Hindawi
International Journal of Aerospace Engineering
Volume 2021, Article ID 8647386, 12 pages
https://doi.org/10.1155/2021/8647386

Research Article

Cognitive Electronic Jamming Decision-Making Method Based on Improved Q-Learning Algorithm

Huiqin Li, Yanling Li, Chuan He, Jianwei Zhan, and Hui Zhang

Xi'an Research Institute of High Technology, Xi'an 710025, China

Correspondence should be addressed to Yanling Li; lyling998@163.com

Received 13 September 2021; Revised 13 October 2021; Accepted 9 November 2021; Published 22 December 2021

Academic Editor: Erkan Kayacan

In this paper, a cognitive electronic jamming decision-making method based on improved Q-learning is proposed to improve the efficiency of radar jamming decision-making. First, the method adopts the simulated annealing (SA) algorithm's Metropolis criterion to enhance the exploration strategy, balancing the contradictory relationship between exploration and utilization in the algorithm to avoid falling into local optima. At the same time, the idea of stochastic gradient descent with warm restarts (SGDR) is introduced to improve the learning rate of the algorithm, which reduces the oscillation and improves convergence speed at the later stage of the algorithm iteration. Then, a cognitive electronic jamming decision-making model is constructed, and the improved Q-learning algorithm's specific steps are given. The simulation experiment takes a multifunctional radar as an example to analyze the influence of exploration strategy and learning rate on decision-making performance. The results reveal that compared with the traditional Q-learning algorithm, the improved Q-learning algorithm proposed in this paper can fully explore and efficiently utilize and converge the results to a better solution at a faster speed. The number of iterations can be reduced to more than 50%, which proves the feasibility and effectiveness of the method applied to cognitive electronic jamming decision-making.

1. Introduction

Cognitive electronic jamming decision-making is a critical link in cognitive electronic warfare [1]. Its task is mainly divided into three steps. First, the jammer completes the recognition of the target working state based on the reconnaissance target signal. Then, the effect of the current jamming action is evaluated, and the best correspondence between the various states of the counter target and the existing jamming action is established. Finally, the optimal jamming strategy is generated intelligently for the different states of the target, which is used to guide the subsequent jamming resource scheduling. In the increasingly complex and changeable electromagnetic environment, jamming methods and antijamming technologies emerge one after another. At present, it is difficult to establish a one-to-one correspondence between the specific radar working state and some jamming action. Therefore, selecting an accurate jamming method can make the radar system give full play to its power. At the same time, with the rapid development of new weapons and equipment and the emergence of many new systems and multifunctional radars, the existing jamming decision-making methods cannot effectively deal with the battlefield environment. Therefore, the research on jamming decision-making methods is urgent.

At present, the traditional research methods of jamming decision-making mainly include the following: methods based on game theory, methods based on decision support systems, and methods based on swarm intelligence optimization algorithms. David et al. [2] proposed a framework that utilizes game-theoretic principles to provide an autonomous determination of the appropriate electronic attack action to be taken for a given scenario. Gao et al. [3] established a profit matrix based on the principles of minimizing loss and maximizing jamming benefits and used the Nash equilibrium strategy to solve the optimal jamming strategy. However, this method relies on the profit matrix's establishment and is only suitable for radar systems with constant parameter characteristics. Sun et al. [4] proposed a method of electronic jamming mode selection based on D-S theory.

Li and Wu [5] proposed a design method for an intelligent decision support system (IDSS) based on a knowledge base and a problem-solving unit. The method's applicability is more extensive, but it is too dependent on the posterior probability and lacks real-time performance. Ye et al. [6, 7] proposed a cognitive collaborative jamming decision-making method based on bee colony algorithm, which finds the globally optimal solution through the process of bee colony searching for high-quality resources. Similarly, there are swarm intelligence algorithms such as genetic algorithm [8, 9], ant colony algorithm [10], and heuristic algorithms such as differential evolution algorithm [11–13] and water wave optimization algorithm [14]. However, these algorithms cannot consider all possible jamming factors when solving the jamming decision model, and the decision accuracy needs to be improved. In summary, how to improve the autonomy, timeliness, and accuracy of jamming decision-making in cognitive electronic warfare remains to be studied.

The above traditional jamming decision-making methods rely on sufficient prior knowledge for "matching," mainly suitable for radars with constant characteristic parameters. They do not have real-time performance and cannot deal with the increasingly complex battlefield environment. Reinforcement learning [15, 16] is a machine learning method specifically used to solve behavioral decision-making problems. The jammer establishes the connection between the jamming resource and the target state through reinforcement learning, continuously optimizes the jamming strategy, and realizes the cognitive electronic jamming decision-making. The Q-learning algorithm is a typical time-series differential reinforcement learning algorithm based on model-free learning. It allows the system to learn independently and make correct decisions in real time without considering environmental model factors and sufficient prior information. Thus, it is fully applicable to the complex and changeable radar system. Therefore, compared with traditional jamming decision-making methods, the jamming decision-making method based on the Q-learning algorithm can realize learning while fighting, which will be the future development trend and main research direction. Currently, the Q-learning algorithm is widely used in robot path planning [17, 18], nonlinear control [19, 20], resource allocation scheduling [21, 22], and other fields, and it has also achieved specific results in electronic jamming decision-making. Aiming at the unknown radar working mode, Xing et al. [23, 24] proposed that the Q-learning algorithm was applied to radar countermeasures to realize intelligent jamming decision. Li et al. [25] proposed using Q-learning to train the behavior of radar systems, which can effectively complete the jamming and adapt to a different combat mission. However, the Q-learning algorithm still has two problems in practical application: (1) exploration strategy cannot be selected. In traditional Q-learning, the exploration strategy is always a single constant value. When the exploration value is large, the algorithm is fully explored in the early stage, but the result is easy to oscillate near the optimal solution in the later stage of the algorithm, and it is difficult to converge. When the exploration value is small, the exploration is scarce in the early stage, and it is easy to converge to the local optimal in the later stage. Therefore, the fixed exploration strategy cannot balance the sufficiency of exploration and the stability of convergence. (2) There is no uniform standard for the selection of learning rate. The learning rate of traditional Q-learning is usually fixed according to experience. When the learning rate is large, the risk of learning is easy to occur in the early stage of the algorithm. When the learning rate is small, the convergence speed of the algorithm becomes slow in the later stage. Therefore, to improve the accuracy and efficiency of the Q-learning algorithm applied to radar jamming decision-making, it is still necessary to improve the Q-learning algorithm.

Considering the problems of the traditional Q-learning algorithm, this paper proposes a cognitive electronic jamming decision-making method based on improved Q-learning. The improved techniques include the following:

(1) The Metropolis criterion of the SA algorithm is introduced to improve the action choice strategy to balance the exploration and utilization of the algorithm

(2) The learning rate decay strategy of SGDR is used to speed up the learning convergence speed and avoid the algorithm shocking to fall into the local optimum

(3) The Q value convergence rule is used as the priority termination condition of the algorithm, and the iteration number rule is suboptimum so that the algorithm is forced to stop the learning process and output the suboptimal jamming strategy when the Q value cannot converge, which can save jamming resources

This paper takes the multifunctional radar provided in [26] as the research object, constructs a cognitive electronic jamming decision model based on improved Q-learning, and compares the simulation results with traditional methods. The results show that the improved method can independently learn the optimal jamming strategy by analyzing the jamming effect with the radar working state change, improve the learning efficiency, and give full play to the adaptability and timeliness of the cognitive electronic countermeasure system.

The rest of this paper is arranged as follows. A detailed introduction to the Q-learning algorithm and a description of the improvement methods are presented in Section 2. The cognitive electronic jamming decision-making model and improved Q-learning algorithm are put forward in Section 3. The simulation experiment and result analysis are given in Section 4. Finally, some conclusions drawn from this study are discussed in Section 5.

2. Improved Q-Learning Algorithm

2.1. Q-Learning Algorithm. The principle of the Q-learning algorithm is shown in Figure 1. Its main idea is based on Markov decision processes. By perceiving the current environment state, the agent determines the action taken by a specific strategy and obtains the immediate reward and the

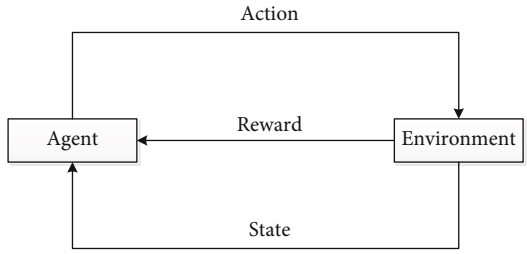

Figure 1: Principle of the Q-learning algorithm.

next state of the environment feedback. Essentially, it learns a mapping from the environment state to the action measure to maximize the overall reward value.

The algorithm can be summarized as the following steps:

Step 1. Define the state set $S = \{s_1, s_2, \cdots, s_n\}$ and the action set $A = \{a_1, a_2, \cdots, a_n\}$, initialize the "state-action" function $Q(s, a)$ and the reward matrix R, and set the parameters such as the maximum number of iterations K.

Step 2. Randomly select an initial state from S. When the state is the target state, the iteration ends, and the initial state is reselected. Otherwise, continue to execute step 3.

Step 3. According to the ε-greedy strategy, select an action among all possible actions in the current state and reach the next state.

Step 4. Equation (1) is used to update the matrix Q.

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha * \left[R(s_t, a_t) + \gamma \max_{a' \in A} Q\left(s_{t+1}, a'\right) - Q(s_t, a_t)\right], \tag{1}$$

where s_t is the state of the environment at the moment t, a_t is the action taken by the agent at the moment t, $Q(s_t, a_t)$ is the "state-action" function at the moment t, s_{t+1} is the state of the environment at the moment $t+1$, $R(s_t, a_t)$ is the immediate reward of the environment's feedback from t to $t+1$, a' is the action that maximizes the value Q when agent arrives s_{t+1}, γ is the discount factor, $\gamma \in [0, 1]$, α is the learning rate, and $\alpha \in (0, 1)$.

Step 5. Set the next moment's state to the current state, that is, $s_t = s_{t+1}$. If s_t is not target state, return to step 3.

Step 6. When the maximum number of iterations is reached, the training is completed, the convergent matrix Q is obtained, and the optimal action strategy is output according to equation (2). Otherwise, it returns to step 2 to enter the next iteration.

$$\pi^*(s) = \arg \max_{a \in A} (Q * (s, a)) \tag{2}$$

2.2. Improvement of Exploration Strategy Based on the SA Algorithm. When selecting actions, the traditional Q-learning algorithm follows the ε-greedy strategy, which randomly explores an action with the probability of ε, and utilizes existing information to select the optimal action with the probability of $1 - \varepsilon$. The larger the ε, the stronger the exploration ability. However, as the agent continuously interacts with the environment, if the agent still uses a larger ε to explore after acquiring empirical knowledge, the algorithm results will likely oscillate near the optimal solution. The smaller the ε, the stronger the utilization ability, but due to the lack of exploration in the early stage, it is easy to converge to the local optimum later. Therefore, the fixed exploration strategy cannot balance exploration and utilization, making the algorithm difficult to converge and easy to falls into a local minimum. Aiming at the above problems, heuristic algorithms such as particle swarm optimization, ant colony optimization, and artificial bee colony algorithm proposed early can be used to solve the contradiction between exploration and utilization. The SA algorithm is an optimization algorithm based on neighborhood search and learns from the idea of the annealing process. It keeps the probability of continuous decay in the search process to jump out of the local optimal value and converge to the global optimal value. Compared with other heuristic algorithms, the SA algorithm can accept the worse solution than the current solution with a certain probability criterion and the better solution with all probabilities. It has the characteristics of simple principle, high iterative search efficiency, strong robustness, asymptotic convergence, and strong global searchability. Therefore, it has been widely used in various fields to solve combinatorial optimization problems. The algorithm is relatively mature. In this paper, the Metropolis criterion [27] in the SA algorithm is introduced to improve the exploration strategy in the Q-learning algorithm. The exploration probability is adjusted by the cooling strategy, so that ε can stably maintain a large value in the early iteration to fully explore and quickly keep a small value in the later iteration to speed up the algorithm convergence while ensuring that the globally optimal solution is obtained. The probability equation for randomly taking actions using the SA algorithm is expressed as follows:

$$\varepsilon = \begin{cases} 1, Q(s, a_r) \le Q\left(s, a_p\right), \\ \exp\left(-\dfrac{Q(s, a_r) - Q\left(s, a_p\right)}{T_k}\right), \text{other}, \end{cases} \tag{3}$$

where a_r is the randomly selected action, a_p is the selected action according to the current ε-greedy strategy, T is the temperature control parameter, and k is the number of iterations. When $Q(s, a_r) \le Q(s, a_p)$, randomly explore action a_r. Otherwise, randomly explore an action with probability of $\exp(-Q(s, a_r) - Q(s, a_p)/T_k)$.

The temperature cooling strategy in the SA algorithm determines ε's change. In the early iteration, T is large, and the probability of accepting randomly selected actions is great, which is conducive to exploration. In the later iteration, the temperature drops, and T becomes smaller; so, the probability of taking the optimal action is larger, which is conducive to utilization. Common annealing strategies

include a geometric cooling strategy, a logarithmic descent strategy, and a linear descent strategy. This paper uses the most common geometric cooling strategy to keep a higher temperature at the beginning of the iteration and quickly cool down at the end of the iteration. The specific description is expressed as follows:

$$T_k = T_0 * \lambda^k, \tag{4}$$

where T_0 is the initial temperature, k is the number of iterations, λ is the cooling parameter, $\lambda \in (0, 1)$, and λ is generally taken as a constant close to 1.

2.3. Improvement of Learning Rate Based on SGDR. The learning rate α determines the learning ability of the agent. The larger α, the stronger the learning ability. Equation (1) can be rearranged as follows:

$$Q(s_t, a_t) = (1 - \alpha) * Q(s_t, a_t) + \alpha * \left[R(s_t, a_t) + \gamma \max_{a' \epsilon_A} Q\left(s_{t+1}, a'\right)\right]. \tag{5}$$

In the traditional Q-learning algorithm, α is a fixed value. As shown in equation (5), the larger the α, the less the previous training's effect, and the shorter the system decision time, but there is a risk of overlearning, and it is easy to fall into the local optimum. The smaller the α, the smaller the oscillation, but the convergence speed will be slower. The traditional learning rate adjustment methods include the equal interval adjustment method, exponential attenuation method, and adaptive adjustment method. The above methods always make α maintain the attenuation trend in the adjustment process, which can slow down the convergence speed later in algorithm iteration. In 2017, Loshchilov and Hutter [28] proposed the SGDR to improve α. The warm restart mechanism is set in the α decrementing process, and then α is reinitialized to a certain preset value to gradually decay after every interval period. Compared with the traditional learning rate adjustment method, the SGDR method makes the algorithm keep a larger value in the early iteration to speed up the convergence speed and keeps a smaller value in the later iteration to prevent falling into the local optimum. At the same time, α's reciprocating rise and fall can prevent small α from affecting the convergence speed. Therefore, this paper uses the SGDR method to improve the learning rate in the Q-learning algorithm, which considers the convergence speed and stability of the algorithm at the same time.

Assuming that the total number of restarts is M, the cosine decay is used to reduce the learning rate before the mth restarting. In the kth iteration, the improved learning rate calculation equations are expressed as follows:

$$\alpha_k = \alpha_{\min}^m + \frac{1}{2}\left(\alpha_{\max}^m - \alpha_{\min}^m\right)\left(1 + \cos\left(\frac{\beta}{\tau_m}\pi\right)\right), \tag{6}$$

$$\alpha_{\min}^m = f(m), \tag{7}$$

$$\alpha_{\max}^m = g(m), \tag{8}$$

$$\tau_m = \tau_0 * \kappa^{(m-1)}, \tag{9}$$

where $\alpha_{\max}^m$ and $\alpha_{\min}^m$ are, respectively, the maximum and minimum values of the learning rate in the mth restart period, which can be gradually reduced with the increase of m. τ_m is the restart period, which can gradually increase with the number of restarts. τ_0 is the initial restart period, κ is the amplification factor, β is the number of iterations since the last restart, and $0 \le \beta \le \tau_m$. When $\beta = \tau_m$, set $\alpha_t = \alpha_{\min}^m$. After restarting, set $\beta = 0$ and $\alpha_k = \alpha_{\max}^{m+1}$.

3. Cognitive Electronic Jamming Decision-Making Based on Improved Q-Learning

3.1. Cognitive Electronic Jamming Decision-Making Model. In this paper, the improved Q-learning algorithm is applied to radar jamming decision-making to realize adaptive cognitive electronic jamming decision-making. The jammer first judges the change of threat level by detecting the working state of the enemy's radar before and after jamming to quantitatively evaluate the jamming effect. The jamming decision system learns the best jamming strategy through the jamming effect. It adjusts Q-learning's exploration strategy and learning rate through improved algorithms to achieve fast and accurate cognitive jamming. Through the Q-learning process, the connection between the jamming resources and the radar working state can be established, the jamming strategy is continuously optimized, and the learning conclusions can be used to support the construction of the jamming rule base and the dynamic threat base. The cognitive electronic jamming decision model based on improved Q-learning is shown in Figure 2.

The Q-learning algorithm principle shown in Figure 1 has the following mapping relationship with the cognitive electronic jamming decision-making model shown in Figure 2:

(1) Agent—jammer

(2) Environment—enemy radar

(3) Status—radar working state

(4) Action—jamming strategy

(5) Reward value—jamming effect evaluation

3.1.1. Definition of State Set and Action Set. With the development of radar working systems, modern radar working states can be divided into searching, tracking, ranging, imaging, monitor, guidance, and other states according to combat tasks. Different working states can be flexibly switched, and their corresponding threat levels will also change. Among them, the threat level of the search state is the lowest, which is the target state of the jammer. Each working state has corresponding jamming actions [26], and the corresponding relationship is shown in Figure 3. Effective jamming can gradually reduce the threat level of radar.

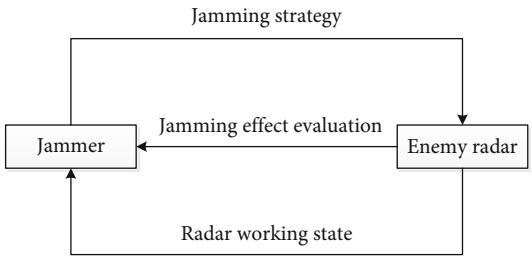

Figure 2: Cognitive electronic jamming decision-making model.

3.1.2. Definition of Reward Function. The reward function determines the decision-making ability of the jammer. Since the ultimate goal of the jammer is to improve the performance of the radar jamming, the radar jamming effect evaluation can be used as the reward function. The evaluation of the radar jamming effect is closely related to the change of threat level. This paper divides the threat level change into four situations:

(1) If the threat level of the radar working state is reduced to the lowest, the jamming effect is the best, and the reward value is +100

(2) If the threat level decreases but not to the lowest level, the jamming effect is good, and the reward value is +1

(3) If the threat level remains unchanged or increases, the jamming effect is poor, and the reward value is -1

(4) If there is no transition between working states, the reward value is 0

The specific calculation equation of the reward function is expressed as follows:

$$R = \begin{cases} 100, \mathrm{TL}' \longrightarrow \min, \\ 1, \mathrm{TL}' < \mathrm{TL}\left(\mathrm{TL}' \longrightarrow \min\right), \\ -1, \mathrm{TL}' \geq \mathrm{TL}, \\ 0, \mathrm{TL}' \leftrightarrow \mathrm{TL}, \end{cases} \tag{10}$$

where min is the lowest threat level, TL is the threat level before jamming, and TL' is the threat level after jamming.

3.2. Cognitive Electronic Jamming Decision-Making Algorithm Based on Improved Q-Learning. To prevent the Q-learning algorithm from falling into a local minimum and improve the convergence speed and decision accuracy of the algorithm, this paper uses the SA algorithm and the SGDR method to, respectively, improve the exploration strategy and learning rate of the Q-learning algorithm so that the improved algorithm has better decision-making performance. Combined with the above analysis of the cognitive electronic jamming decision-making model, a cognitive electronic jamming decision-making algorithm based on improved Q-learning is proposed. The algorithm flow is shown in Figure 4.

The specific steps of the improved Q-learning algorithm obtained from Figure 4 are as follows:

Step 1. Initialize the function Q, discount factor, and initial temperature and set the maximum number of iterations.

Assuming that the number of radar working state is M, and the number of jamming action is N, the function Q is initialized to a zero matrix of $M \times N$, and the rows of the matrix represent the radar working state. The columns represent the possible jamming actions.

Step 2. When the iteration rule is met, the maximum number of iterations is reached and terminates the learning process and output the optimal or suboptimal jamming strategy. Otherwise, go to step 3.

Step 3. According to the results of cognitive electronic reconnaissance, identify the current radar working state and analyze the threat level in this state.

Step 4. Determine whether the current state is the target state. If it is, use equation (4) to reduce the temperature control parameters and return to step 2. Otherwise, go to step 5.

Step 5. According to the Metropolis criterion in the SA algorithm, use equation (3) to select jamming actions and change the current state to the next state.

Step 6. Analyze the changes in the radar threat level before and after the jamming and calculate the reward value of the jamming action by equation (10).

Step 7. Iteratively update the function Q according to equations (1) and (6)–(9) and set the converted state after jamming to the current state.

Step 8. Define the difference between the sum Γ_Q of all Q before and after a iteration is $\Delta(\Gamma_Q)$. When $\Delta(\Gamma_Q)$ is less than the convergence threshold, terminate the learning process and output the optimal jamming strategy according to equation (2). Otherwise, return to step 4.

Combined with the implementation steps of the improved Q-learning algorithm, the specific pseudocode is summarized as follows:

4. Simulation Experiment and Analysis of Results

To achieve improved Q-learning algorithm verification and process experimental results, the simulation experiment platform of this paper is as follows:

(i) The operating system is Windows 10

(ii) The CPU is Intel(R) Core (TM)i7 2.6GHz

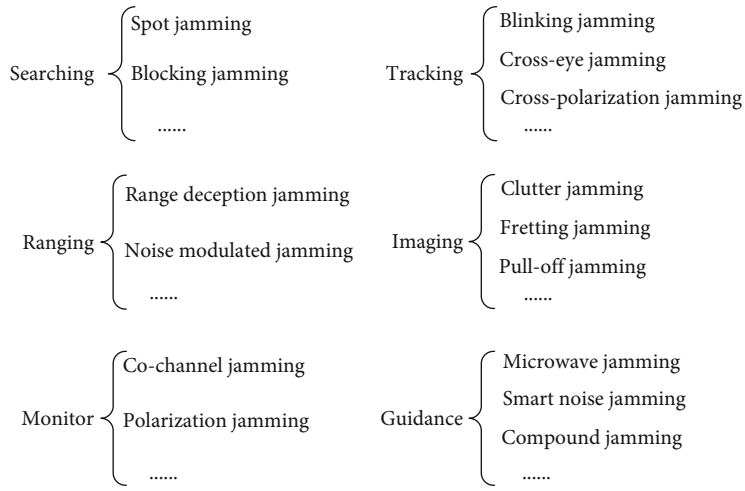

Figure 3: Jamming action corresponding to working state.

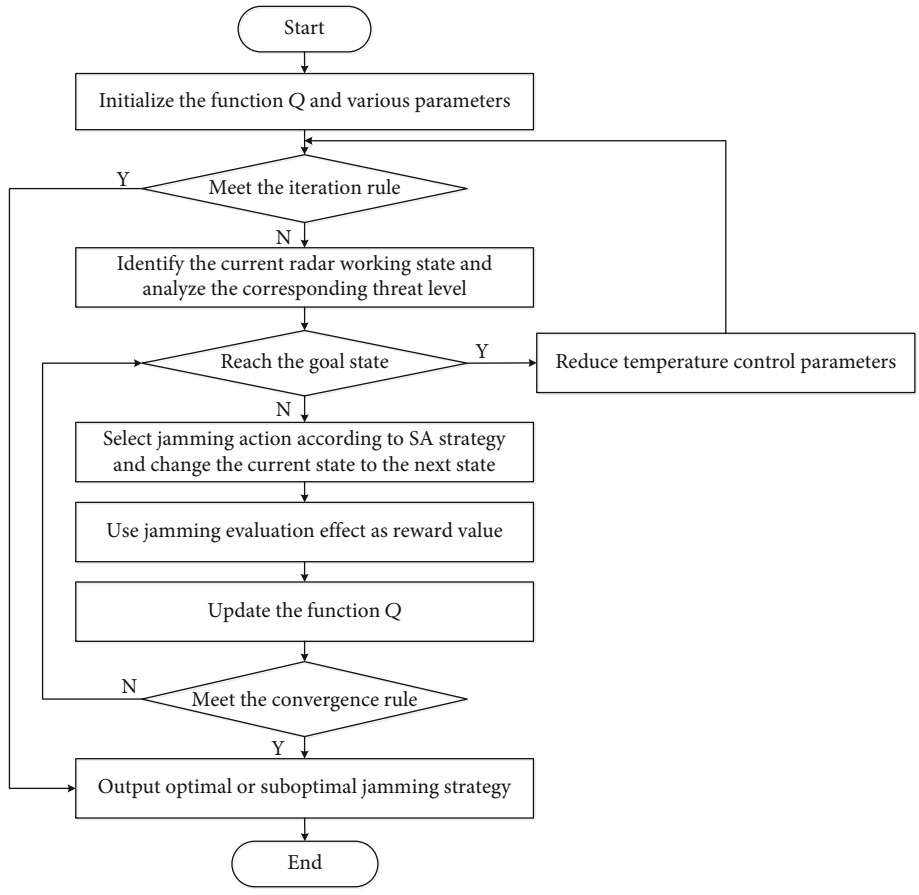

Figure 4: Flow of the improved Q-learning algorithm.

(iii) The memory is 8 GB

(iv) The programming tool is MATLAB R2016b

4.1. Experimental Simulation Settings. From the analysis in Section 3.1, it can be seen that the multifunction radar has multiple working states such as search, tracking, and ranging. The jamming actions adopted by the jammer include suppressive jamming and deceptive jamming. This experiment assumed that the multifunctional radar had 6 working states, and its threat level from high to low was $S_1 > S_2, S_3 > S_4, S_5 > S_6$. S_6 was the target state. The transition diagram between each working state is shown in Figure 5.

Input: jamming action set A, working state set S, initialization matrix Q, reward matrix R, discount factor γ and other parameters.
Output: optimal jamming strategy.
1: Begin.
2: While $(k \le K)//K$ is the maximum number of iterations.
3: For $(m = 1 \text{ to } M)//M$ is the number of restart cycles.
4: Calculate the restart period τ according to equation (9);
5: For $(\beta = 1 \text{ to } \tau)$.
6: Update the learning rate α_k according to equation (6);
7: Update the temperature according to equation (4);
8: Randomly initialize the working state;
9: **While** (the current state is not the target working state).
10: Randomly select action A for the current state from the jamming action set;
11: Choose optimal action B according to equation (2);
12: Calculate the exploration probability ε according to equation (3);
13: Generate random number r between [0,1];
14: If $(r < \varepsilon)$
15: action=A;
16: Else
17: action=B;
18: End if.
19: Execute the current action, update the radar state, and obtain the.
reward value according to equation (10);
20: Update the function Q according to equation (1);
21: Calculate the difference $\Delta(\Gamma_Q)$;
22: If $(\Delta(\Gamma_Q) < N)//N$ is the convergence threshold.
23: Jump out of the loop and terminate the learning process;
24: **End if**
25: End while
26: End for
27: Update the learning rate range according to equations (7) and (8), and restart the learning rate;
28: $k = \tau - \tau_0 + \beta//\tau_0$ is the initial restart period.
29: **End for**
30: **End while**
31: Output Q table to get the optimal interference strategy;
32: **End.**

Pseudocode 1

In Figure 5, a_{ij} represents the required jamming action from state i to state j.

Combining the definition of the radar working state transition diagram and the reward function, the reward matrix R is obtained as follows:

$$R = \begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \end{array} \begin{array}{c} \begin{array}{cccccc} S_1 & S_2 & S_3 & S_4 & S_5 & S_6 \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & -1 & 1 & 0 & 0 \\ -1 & -1 & 0 & 1 & 1 & 0 \\ 0 & -1 & -1 & 0 & -1 & 100 \\ 0 & 0 & -1 & -1 & 0 & 100 \\ 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix} \end{array}. \tag{11}$$

In this paper, the relevant parameters in the improved Q -learning algorithm were set as follows:

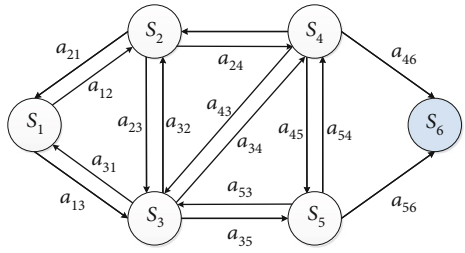

Figure 5: Transition diagram of radar working state.

The calculation equations of the learning rate range in the m_{th} restart cycle are expressed as follows:

$$\begin{aligned} \alpha_{\max}^{m} &= \alpha_{\max}^{0} * 0.6^{m}, m \in [1, M], \\ \alpha_{\min}^{m} &= \alpha_{\min}^{0} * 0.85^{m} m \in [1, M]. \end{aligned} \tag{12}$$

4.2. Algorithm Evaluation Index. In this paper, the sum Γ_Q of all values in the matrix Q is used as the basis for judging whether the algorithm had converged. When the change of

Table 1: Parameter settings of the improved Q-learning algorithm.

Parameter description	Parameter settings
The maximum number of iterations	$K = 200$
Convergence threshold	$N = 10^{-7}$
Discount factor	$\gamma = 0.8$
Cooling parameter	$\lambda = 0.92$
Initial temperature	$T_0 = 800$
The number of restart cycles	$M = 3$
Initial restart cycle	$\tau_0 = 30$
Amplification factor	$\kappa = 2$
Initial lower bound of learning rate	$\alpha_{\min}^0 = 0.3$
Initial upper bound of learning rate	$\alpha_{\max}^0 = 0.8$

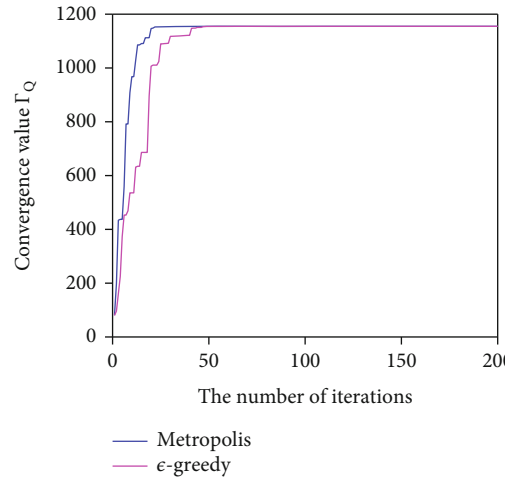

Figure 6: Comparison curve of two exploration strategies.

Γ_Q in two adjacent experiments is less than the convergence threshold, it indicates that the algorithm has converged. To quantitatively evaluate the algorithm's performance, this paper took the number of iterations during convergence as the evaluation index of the convergence speed, and the fewer the number of convergence steps, the better the effect.

4.3. Analysis of Parameters' Impact on the Jamming Decision-Making Performance

4.3.1. Comparison of Exploration Strategies. The exploration strategy affects the accuracy and timeliness of selecting the best jamming action. It is generally set to a constant value. A smaller value is likely to lead to premature maturity. A larger value can ensure that the algorithm is fully exploratory in the early stage, but it makes the algorithm produce oscillations in the later stages and difficult to converge quickly. This paper adopted the Metropolis criterion in the SA algorithm to improve the exploration strategy to solve the above problems. To facilitate the comparison of the exploration strategy's impact on the performance of jamming decision-making, this paper adopted two exploration strategies, such as the SA algorithm and the ε-greedy algorithm, initialized the learning rate $\alpha = 0.8$ and set $\varepsilon = 0.8$ in the experiment. The other parameter settings in the algorithm were the same as those in Table 1. The relationship between the convergence value Γ_Q of different exploration strategies and the number of iterations is shown in Figure 6.

As presented in Figure 6, both of the two methods' convergence values can finally converge to 1154.9. The ε-greedy algorithm adopted a fixed action selection probability, which made the Q-learning algorithm oscillate at the end of the iteration. It started to converge when it reached the 76th generation. The SA algorithm adopted the Metropolis criterion, making it possible to keep a large value in the early iteration to fully explore. During the iteration process, due to the cooling strategy, the ε gradually became smaller, making the Q-learning algorithm converge quickly, and it tended to be convergent in the 45th generation. Therefore, compared with the ε-greedy algorithm, the dynamic adaptive change of the exploration probability improved by the SA algorithm meets the requirement of fully exploring to avoid falling into the local optimum and has a faster convergence speed.

4.3.2. Comparison of Learning Rate. The learning rate represents the learning ability of the decision-making system for each increment. In the traditional Q-learning algorithm, the learning rate is constant in most cases and is generally set to $\alpha = 0.8$. However, a fixed α is difficult to jump out of the locally optimal solution at the later iteration. The study [29] has proposed an adaptive learning rate to replace the fixed learning rate. The learning rate gradually decreases as the number of iterations increases. The calculation equations are shown in (13).

$$\alpha_k^0 = \frac{1}{\sqrt{\lfloor k/10 \rfloor + 2}},$$
$$\alpha_k = \alpha_k^0 + \frac{1 - \alpha_k^0}{n(s,a) + 1}, \tag{13}$$

where k is the number of iterations, and $n(s,a)$ is the number of $Q(s,a)$'s traversals in the k_{th} iteration.

To verify the learning rate's impact on the performance of jamming decision-making, this experiment adopted the traditional ε-greedy exploration strategy and initialized $\varepsilon = 0.8$. Combining the improvement equation of learning rate in Section 2.3 and the related parameter settings of the improved Q-learning algorithm, the proposed SGDR learning rate, the adaptive declining learning rate [29], and the fixed learning rate are shown in Figure 7.

As shown in Figure 7, the fixed α was always kept constant throughout the iteration process, and it was easy to fall into the local optimum at the latter iteration. The adaptive α kept a small value in the later iteration for the global optimization, but too small α affected the convergence speed. Due to the introduction of the warm restart mechanism, the SGDR α was maintained at a large value in the early iteration, and the learning efficiency was improved. The SGDR α was increased repeatedly in the later iteration, which

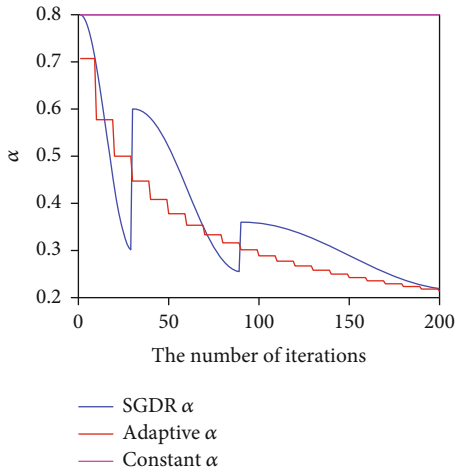

Figure 7: Comparison curve of three learning rate setting methods.

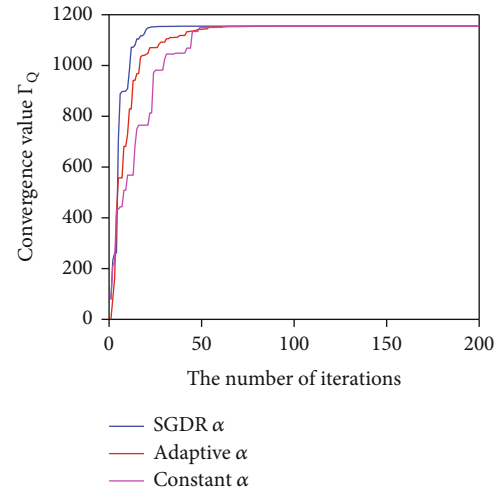

Figure 8: Comparison of convergence results of three learning rates.

avoided shocks and sped up the convergence rate to fully satisfy the timeliness and accuracy of cognitive electronic warfare.

Figure 8 compares the relationship between the sum Γ_Q of all Q elements and the number of iterations of the Q -learning algorithm under different learning rate setting methods. The convergence values of the three methods can eventually converge to 1154.9. Still, the constant α tends to converge in the 80th generation, the adaptive α tends to converge in the 64th generation, and the SGDR α tends to converge in the 42th generation. The SGDR α kept a larger value in the early iteration so that the number of Γ_Q convergence iterations was significantly shortened. With the increase of iterations, the change of the SGDR curve did not oscillate and avoided falling into the local optimum. However, the adaptive α and constant α iterative curves oscillated at the end of the iteration. Thus, the improved learning rate method proposed in this paper is the best, followed by the adaptive learning rate, and the constant learning rate has the worst convergence effect. Thus, the learning rate improved by SGDR is reasonable. It further illustrates that the improved method can overcome the shortcomings of slow convergence in the later iteration and falling into local optimum. When the number of iterations was 42, the algorithm improved by SGDR converged and had learned the optimal jamming strategy. If continue to learn and train, it would increase the cost of learning time.

4.3.3. Comparison of Discount Factor. According to equation (1), the discount factor γ also has a certain impact on the decision-making performance of the system, and the value range is (0,1]. γ represents the importance attached to future rewarding. The larger the γ, the more the agents tend to consider all possible states in the future, which means that the training is more difficult. With the continuous reduction of γ, the rewarding of possible states in the future has less and less impact on the Q value, which means that the agent only pays attention to several possible states at present. A too large discount factor can easily lead to difficult algorithm

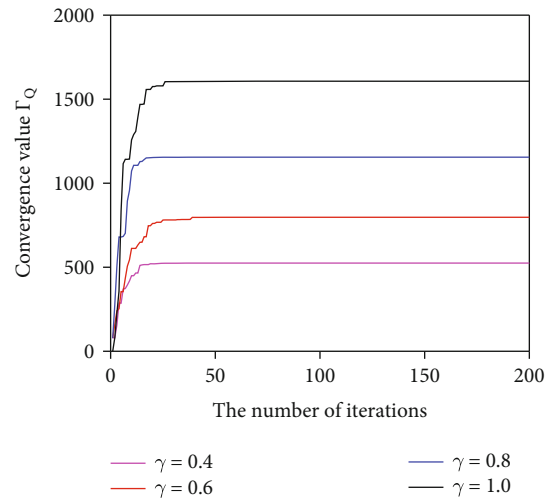

Figure 9: Comparison of convergence results of different discount factors.

convergence, while a too small discount factor is easy to fall into local optimization.

To test the influence of discount factor on the jamming decision-making performance of Q-learning algorithm and ensure the universality and reference significance of the test results, this experiment initialized $\varepsilon = 0.8$ and $\alpha = 0.8$ according to the general empirical value of Q-learning. The test simulation results are shown in Figure 9.

Figure 9 shows the influence of discount factor γ on the number of iterations of the Q-learning algorithm. As shown in Figure 9, when γ's value is set to 0.4, the convergence of the Q-learning algorithm is stable. Still, it cannot converge to the optimal solution within the specified number of iterations. As γ continues to increase, the convergent value Γ_Q increases, which means that the discount of future rewarding is getting smaller and smaller. When γ's value is set to 1, the Q-learning algorithm oscillates when the early iteration speed is fast, resulting in too large initial solution and a higher probability of converging to the suboptimal solution.

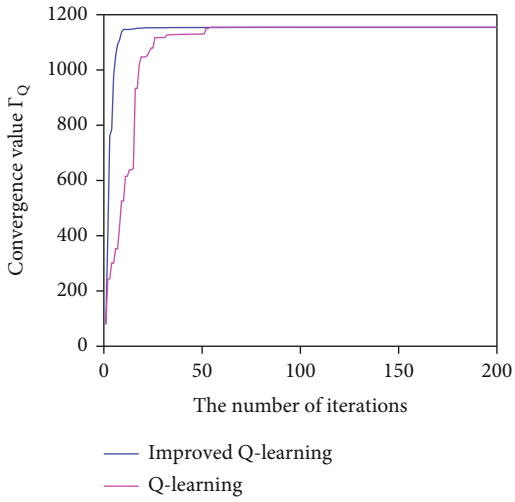

Figure 10: Comparison of two algorithm convergence results.

Table 2: Q value convergence result table.

$Q(S, a)$	Convergence value
$Q(S_1, a_{12})$	65.6
$Q(S_1, a_{13})$	65.8
$Q(S_2, a_{21})$	51.4
$Q(S_2, a_{23})$	63.8
$Q(S_2, a_{24})$	81.0
$Q(S_3, a_{31})$	51.5
$Q(S_3, a_{32})$	63.1
$Q(S_3, a_{34})$	80.5
$Q(S_3, a_{35})$	81.0
$Q(S_4, a_{42})$	57.3
$Q(S_4, a_{43})$	63.8
$Q(S_4, a_{45})$	78.4
$Q(S_4, a_{46})$	100.0
$Q(S_5, a_{53})$	63.8
$Q(S_5, a_{54})$	79.0
$Q(S_5, a_{56})$	100.0

When the value is set to 0.8, the initial solution is small, the later convergence is stable, and the convergence speed is faster. Therefore, the best discount factor value in this paper should be set to 0.8.

4.4. Decision Simulation and Result Analysis. To verify the effectiveness of the abovementioned improved Q-learning algorithm, the traditional Q-learning algorithm and the improved Q-learning algorithm were compared and tested in the same simulation environment. The curve of Γ_Q with the number of iterations is shown in Figure 10.

As presented in Figure 10, both algorithms could eventually converge, but the improved Q-learning algorithm stabilized after iterating to the 35th generation due to the introduction of the SA algorithm and the SGDR method. However, the Q-learning algorithm oscillated at the end of the iteration, and it only tended to converge until the 60th generation. It shows that the search space in the early stage is large, which leads to a long time to converge to the optimal solution for the first time, and the algorithm is unstable in the iterative process. The improved $Q-$learning algorithm had a significantly faster convergence speed than the traditional Q-learning algorithm. This is because the SA algorithm reduces the exploration probability, and the Q matrix does not change in the later iteration process of the algorithm, which reduces the possibility of Q-learning deviating from the optimal solution to avoid oscillation. In addition, the SGDR method adjusts the learning rate through the hot restart mechanism. It repeatedly increases the learning rate in the later stage of the iteration to improve the convergence speed. The improved Q-Learning algorithm effectively solves the balance problem about the exploration and utilization and the selection of learning rate. The improved Q -learning algorithm speeds up the training speed of learning. It improves the efficiency of jamming decision-making, which further proves the effectiveness and feasibility of the algorithm applied to cognitive electronic jamming decision-making.

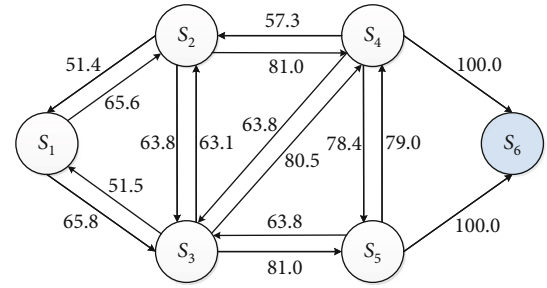

Figure 11: State action value.

The improved Q-learning algorithm was used to simulate the decision-making process. After many iterations, the final convergence results are shown in Table 2.

It can be seen from Table 2 that after applying jamming actions to different states, the convergence value Q is different. For example, when the a_{21}, a_{23}, and a_{24} are applied to S_2, the convergence values Q are, respectively, 51.4, 63.8, and 81.0. Therefore, the jammer prefers to take jamming action a_{24} to make the state of radar transition from S_2 to S_4. Since S_6 is the target state, the convergence value Q from S_4 and S_5 to S_6 after applying the jamming action is maximum, that is, 100. Sum all the values in Table 2 to get the final Γ_Q, and the result is 1146.

According to Table 2, the jamming path can be obtained after applying the best jamming action to any radar working state, as shown in Figure 10.

The value of the arrow in Figure 11 represents the value Q after the best jamming action is adopted. According to the selection action strategy with the maximum Q value, it can be seen that when the radar is in any state, the final convergence value Q will guide the jammer to choose the optimal jamming action to make the radar work state gradually shift

to the target state S_6. For example, when the detected state is S_1, the best jamming action strategy is $S_1 \longrightarrow^{a_{13}} S_3 \longrightarrow^{a_{35}} S_5 \longrightarrow^{a_{56}} S_6$, and the sum of the values Q converges to 246.8. For another example, when the initial state of the radar is S_2, the threat level of the radar is minimized by implementing the optimal jamming strategy $S_2 \longrightarrow^{a_{24}} S_4 \longrightarrow^{a_{46}} S_6$.

5. Conclusions

Aiming at radar jamming decision-making, a cognitive electronic jamming decision-making model based on improved Q-learning is proposed. The conclusions obtained are as follows:

(1) By the Metropolis criterion of the SA algorithm and the SGDR, the improved Q-learning algorithm improved the learning efficiency of the algorithm, sped up the convergence speed, and avoided falling into the local optimum

(2) By applying the improved Q-learning algorithm to jamming decision-making, the correspondence between radar working state and jamming strategy was established, a cognitive electronic interference decision-making model based on improved Q -learning was constructed, and the specific Q-learning algorithm was proposed. The Q value convergence rule and the iteration number rule are used as the learning termination rule to avoid the waste of jamming resources

(3) This model overcomes the shortcomings of the traditional Q-learning algorithm, such as slow convergence speed and local optimization. By interacting with the radar in an environment without any prior information, the jammer can continuously and autonomously learn and finally find the optimal jamming strategy

This paper also has some limitations. For example, this paper takes the limited working state of a single radar as the research object. However, the radar is networked in the actual combat environment, and the working state is diversified. At this moment, the decision-making performance of this method will decrease. Therefore, our next step will study the jamming decision method that combines deep learning and reinforcement learning. For another example, except the SA algorithm mentioned in this paper, other recently proposed metaheuristic algorithms such as monarch butterfly optimization (MBO) [30], moth search (MS) algorithm [31], slime mold algorithm (SMA) [32], and harris hawks optimization (HHO) [33] can also be used to solve the problem of exploration strategy. Therefore, we will use the above metaheuristic algorithm to optimize the Q-learning algorithm and make a comparative analysis in the next step.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (61773389), Natural Science Foundation of Shaanxi Province (2021KJXX-22, 2020JQ-298), Postdoctoral Science Foundation of China (2019M663635), and Special Support Plan for High-level Talents in Shaanxi Province.

References

[1] Z. Yang, S. Guangya, and W. Yanzheng, "Modelling and simulation of cognitive electronic attack under the condition of system of systems combat," *Defence Science Journal*, vol. 70, no. 2, pp. 183–189, 2020.

[2] W. David, S. Teresa, and C. Vasu, "Game theoretic decision support framework for electronic warfare applications," in *Proceedings of the IEEE Radar Conference*, pp. 1–5, Philadelphia, PA, USA, May 2016.

[3] Y. L. Gao, Y. Xiao, M. M. Wu, M. Xiao, and J. L. Shao, "Game theory-based anti-jamming strategies for frequency hopping wireless communications," *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5314–5326, 2018.

[4] H. W. Sun, N. N. Tong, and F. J. Sun, "Jamming design selection based on D-S theory," *Journal of Projectiles, Rockets, Missiles and Guidance*, vol. 2, pp. 140–142, 2003.

[5] W. G. Li and C. H. Wu, "Research on self-learning model of intelligent radar jamming system," *Modern Defence Technology*, vol. 37, no. 1, pp. 83–86, 2009.

[6] F. Ye, F. Che, and L. P. Gao, "Multiobjective cognitive cooperative jamming decision-making method based on tabu search-artificial bee colony algorithm," *International Journal of Aerospace Engineering*, vol. 2018, Article ID 7490895, 10 pages, 2018.

[7] F. Ye, F. Che, and H. Tian, "Cognitive cooperative-jamming decision method based on bee colony algorithm," in *Proceedings of the Progress in Electromagnetics Research Symposium-Fall*, pp. 531–537, Singapore, November 2017.

[8] W. Pan, X. Jin, H. X. Xie, and Y. Xia, "Radar jamming strategy allocation algorithm based on improved chaos genetic algorithm," in *Chinese Control and Decision Conference*, pp. 4478–4483, Hefei, China, August 2020.

[9] Y. J. Tan, W. Pan, Y. Han, and S. L. Xu, "Research on force assignment of radar jamming system based on chaos genetic algorithm," in *Proceedings of the Chinese Control and Decision Conference*, pp. 1193–1197, Nanchang, China, June 2019.

[10] W. Y. Gu, L. N. Zhu, Y. H. Bu, W. W. Yue, and Y. G. Fan, "Collaborative jamming decision-making mechanism using ant colony algorithm in electromagnetic antagonism," in *Proceedings of the 20th IEEE International Conference on Communication Technology*, pp. 1645–1649, Nanning, China, October 2020.

[11] W. L. Liu, Y. J. Gong, W. N. Chen, Z. Q. Liu, H. Wang, and J. Zhang, "Coordinated charging scheduling of electric vehicles: a mixed-variable differential evolution approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 5094–5109, 2020.

[12] S. Zhou, L. Xing, X. Zheng, N. du, L. Wang, and Q. F. Zhang, "A self-adaptive differential evolution algorithm for scheduling a single batch-processing machine with arbitrary job sizes and release times," *IEEE Transactions on Cybernetics*, vol. 51, no. 3, pp. 1430–1442, 2021.

[13] F. Q. Zhao, L. X. Zhao, L. Wang, and H. B. Song, "An ensemble discrete differential evolution for the distributed blocking flowshop scheduling with minimizing Makespan criterion," *Expert Systems with Applications*, vol. 160, no. 113678, pp. 1–35, 2020.

[14] F. Q. Zhao, L. X. Zhao, Y. Zhang, W. M. Ma, C. Zhang, and H. B. Song, "A hybrid discrete water wave optimization algorithm for the no-idle flowshop scheduling problem with total tardiness criterion," *Expert Systems with Application*, vol. 146, no. 113166, pp. 1–41, 2019.

[15] L. Safatly, M. Bkassiny, M. Al-Husseini, and A. El-Hajj, "Cognitive radio transceivers: RF, spectrum sensing, and learning algorithms review," *International Journal of Antennas and Propagation*, vol. 2014, Article ID 548473, 21 pages, 2014.

[16] E. Akanksha, N. S. Jyoti, and K. Gulati, "Review on reinforcement learning, research eEvolution and scope of application," in *Proceedings of the 5th International Conference on Computing Methodologies and Communication*, pp. 1416–1423, Erode, India, April 2021.

[17] A. K. Sadhu and A. Konar, "An efficient computing of correlated equilibrium for cooperative Q-learning-based multi-robot planning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 8, pp. 2779–2794, 2020.

[18] G. P. Kontoudis and K. G. Vamvoudakis, "Kinodynamic motion planning with continuous-time Q-learning: an online, model-free, and safe navigation framework," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 12, pp. 3803–3817, 2019.

[19] J. N. Li, T. Y. Chai, F. L. Lewis, Z. T. Ding, and Y. Jiang, "Off-policy interleaved Q -learning: optimal control for affine nonlinear discrete-time systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 5, pp. 1308–1320, 2019.

[20] Y. J. Peng, Q. Chen, and W. J. Sun, "Reinforcement Q-learning algorithm for H∞tracking control of unknown discrete-time linear systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 11, pp. 4109–4122, 2020.

[21] D. S. Wang, W. Z. Zhang, H. He, and Y. C. Tian, "Efficient hybrid central processing unit/ input–output resource scheduling for virtual machines," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 3, pp. 2714–2724, 2021.

[22] F. X. Tang, Y. B. Zhou, and N. Kato, "Deep reinforcement learning for dynamic uplink/downlink resource allocation in high mobility 5G hetnet," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 12, pp. 2773–2782, 2020.

[23] Q. Xing, W. G. Zhu, and X. Jia, "Intelligent Countermeasure Design of Radar Working-Modes Unknown," in *Proceedings of the IEEE International Conference on Signal Processing, Communications and Computing*, pp. 1–5, Xiamen, China, October 2017.

[24] Q. Xing, W. G. Zhu, and X. Jia, "Research on method of intelligent radar confrontation based on reinforcement learning," in *Proceedings of the 2nd IEEE International Conference on Computational Intelligence and Applications*, pp. 471–475, Beijing, China, September 2017.

[25] K. Li, B. Jiu, H. W. Liu, and S. L. Yuan, "Reinforcement learning based anti-jamming frequency hopping strategies design for cognitive radar," in *Proceedings of the IEEE International Conference on Signal Processing, Communications and Computing*, pp. 1–5, Qingdao, China, September 2018.

[26] B. K. Zhang and W. G. Zhu, "A cognitive jamming decision method for multi-functional radar based on Q-learning," *Telecommunication Engineering*, vol. 60, no. 2, pp. 129–136, 2020.

[27] Y. Z. Wang and S. Q. Zhu, "Main-Beam range deceptive jamming suppression with simulated annealing FDA-MIMO radar," *IEEE Sensors Journal*, vol. 20, no. 16, pp. 9056–9070, 2020.

[28] I. Loshchilov and F. Hutter, *SGDR: Stochastic Gradient Descent with Restarts*, pp. 1–16, 2016, https://arxiv.org/abs/1608.03983.

[29] Z. Li, *Radar Radiation Source and Working State Recognition*, Xidian University, Xi'an, China, 2019.

[30] G. G. Wang, S. Deb, and Z. H. Cui, "Monarch butterfly optimization," *Neural Computing and Applications*, vol. 31, no. 7, pp. 1995–2014, 2019.

[31] P. Singh, S. K. Bishnoi, and N. K. Meena, "Moth search optimization for optimal DERs integration in conjunction to OLTC tap operations in distribution systems," *IEEE Systems Journal*, vol. 14, no. 1, pp. 880–888, 2020.

[32] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: a new method for stochastic optimization," *Future Generation Computer Systems*, vol. 111, no. 1, pp. 300–323, 2020.

[33] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. L. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.