t = \begin{cases} (1 - isActive_i)\left(\sum_{t_j \in c_i} \left\lfloor \frac{1}{tc_j+1} \right\rfloor - \frac{E-e_i}{E}\right), & \forall\, e_i > 0 \\ 0, & \forall\, e_i = 0 \end{cases} \tag{17}$$

5.2. Network Structure and Algorithm

The network of the Q-Learning algorithm for the target coverage is shown in Figure 4. The state s_t inputs to the Q-Network, and the smart body selects the action corresponding to the maximum action value with probability $1 - \varepsilon$, and randomly explores the environment with probability ε to expand the search space. The algorithm collects the smart body tracks (s_t, a_t, r_t, s_{t+1}) and adds them to the memory pool, and randomly selects appropriate batches of experience for training to break the correlation of experience and increase the training speed. Meanwhile, this paper adopts the Target-Network to alleviate the bootstrap and the target value overestimation in the DQN algorithm. The Target-Network and the Q-Network have the same structure, but the network parameters are different, which are denoted as $\widetilde{\omega}$ and ω, respectively. After each round of training, the Q-Network updates the network parameters ω. After a predesigned number of training rounds, the network parameters of the Target-Network are updated to the current network parameters of the Q-Network, i.e., $\omega = \widetilde{\omega}$. The algorithm is implemented by the Target-Network's $\max\limits_{a\in A} Q\left(s_{t+1}, a_{t;}\widetilde{\omega}\right)$ to estimate the action value of s_{t+1}. Combined with obtaining the real feedback r_t given by the environment, the prediction of the action value y_t of s_t can be expressed as Formula (18).

$$y_t = r_t + \gamma \max_{a\in A} \mathrm{Q}\left(s_{t+1}, a_t; \widetilde{\omega}\right) \tag{18}$$

Since y_t is partially based on the facts, $Q(s_t, a_t; \omega)$ should be close to y_t. In this paper, the deep Q-network is trained using the mean-variance loss function $L(\omega)$, which is calculated as Formula (19).

$$\mathrm{L}(\omega) = \frac{1}{2}[Q(s_t, a_t;\ \omega) - y_t]^2 \tag{19}$$

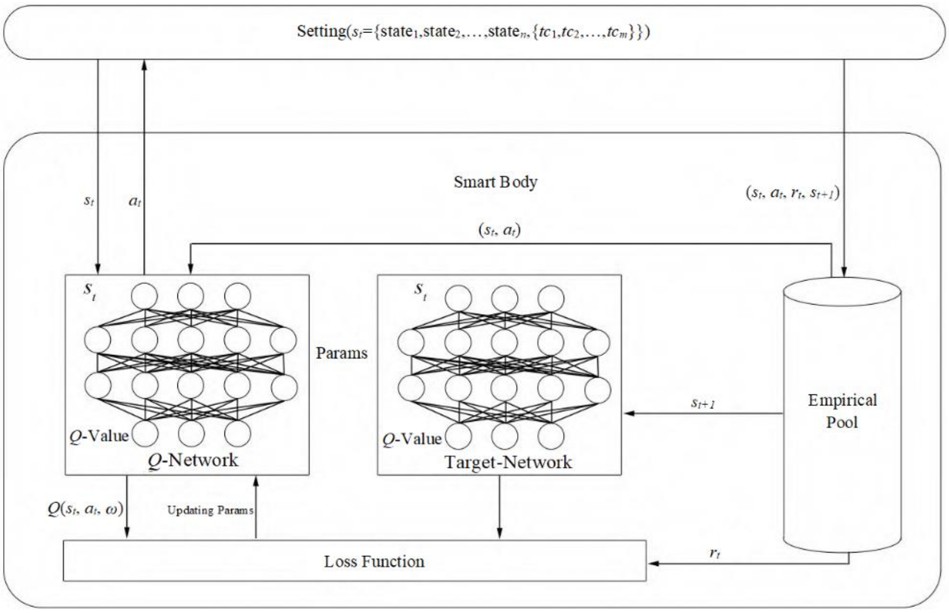

Figure 4. The network for the Q-Learning algorithm.

The exact flow of the algorithm is shown in Algorithm 1.

Algorithm 1. Q-Learning algorithm for the Target Coverage

```
Require: The number of training rounds $e$, synchronization parameter interval value $C$,
    the memory pool capacity $M$, the number of data extraction batches $b$.
Ensure: The cumulative rewards and network lifetime.
 1: Initialize the Q-network parameters to $\omega$, the target network parameters to $\widetilde{\omega} = \omega$,
    and initialize the memory pool.
 2: repeat
 3:     Initialize the cumulative rewards and network lifetime to 0.
 4:     The smart body senses the environment to get the state $s_t$.
 5:     $e = e - 1$
 6:     repeat
 7:         The network selects the current action $a_t$ according to the $\varepsilon$-greedy policy.
 8:         Obtain the instantaneous reward $r_t$ and update $s_{t+1}$ after executing the action $a_t$.
 9:         $(s_t, a_t, r_t, s_{t+1}) \rightarrow$ the memory pool.
10:         Update the cumulative rewards.
11:         Predict the value of action according to Formula (18).
12:         Randomly select the $b$-th batche of data from the empirical pool for training and
            updating the Q-network parameters $\widetilde{\omega} = \omega$.
13:         Synchronize $\widetilde{\omega} = \omega$ every $C$ steps.
14:     until a feasible solution cannot continue to form.
15:     Output the cumulative rewards and network lifetime obtained this time.
16: until $e = 0$
```

6. Experiments and Analysis of Results

In total, 50, 100, 150 sensor nodes are randomly deployed in a rectangular area of 80 m × 80 m corresponding to 10, 20, 30 targets. The sensing radius of them is 10, 20, 30, 40 m, and the initial energy is three units. The Q-network has four layers, in which the number of neurons in the input layer is related to the state dimension, the number of neurons in the middle two fully connected layers are 256 and 128, respectively, and the number of neurons in the output layer is the same as the number of sensor nodes in the network. The experiments use the ReLU activation function. More than 6000 data can be stored in the empirical pool, and 64 batches of data are randomly selected each time to participate in the training. When the data in the empirical pool are full, the newly generated data will replace the old data. The discount factor in the learning process is set to $\gamma = 0.9$.

6.1. Convergence Proof

As shown in Figure 5, 10 targets are randomly deployed in a WSN consisting of 50 sensors with a sensing radius of 40 m. The experiment recorded the cumulative gains of the smart body in 3000 rounds of training. In Figure 5, the smart body selects actions blindly at the beginning of training, and selects more redundant sensors, activated sensors and dead sensors. As the number of training rounds increases, the smart body gradually learns the strategy of obtaining more single-step rewards, and the cumulative rewards gradually increase. After 2500 rounds of training, the cumulative gains obtained by the smart body in each round of training are gradually smooth, and the algorithm tends to converge.

6.2. Comparison with Similar Algorithms

The network is randomly deployed with 10, 20 and 30 targets and 50, 100 and 150 sensor nodes with a sensing radius of 40 m. Using the Deep Q-learning algorithm proposed in this paper, three greedy policy algorithms (Greedy 1 adopts the policy of selecting the sensor node with the highest number of covered targets at a time, Greedy 2 adopts the policy of selecting the sensor node with the highest residual energy and Greedy 3 adopts the policy of selecting the sensor node that covers the most number of uncovered targets), the Maximum Life-cycle Target Coverage Algorithm (MLTC) [22] and the Adaptive Learning Automata Algorithm (ALAA) [14] were used to calculate the network lifecycle, respectively, and the results are shown in Figures 6–8. It can be seen that for Greedy 1 and Greedy 2 algorithms, in the case of a small number of sensor nodes, the difference between the network lifetime and the other algorithms is very small, when the number of sensors increases, the coverage of the situation is complex, the two algorithms are not good. In Greedy 3, the MLTC and ALAA algorithms have better results, but the stability of the different network environments needs to be improved. Q-learning has better results than several other algorithms and achieves a longer lifetime. Meanwhile, the results show that the lifetime is positively correlated with the number of sensor nodes.

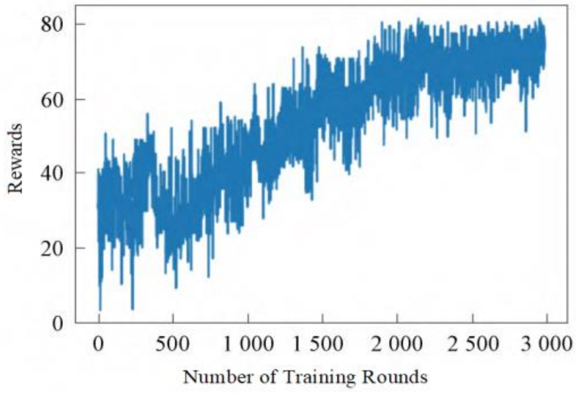

Figure 5. The reward convergence curve.

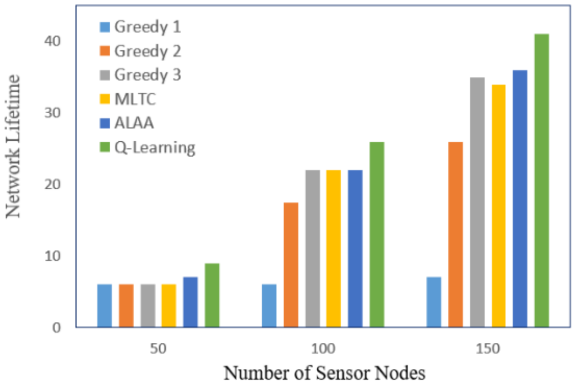

Figure 6. Comparison of network lifetime with a target number of 10.

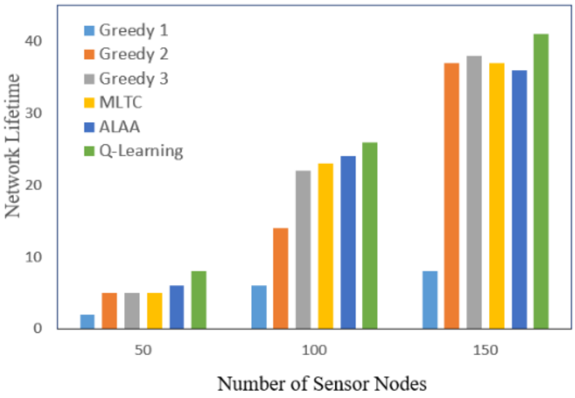

Figure 7. Comparison of network lifetime with a target number of 20.

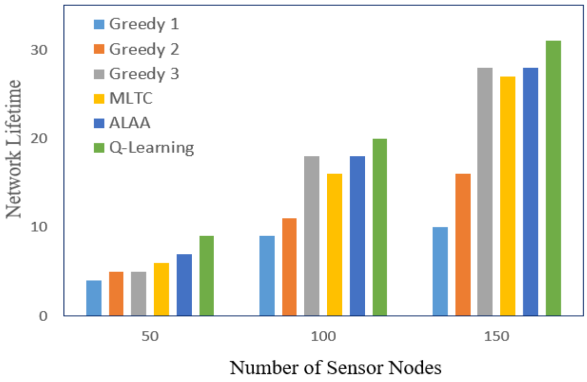

Figure 8. Comparison of network lifetime with a target number of 30.

6.3. *Impact of Sensing Radius on the Lifetime*

In total, 100 sensor nodes and 40 targets are randomly deployed in the network, and the sensing radius of the sensors are 10, 20, 30 and 40 m. The lifetime is calculated using each of the above algorithms, and the results are shown in Figure 9. It can be seen that when the sensing radius is 10 m, the number of targets covered by each sensor in the network are small, and the lifetime calculated by each algorithm is approximate. With the increase in the sensing radius, the number of targets covered by each sensor increases, and each target can be covered by more sensors. Therefore, although there are differences in the lifetime calculated by the algorithms, they all show that the network lifetime is positively correlated with the sensing radius of sensors.

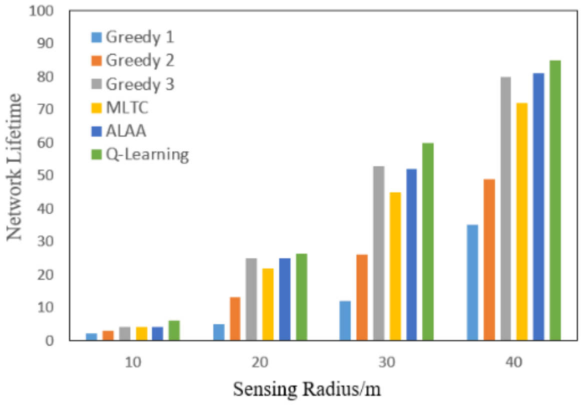

Figure 9. Comparison of the lifetime for different sensory radii.

6.4. *Impact of Initial Energy of Sensor on the Lifetime*

The network is deployed with 100 sensor nodes, and 20 targets at the same time. The sensor sensing radius is fixed at 30 m, and the initial energy carried by the sensors is increased step by step from 3 units to 12 units, and the results of the experiments are shown in Figure 10. It can be seen that when the sensor carries less energy, the lifetime of each algorithm is shorter. With the increase in carrying energy, the lifetime of each algorithm is extended. Therefore, the experiments show that the lifetime is positively correlated with the initial energy of sensor.

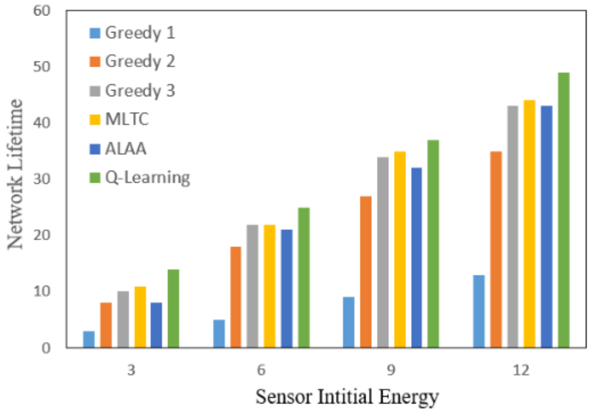

Figure 10. Comparison of the lifetime for different initial energies.

7. Conclusions

To address the problems of an unclear mechanism of node activation strategy and a redundancy of feasible solution in the process of solving the target coverage of wireless sensor networks, this paper proposes a target coverage algorithm based on Q-learning to model and solve the target coverage of wireless sensor networks. First, the energy, activation and target coverage of sensors are used to describe the network state. Second, the actions of the smart body are taken as the sensors to be activated in the network. And finally, the rewards of the smart body are used to consider the number of targets to be covered by the sensors in combination with its own energy consumption. Through training, the smart body learns the strategy of scheduling sensors in the network, which can reduce the redundancy and increase the number of feasible solutions. The simulation results show that the proposed algorithm in this paper can reasonably schedule the sensor nodes and extend the network lifetime.

It is important to note here that from previous experimental results, the Greedy 3 algorithm also maintains the performance as the network size becomes larger, the node sensing radius increases or the initial energy carried increases. For the proposed algorithm, reinforcement learning is better at dealing with highly complex computational problems. From this point of view, the proposed algorithm should have more room for improvement compared with the Greedy 3 algorithm in solving the target coverage problem. However, as far as the current simulation results are concerned, the proposed algorithm has the problem of slow convergence speed, and this is a problem that needs to be solved in future work.

Author Contributions: Conceptualization, P.X.; Software, P.X., D.H. and T.L.; Formal analysis, P.X.; Resources, P.X., D.H. and T.L.; Data curation, D.H. and T.L.; Writing—original draft, P.X.; Writing—review & editing, P.X.; Supervision, P.X. and D.H.; Project administration, P.X.; Funding acquisition, P.X. and D.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China; grant number 61862045.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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

References

1. Lee, H.; Keshavarzian, A. Towards energy-optimal and reliable data collection via collision-free scheduling in wireless sensor networks. In Proceedings of the INFOCOM, Phoenix, AZ, USA, 13–18 April 2008.
2. Zhu, C.; Zheng, C.L.; Shu, L.; Han, G. A Survey on Coverage and Connectivity Issues in Wireless Sensor Networks. *J. Netw. Comput. Appl.* **2012**, *35*, 619–632. [CrossRef]

3. Carrabs, F.; Cerulli, R.; Raiconi, A. A Hybrid Exact Approach for Maximizing Lifetime in Sensor Networks with Complete and Partial Coverage Constraints. *J. Netw. Comput. Appl.* **2015**, *58*, 12–22. [CrossRef]
4. Castaio, F.; Rossi, A.; Sevaux, E.M.; Velasco, N. A Column Generation Approach to Extend Lifetime in Wireless Sensor Networks with Coverage and Connectivity Constraints. *Comput. Oper. Res.* **2014**, *52*, 220–230. [CrossRef]
5. Rossi, A.; Singh, A.; Sevaux, M. An Exact Approach for Maximizing the Lifetime of Sensor Networks with Adjustable Sensing Ranges. *Comput. Oper. Res.* **2012**, *39*, 3166–3176. [CrossRef]
6. Rossi, A.; Singh, A.; Sevaux, M. Focus Distance-Aware Lifetime Maximization of Video Camera-Based Wireless Sensor Networks. *J. Heuristics* **2021**, *27*, 5–30. [CrossRef]
7. Shahrokhzadeh, B.; Dehghan, E.M. A Distributed Game-Theoretic Approach for Target Coverage in Visual Sensor Networks. *IEEE Sens. J.* **2017**, *17*, 7542–7552. [CrossRef]
8. Yen, L.H.; Lin, C.M.; Leung, V.C.M. Distributed Lifetime-Maximized Target Coverage Game. *ACM Trans. Sens. Netw. (TOSN)* **2013**, *9*, 1–23. [CrossRef]
9. Cardei, I.M.; Wu, J. Energy-Efficient Coverage Problems in Wireless Ad-Hoc Sensor Networks. *Comput. Commun.* **2006**, *29*, 413–420. [CrossRef]
10. Rangel, E.O.; Costa, D.G.; Loula, A. On Redundant Coverage Maximization in Wireless Visual Sensor Networks: Evolutionary Algorithms for Multi-objective Optimization. *Appl. Soft Comput.* **2019**, *82*, 105578. [CrossRef]
11. Akram, J.; Munawar, H.S.; Kouzani, A.Z.; Mahmud, M.A.P. Using Adaptive Sensors for Optimised Target Coverage in Wireless Sensor Networks. *Sensors* **2022**, *22*, 1083. [CrossRef] [PubMed]
12. Liang, D.Y.; Shen, H.; Chen, L. Maximum Target Coverage Problem in Mobile Wireless Sensor Networks. *Sensors* **2020**, *21*, 184. [CrossRef] [PubMed]
13. Duan, J.; Yao, A.N.; Wang, Z.; Yu, L.T. Improved Sparrow Search Algorithm Optimises Wireless Sensor Network Coverage. *J. Jilin Univ. (Eng. Technol. Ed.)* **2024**, *54*, 761–770.
14. Chand, S.; Kumar, B. Target Coverage Heuristic Based on Learning Automata in Wireless Sensor Networks. *IET Wirel. Sens. Syst.* **2018**, *8*, 109–115.
15. Allah, M.N.; Motameni, H.; Mohamani, H. A Genetic Algorithm-Based Approach for Solving the Target Q-Coverage Problem in Over and Under Provisioned Directional Sensor Networks. *Phys. Commun.* **2022**, *54*, 101719.
16. Littman, M.L. Markov games as a framework for multi-agent reinforcement learning. In Proceedings of the Eleventh International Conference on International Conference on Machine Learning, New Brunswick, NJ, USA, 10–13 July 1994; pp. 157–163.
17. Cao, X.B.; Xu, W.Z.; Liu, X.X.; Peng, J.; Liu, T. A Deep Reinforcement Learning-Based on-Demand Charging Algorithm for Wireless Rechargeable Sensor Networks. *Ad Hoc Netw.* **2021**, *110*, 102278. [CrossRef]
18. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with Deep Reinforcement Learning. *arXiv* **2013**, arXiv:1312.5602.
19. Banoth SP, R.; Donta, P.K.; Amgoth, T. Dynamic Mobile Charger Scheduling with Partial Charging Strategy for WSNs Using Deep Q-Networks. *Neural Comput. Appl.* **2021**, *33*, 15267–15279. [CrossRef]
20. Lu, H.; Zhang, X.W.; Yang, S. A Learning-Based Iterative Method for Solving Vehicle Routing Problems. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 30 April 2020; pp. 1–15.
21. Younus, M.U.; Khan, M.K.; Anjum, M.R.; Afridi, S.; Arain, Z.A.; Jamali, A.A. Optimizing the Lifetime of Software Defined Wireless Sensor Network via Reinforcement Learning. *IEEE Access* **2020**, *9*, 259–272. [CrossRef]
22. Saadi, N.; Bounceur, A.; Euler, R.; Lounis, M.; Bezoui, M.; Kerkar, M. Maximum Lifetime Target Coverage in Wireless Sensor Networks. *Wirel. Pers. Commun.* **2020**, *111*, 1525–1543. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.