remote sensing

MDPI

Article

A Cognitive Electronic Jamming Decision-Making Method Based on *Q-Learning* and Ant Colony Fusion Algorithm

Chudi Zhang, Yunqi Song, Rundong Jiang, Jun Hu * and Shiyou Xu

School of Electronics and Communication Engineering, Sun Yat-sen University, Shenzhen 528406, China; zhangchd@mail2.sysu.edu.cn (C.Z.); songyq26@mail2.sysu.edu.cn (Y.S.); jiangrd3@mail2.sysu.edu.cn (R.J.); xushy36@mail.sysu.edu.cn (S.X.)

* Correspondence: hujun25@mail.sysu.edu.cn

Abstract: In order to improve the efficiency and adaptability of cognitive radar jamming decision-making, a fusion algorithm (*Ant-QL*) based on ant colony and *Q-Learning* is proposed in this paper. The algorithm does not rely on a priori information and enhances adaptability through real-time interactions between the jammer and the target radar. At the same time, it can be applied to single jammer and multiple jammer countermeasure scenarios with high jamming effects. First, traditional *Q-Learning* and *DQN* algorithms are discussed, and a radar jamming decision-making model is built for the simulation verification of each algorithm. Then, an improved *Q-Learning* algorithm is proposed to address the shortcomings of both algorithms. By introducing the pheromone mechanism of ant colony algorithms in *Q-Learning* and using the ε-greedy algorithm to balance the contradictory relationship between exploration and exploitation, the algorithm greatly avoids falling into a local optimum, thus accelerating the convergence speed of the algorithm with good stability and robustness in the convergence process. In order to better adapt to the cluster countermeasure environment in future battlefields, the algorithm and model are extended to cluster cooperative jamming decision-making. We map each jammer in the cluster to an intelligent ant searching for the optimal path, and multiple jammers interact with each other to obtain information. During the process of confrontation, the method greatly improves the convergence speed and stability and reduces the need for hardware and power resources of the jammer. Assuming that the number of jammers is three, the experimental simulation results of the convergence speed of the *Ant-QL* algorithm improve by 85.4%, 80.56% and 72% compared with the *Q-Learning*, *DQN* and improved *Q-Learning* algorithms, respectively. During the convergence process, the *Ant-QL* algorithm is very stable and efficient, and the algorithm complexity is low. After the algorithms converge, the average response times of the four algorithms are 6.99×10^{-4} s, 2.234×10^{-3} s, 2.21×10^{-4} s and 1.7×10^{-4} s, respectively. The results show that the improved *Q-Learning* algorithm and *Ant-QL* algorithm also have more advantages in terms of average response time after convergence.

Keywords: cognitive electronic jamming decision-making; reinforcement learning; ant colony algorithm; cooperative jamming decision-making

Citation: Zhang, C.; Song, Y.; Jiang, R.; Hu, J.; Xu, S. A Cognitive Electronic Jamming Decision-Making Method Based on *Q-Learning* and Ant Colony Fusion Algorithm. *Remote Sens.* **2023**, *15*, 3108. https://doi.org/10.3390/rs15123108

Academic Editors: Khaled Rabie, Pascal Lorenz, Muhammad Asghar Khan, Syed Agha Hassnain Mohsan and Muhammad Shafiq

Received: 16 April 2023
Revised: 11 June 2023
Accepted: 12 June 2023
Published: 14 June 2023

1. Introduction

Electronic warfare (EW) plays an important role in modern warfare [1]. With the improvement in technologies such as artificial intelligence (AI) and software radio, radars and jammers with cognitive capabilities have been developed significantly [2–8]. These new technologies have rendered traditional EW means inadequate for adapting to the modern battlefield environment. For example, a cognitive radar has strong anti-jamming and target detection capabilities. Therefore, conventional jammers are unable to create effective jamming against them.

1.1. Cognitive Electronic Warfare

Cognitive electronic warfare (CEW) is widely believed to have a more significant role in future warfare [9–11]. The task of CEW can be broken down into three steps. First, the jamming system identifies the operating state of the reconnaissance target based on its radar signal. Then, by evaluating the effectiveness of the current jamming action, the jamming system establishes the optimal correlation between the target radar states and the current jamming techniques. Finally, based on the optimal jamming strategy generated, it guides the subsequent scheduling of jamming resources and implements jamming [9].

CEW is a combination of cognitive concepts and EW technology, in which cognition is a process that mimics human beings in information processing and knowledge application. The progress of cognitive technology is due to the development of AI technology. In the 1980s, AI technology was proposed and applied to EW in order to enhance the agility and adaptability of EW operations [12–14]. However, the research results were not publicly available due to security concerns. Up until 2010, DARPA released projects such as Blade, CommEex and ARC [4–7], following which the application of AI in EW developed rapidly. Reinforcement learning (RL) is known as the hope of real AI and is one of the most active research areas in AI [15,16]. It focuses on how agents change based on the state of their environment and decide what actions to take to maximize the cumulative reward. As shown in Figure 1, the development of CEW technology over the past few decades has benefited from the advancement and integration of multiple technologies. As a key component of CEW, radar jamming decision-making methods can be divided into two categories: traditional radar jamming decision-making methods and cognitive radar jamming decision-making methods.

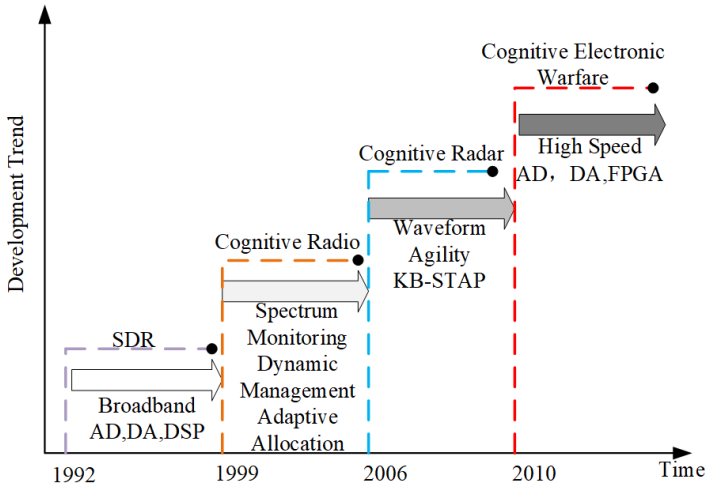


Figure 1. The development path of CEW.

1.2. Traditional Radar Jamming Decision-Making Methods

The traditional radar countermeasure model is shown in Figure 2. The jamming system is only fixed according to the basic parameters of reconnaissance and an a priori radar database to call the jamming style of the jamming resource base. The system is unable to adjust the jamming method based on the jamming effect and environmental information, which would lead to inefficient jamming.

Currently, traditional radar jamming decision-making methods mainly include the following: game theory-based methods, template-matching-based methods and inference-based methods. David et al. [17] proposed a framework that uses game theory principles to provide an autonomous decision-making for appropriate electronic attack actions for a given scenario. Gao et al. [18] established a profit matrix based on the principle of minimizing losses and maximizing jamming gains and used a Nash equilibrium strategy to solve for the optimal jamming strategy. They relied on the establishment of the profit

matrix, which is only applicable to radar systems with constant parameter characteristics. Sun et al. [19] proposed an electronic jamming pattern selection method based on D-S theory. Li and Wu [20] proposed an intelligent decision-making support system (IDSS) design method based on a knowledge base and problem-solving units. The method has wide applicability but relies too much on posterior probability and has poor real-time performance. Ye et al. [21,22] proposed a cognitive collaborative jamming decision-making method based on a swarm algorithm, which finds the global optimal solution through the process of searching for quality resources using a swarm. There are many similar population intelligence algorithms, such as genetic algorithms [23,24], ant colony algorithms [25], differential evolutionary algorithms [26–28] and water wave optimization algorithms [29]. All these algorithms can be useful for solving jamming decision-making models, but the autonomy, real time use and accuracy cannot fully meet the requirements of CEW. These algorithms are mainly applicable to radars with constant feature parameters and rely heavily on adequate a priori information.

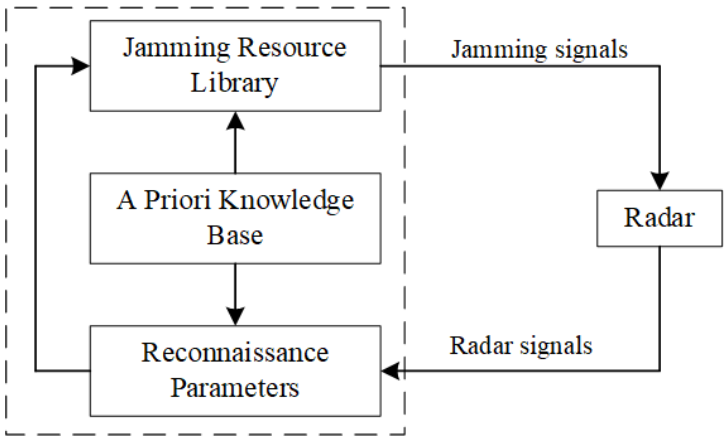


Figure 2. Architecture of traditional radar countermeasure.

1.3. Cognitive Radar Jamming Decision-Making Methods

A CEW system can be defined as an Observe, Orient, Decide and Act (OODA) cycle with an adaptation capability (i.e., AI). Figure 3 shows the working process of a typical OODA. The jammer first reconnoiters and sorts out the target signals from the threat environment system, then measures the parameters and identifies the target state. After the jammer adapts to the target state change, jammer evaluates the jamming efficiency, schedules jamming resources and optimizes jamming parameters to achieve effective jamming. The biggest difference between CEW and EW is that CEW and the environment can form a closed-loop system.

The cognitive intelligence of the agent is reflected in the process from cognition to memory, judgment, imagination and expression of the result. The cognitive radar jamming decision-making model is shown in Figure 4. Its characteristics are as follows:

(1) **Observe**: The agent is able to perceive relevant information about the external environment and to obtain relevant information.
(2) **Orient**: The agent can use existing knowledge to guide thinking activities based on the perceived information.
(3) **Decide**: The agent is able to learn independently, interact with the external environment and adapt to changes in the external environment.
(4) **Act**: The agent is able to make autonomous decisions in response to changes in the external environment.

In the increasingly complex and changing electromagnetic environment, jamming and anti-jamming techniques are emerging. During countermeasures, the position between the radar and jammer dynamically changes, and power and distance greatly influence the effectiveness of different jamming styles. In this case, it is difficult to establish a one-to-

one mapping between the specific working state of the radar and the available jamming techniques. At the same time, with the rapid development of new weapons and equipment, many new systems and multifunctional radars have emerged, and the jamming decision-making methods relying on a priori information and template matching are completely unable to adapt to current battlefield environments. Therefore, the research on intelligent radar jamming decision-making methods adapted to CEW has received much attention. CEW is a game process between two intelligences: the jammer and the radar. The research in this paper focuses on the countermeasure decision-making algorithm of the jammer.

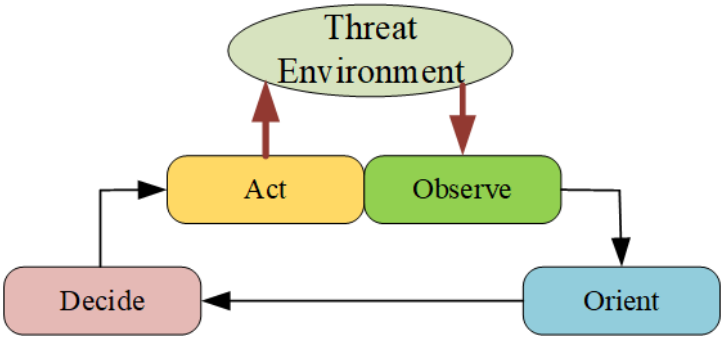


Figure 3. Typical OODA.

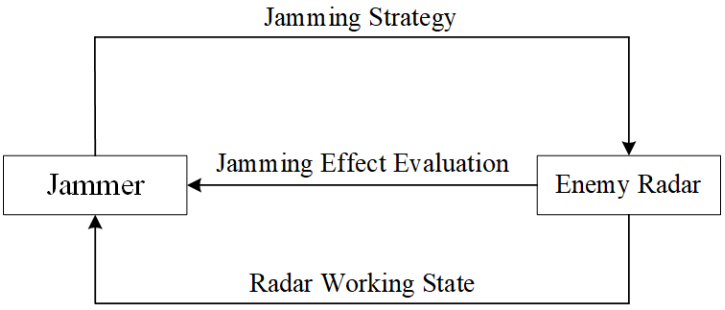


Figure 4. Cognitive electronic jamming decision-making model.

Cognitive radar jamming decision-making has received extensive attention and development in recent years. Cognitive radar jamming decision-making is the ability to establish the best correspondence between radar and jamming styles in radar countermeasure systems through threat target awareness. Radar jamming decision-making based on the RL algorithm has been the focus of research in recent years. The RL algorithm generates the optimal strategy by continuously interacting with the environment [30,31].

The *Q-Learning* algorithm is a typical time-series differential RL algorithm based on model-free learning. It enables the system to learn autonomously and to make correct decisions in real time without considering environmental models or prior knowledge [32]. Therefore, compared with traditional jamming decision-making methods, *Q-Learning* algorithm-based jamming decision-making methods can realize learning while fighting, which is expected to be a major research direction and future trend. *Q-Learning* is currently widely used in robot path planning [33,34], nonlinear control [35,36] and resource allocation scheduling [37,38] and has yielded specific results in recent years for radar jamming decision-making. Xing et al. [39,40] proposed applying the *Q-Learning* algorithm to radar countermeasures for the problem of unknown radar operating modes. The jamming system continuously monitors the state of the radar target, evaluates the effectiveness of the jamming, and feeds the results to the jamming decision-making module. Li et al. [41] suggested using *Q-Learning* to train the behavior of radar systems to achieve effective jamming and to adapt to various combat scenarios. Zhu et al. [10,42,43] proposed applying the *Q-Learning* algorithm to radar jamming decision-making for a particular multifunctional radar model and discussed the effect of prior knowledge and various

hyperparameters on the convergence of the algorithm. From the simulation experimental results, it is known that by adding prior knowledge and adjusting the hyperparameters, the algorithm can improve the convergence speed and stability, and increase its robustness. Li et al. [44] designed a radar confrontation process based on the *Q-Learning* algorithm and verified the convergence of Q-values and the performance improvement in the algorithm through simulations using prior knowledge. Zhang et al. [45] proposed the *DQN* algorithm applied to radar jamming decision-making research for efficiency degradation caused by the increase in the number of target radar states. By comparing with the simulation experiments of the *Q-Learning* algorithm, this method can better learn the effect of jamming in an actual battlefield autonomously and carry out jamming decision-making for multifunctional radars.

The *Q-Learning* algorithm also has its shortcomings when applied. First, the practical application of *Q-Learning* relies on several hyperparameters in the algorithm. Most of the existing results use a fixed exploration factor. When the exploration factor is large, the algorithm will explore sufficiently during the early stage and reach the optimal solution nearer quickly. However, it does not reach the convergence value effectively and oscillates at the optimal solution attachment, creating a difficult convergence situation. When the exploration factor is small, the algorithm does not explore sufficiently during the early stage and tends to fall into the local optimum during the later stage. Therefore, ensuring both exploration sufficiency and the stability of convergence using a fixed exploration factor is difficult. There are also no fixed rules for the learning rate and discount factor, which are generally set to fixed values by researchers through experience. When the learning rate is large, the algorithm is vulnerable to a learning risk early on. When the learning rate is small, the algorithm converges slower in the later stages. Secondly, the algorithm encounters problems such as slow and unstable convergence during the convergence process, which seriously affects the decision-making effect of the intelligent systems. Therefore, it is necessary to improve the *Q-Learning* algorithm in order to improve the accuracy and efficiency of the *Q-Learning* algorithm applied to radar jamming decision-making. Li et al. [32] proposed an improved *Q-Learning* algorithm that introduced the Metropolis criterion of a simulated annealing algorithm, which effectively solved the local optimum problem in the radar jamming decision-making process. Meanwhile, a stochastic gradient descent with the warm restarts method is used for the learning rate in the algorithm, which reduces the oscillations in the late iteration and improves the depth of convergence.

1.4. Research Focus

In today's battlefield environment, multi-function radars are usually phased-array radars, which can simultaneously detect and identify targets in multiple states and directions. Using a single jammer as the jamming side is not effective in containing the radar's threat. Additionally, a single jammer's RL process will be slow when facing a radar operating in multiple states simultaneously. The longer it takes for the jammer to reach convergence, the higher the probability of the opposing radar detecting and destroying it. Therefore, clustered multiple jammers can achieve collaborative radar jamming decision-making with shared information. The clustered multiple jammers method has several advantages over using a single jammer, including the following:

(1) **Convergence speed**: Multiple jammers can converge more rapidly with target radars in multiple operating states during countermeasures, reducing the probability that individual agents will be detected and destroyed.
(2) **Coverage area**: Multi-functional radars operate in different states and orientations. Multiple jammers respond to the target radar in different orientations on the basis of information sharing.
(3) **Hardware and power limitations**: When jamming a single radar using a cluster of jammers, UAVs are often used as carriers for jammers. A single jammer requires higher requirements for power and hardware resources to implement effective jamming. Single-function jammers are overwhelmed by powerful multifunctional radars

during confrontation and can be limited by distance, power and destruction. Moreover, a single jammer is often unable to operate in multiple states simultaneously due to its size and poor antenna isolation, among other reasons. Therefore, multiple jammers form clusters based on information sharing in order to achieve a greater advantage in an actual battlefield.

In response to the above problems, a radar jamming decision-making method *Ant-QL* based on the improved *Q-Learning* and ant colony algorithm is proposed in this paper. The algorithm includes the following:

(1) **Explore strategy**: The ε-greedy algorithm is introduced to change the search factor, thus balancing exploration and exploitation in the algorithm. This method ensures that the larger exploration factors can be fully explored during the early stage and the smaller exploration factors can reach the convergence state smoothly during the later stage of the algorithm.

(2) **Pheromone mechanism**: The *Q-Learning* algorithm converges slowly and tends to fall into local optima. To address the above problems, this paper introduces the pheromone mechanism of the ant colony algorithm into the *Q-Learning* algorithm to form an improved *Q-Learning* algorithm. The optimization process of the ant colony algorithm simulates the pheromone released by ants to explore the optimal solution. By combining *Q-Learning* with pheromones, not only is the reward value learned during the interaction between the agent and the environment but also the pheromone matrix for the transition between states is obtained. The pheromone as a path guide for the agent will make the convergence time of the agent shorter, while avoiding running into the local optimum.

(3) **Termination conditions**: In the convergence algorithm based on pheromone and *Q-Learning* during iteration, we use the Q-value convergence rule as the highest priority termination condition and the limit of the number of iterations as the next highest priority termination condition. After reaching the termination state, the algorithm outputs the optimal policy and the agent reaches the optimal state. In an actual battlefield environment, the jammer maximizes savings in terms of jamming resources and performs effective jamming.

(4) **Advantages of cluster confrontation**: We extend the confrontation between jammers and target radars to a cluster confrontation scenario to achieve the goal of collaborative jamming decision-making by multiple jammers against multifunctional radars. Cluster adversarial helps to reduce the need for hardware and a power system of individual jamming systems. Additionally, effective cooperative jamming can suppress a target radar in multiple airspace/time/frequency domains at the same time. During the adversarial process, the method reduces the convergence time, thus reducing the probability of the jammers being destroyed. At the same time, clustered jammers are more conducive to achieving effective countermeasures and ensuring our dominance in the radar countermeasure process.

(5) **Termination conditions for cluster adversarial**: Determining the termination state in the learning process of a cluster jammer is crucial. In this paper, we assume that each jammer works independently, and when all the jammers have completed one round of iterations, information such as the Q-value matrix and pheromones are shared, and the shared information of all the jammers is updated. This method can greatly accelerate the iteration speed and effectively improve the stability and robustness of the algorithm.

In this paper, we construct a multifunctional radar state transfer model with the number of states at 16 and the number of jamming modes at 10, drawing on the multifunctional radar model provided by [46]. We propose an improved *Q-Learning* algorithm and an *Ant-QL* algorithm based on the ant colony algorithm. The conventional *Q-Learning* algorithm and *DQN* algorithm are simulated under the same decision-making model and used as comparison methods. After that, we perform simulation experiments on the im-

proved *Q-Learning* algorithm and *Ant-QL* algorithm and compare them with the basic methods. The simulation results show that the new methods are useful for improving the autonomous learning efficiency of the jammer, shortening the response time of intelligent decision-making and greatly enhancing the adaptability of the jamming system.

The rest of the paper is structured as follows. Section 2 details the traditional RL algorithm and the cognitive radar jamming decision-making model. Section 3 describes the improved *Q-Learning* and *Ant-QL* algorithms in detail. Section 4 gives the simulation experiments and the analysis of the results of four algorithms. Finally, Section 5 discusses some conclusions drawn from this study and some future research hotspots and approaches in this area.

2. Radar Jamming Decision-Making Model

2.1. Multifunctional Radar Signal Model

The phased array radar system is a typical multifunction radar. By modulating the antenna array units and the programmable devices inside the radar, various waveforms can be generated to achieve search, tracking and guidance functions. Viseneski et al. proposed a multilevel multifunction radar signal model [47,48]. As shown in Figure 5, the model is divided into three layers: radar word layer, radar phrase layer and radar sentence layer. Among them, the radar word layer is a fixed arrangement of a limited number of radar pulses and is the most basic signal unit. A limited number of radar words then form radar phrase, and the arrangement of radar phrases is fixed. With specific arrangement rules, radar phrases affect the multifunction radar's operational performance in different environments. Finally, radar phrases form radar sentences, which are highly symbolic forms of the radar signal sequence.

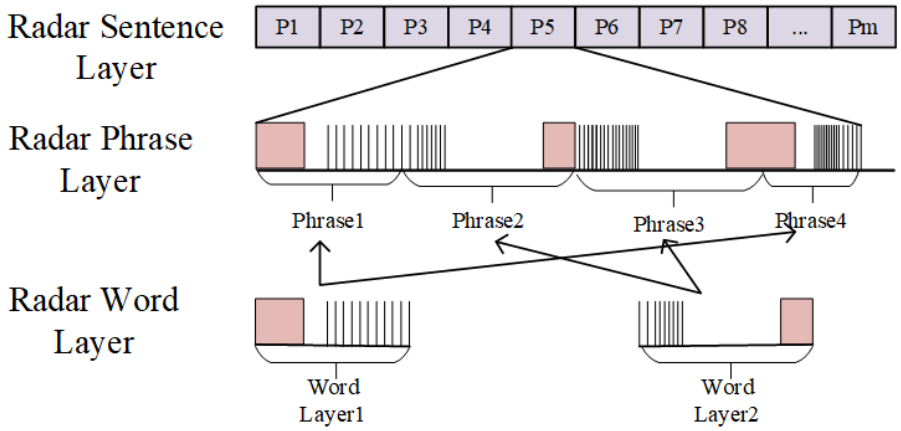


Figure 5. Multifunctional radar signal model.

2.2. Reinforcement Learning

RL is a machine learning method that studies the optimal behavioral strategies that an intelligent body learned by interacting with its environment. The method takes corresponding actions to adapt to the environment by observing the data sequences obtained from the interaction between the agent and the environment, and its essence is to learn optimal sequential decisions. The concept of RL was first proposed by Minsky in 1954 and has gradually become a research hotspot in machine learning, which is widely used in various real-world applications such as intelligent control and decision analyses.

As shown in Figure 6, *Q-Learning*, as the most applied classical RL algorithm, works in a process that can be mapped to the cognitive radar jamming decision-making process.

The *Q-Learning* algorithm maps the state S_t obtained by the agent to the radar state S_{Rt} detected by the jamming system; the optional action set A is mapped onto the set A_{jam} of scheduling schemes for different jamming resources of the jamming system; the optional action a is mapped to the jamming action a_{jam}; the reward r given by the environment

is mapped to the evaluation value r_{jam} of the jamming effectiveness; and the meaning of the $Q(S_{Rt}, a_{jam})$ function indicates the sum of the discounts obtained in the future after choosing the jamming action a_{jam} in the current radar state S_{Rt}.

In recent years, many excellent RL algorithms have been used for radar jamming decision-making, such as *Q-Learning*, *DQN*, *Double DQN* and *A3C* [32,42–46,49]. Most of these studies are based on *Q-Learning* and *DQN* algorithms. Through different modeling methods and parameter optimization, these results have verified the effectiveness and adaptability of these two classical methods.

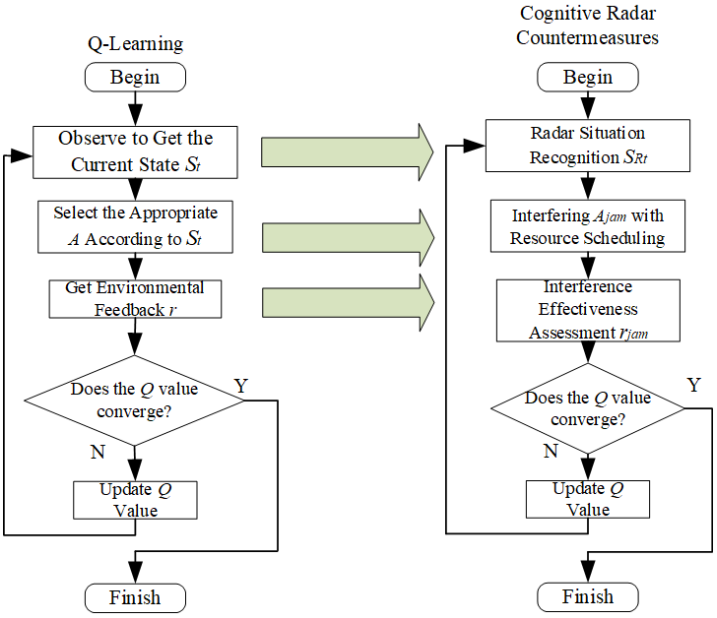


Figure 6. Mapping of RL to radar jamming decision-making.

2.2.1. *Q-Learning* Algorithm

Q-Learning is a time-series differential reinforcement learning algorithm. The specific flow of the algorithm is Algorithm 1. In the algorithm, a_t denotes the value of the action taken by the agent at the current moment, s_t denotes the state of the current environment, α denotes the learning rate, γ is the discount factor and ε is the exploration factor. At each cycle, the agent updates $Q(s, a)$ via Equation (1).

Algorithm 1 *Q-Learning* algorithm workflow.

Initialize $Q(s, a)$, the number of iterations is N, the learning rate is α, the discount factor is γ, the exploration factor ϵ, and the maximum number of single iterations is T
The reward value and number of each iteration, and the converged $Q(s, a)$ matrix.
for $n = 1 \rightarrow N$ **do**
  Get the initial state s.
  for $t = 1 \rightarrow T$ **do**
    Select the action a in the current state s based on Q using the $\epsilon - greedy$ strategy.
    Get the feedback r, $s^{'}$ from the environment.

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma max_{a'} Q(s^{'}, a^{'}) - Q(s, a)] \quad (1)$$

    $s \leftarrow s^{'}$
  end for
end for

2.2.2. *DQN* Algorithm

One drawback of *Q-Learning* is that its Q-table will become very large when the number of states in the state space is very high. *DQN* combines *Q-Learning* with a neural network, which fits the state-behavior value function of the current system. The algorithmic flow of *DQN* is Algorithm 2. *DQN* has two very important modules: experience replay and target network, which can help *DQN* achieve stable and excellent performance. Experience replay is a process of storing the data (state, action, reward and next state) in a replay buffer for each quaternions obtained and training the Q-network by sampling a random number of data from the replay buffer. Experience replay can break the correlation between samples and make the samples satisfy the independence assumption. Additionally, it improves the efficiency of the samples. The target network uses two sets of Q networks.

(1) The original training network, $Q_w(s,a)$, is used to calculate $Q_w(s,a)$ in the original loss function $\frac{1}{2}[Q_w(s,a) - (r + \gamma max_{a'} Q_{w^-}(s^{'}, a^{'})]^2$ and is updated using the normal gradient descent method.

(2) The target network, $Q_{w^-}(s,a)$, is used to calculate the $(r + \gamma max_{a'} Q_{w^-}(s^{'}, a^{'})$ term in the original loss function, $\frac{1}{2}[Q_w(s,a) - (r + \gamma max_{a'} Q_{w^-}(s^{'}, a^{'})]^2$, where w^- denotes the parameter in the target network. If the parameters of the two networks are the same, the problem of insufficient stability will occur. In order to make the update target more stable, the target network is not updated at every step. Specifically, the target network uses an older set of parameters from the training network, and the training network $Q_w(s,a)$ is updated at each step of training, while the parameters of the target network are only synchronized every C steps, i.e., $\omega^- \leftarrow \omega$. Doing so makes the target network more stable relative to the training network.

Algorithm 2 *DQN* algorithm workflow.

Initialize network $Q_w(s,a)$ with random network parameters ω.
Copy the same parameter $\omega^- \leftarrow \omega$ to initialize the target network $Q_{w'}$.
Initialize the experience replay pool R.
for $\epsilon = 1 \rightarrow E$ **do**
  Get the initial state of the environment s_1.
  for $t = 1 \rightarrow T$ **do**
    Select the action a_t according to the current network $Q_w(s,a)$ with $\epsilon - greedy$ strategy.
    Execute action a_t, get reward r_t, and the environment state changes to s_{t+1}.
    Store (s_t, a_t, r_t, s_{t+1}) into the replay pool R.
    Randomly select N samples $(s_i, a_i, r_i, s_{i+1})_{i=1,\ldots,N}$ from R.
    Compute $y_i = r_i + \gamma max_a Q_{\omega^-}(s_{i+1}, a)$ with the target network.
    Minimize the target loss $L = \frac{1}{N}\Sigma_i(y_i - Q_w(s_i, a_i))^2$, and use it to update the current network Q_ω.
    Update the target network.
  end for
end for

2.3. Confrontation Decision-Making Model

We use a multifunction radar as the target radar and a jammer with RL algorithm as the core to form a radar countermeasure model. The state transfer matrices of the target radar are the core of the decision model. In this paper, the state transfer matrices are set to empirical values. The radar operating state and threat level in the model we set are both 16. When the radar operating state is 0, the threat level is 0, indicating a shutdown state; when the radar operating state is 15, the threat level is 15, indicating that the radar is in a destroyed state.

For a cognitive radar jamming decision-making system, the reward function is an important measure of the learning process. The reward value obtained according to the

jamming function determines the decision capability of the jammer. Since the ultimate goal of the jammer is to improve the radar jamming performance. The evaluation of radar jamming effectiveness is closely related to the change in threat level. The specific calculation equation of the reward function, Equation (2), is expressed as follows. In the equation, *min* means that the radar is in the terminated state, and the reward value obtained by the jammer is 50. TL represents the change in threat level when the radar is in the non-terminating state $s_i, i \neq min$.

$$r = \begin{cases} 50, & min \\ TL, & s_i \rightarrow min \end{cases} \tag{2}$$

3. Improved Cognitive Electronic Jamming Decision-Making Method

3.1. Q-Learning Algorithm with Pheromone Mechanism

The jammer determines the difference in the threat level of the target by detecting changes in the working state of the enemy's radar before and after the implementation of jamming. The jamming decision-making system learns the best jamming strategy through the jamming effect. Both the *Q-Learning* and *DQN* algorithms have real-time learning capabilities. Among them, the *Q-Learning* algorithm, as the core basic algorithm in RL, iterates in real time by building a Q-table and finally obtains a converged Q-value table. The disadvantage of the *Q-Learning* algorithm is that the hyperparameters, especially the variation in the exploration factor, have a great impact on the convergence. Usually, the setting of hyperparameters for the algorithm is empirically dependent. *Q-Learning* algorithms are less efficient when learning for large-scale states, and during radar confrontation, longer convergence times mean higher chances of being detected and destroyed. Additionally, the *Q-Learning* algorithm explores the suboptimal states repeatedly during the iterative process due to the certain randomness of action selection and the limited update magnitude of the Q-table elements. Due to the presence of neural networks, the *DQN* algorithm can cope better with the goal of a large state size. However, the complexity of the *DQN* algorithm is much greater than that of *Q-Learning*, including the hyperparameters of the network and the learning time. At the same time, the *DQN* algorithm suffers from an overestimation problem, which has a significant negative effect on the actual jamming decision-making. We have to improve the *DQN* algorithm to solve the overestimation problem, and the complexity and hyperparameters of the improved *DQN* algorithm will be improved. These situations are not favorable for our application in practice.

In this paper, we propose the *Ant-QL* algorithm to address the shortcomings of the existing typical *Q-Learning* and *DQN* algorithms. We introduce the pheromone mechanism of the ant colony algorithm in the *Q-Learning* algorithm. As a bio-inspired algorithm, the ant colony algorithm finds a path from the starting point to the end point that meets the conditional constraints by simulating the process of ants foraging for food.

The essence of the cognitive radar jamming decision-making problem is that the jammer responds more effectively to the state transfer of the target radar. Ant colony algorithms are mostly used in problems such as optimization functions and path finding. As shown in Figure 7, the core pheromone in the ant colony algorithm can be mapped to the Q-value table in the *Q-Learning* algorithm. The Q table is initialized with $s * a$ storage cells, where s is the number of states and a is the number of actions. Figure 7a shows the structure of the Q-table. A pheromone table is generated according to the structure of the Q-table, as shown in Figure 7b. During the iteration, the agent leaves pheromones in the pheromone table. As the pheromones are continuously updated, the strategy provided by the pheromone matrix to the agent becomes closer and closer to the optimal strategy.

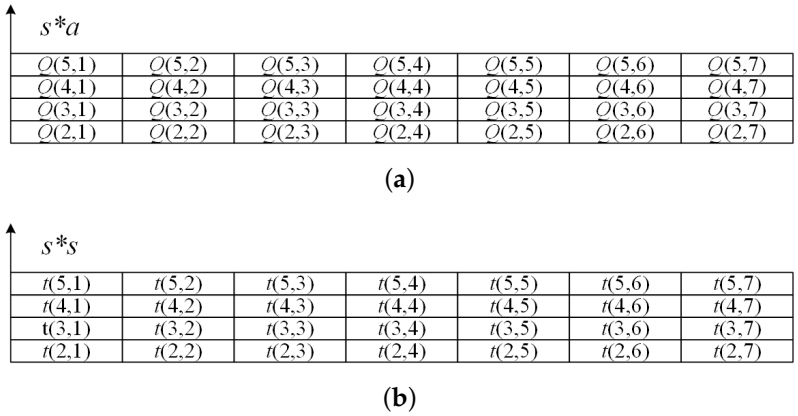


Figure 7. Structure of Q-table and pheromone table. (**a**) Structure of the Q table; (**b**) structure of the pheromone table.

The process of the *Q-Learning* algorithm based on a pheromone mechanism is show in Algorithm 3. The improved *Q-Learning* algorithm first initializes the Q matrix and all algorithm parameters. The agent first obtains the initial state from the environment. The agent performs an action to feed back to the environment. After the agent obtain the reward and the next state from the environment, the Q matrix is updated according to Equation (3), where τ_{ij} is the pheromone value of the two states before and after the transfer. After each iteration, the jammer performs a global pheromone update. After several iterations, the distribution of pheromones becomes closer to the optimal path distribution. The pheromone update process is shown in Equation (4), where $\tau_{ij}(t)$ represents the pheromone value located at position ij in the matrix at moment t, $\tau_{ij}(t+1)$ represents the pheromone value at the same position at the next moment and $\Delta\tau_{ij}(t)$ is the pheromone change value. The calculation process of $\Delta\tau_{ij}(t)$ is shown in Equation (5), and the pheromone increase is obtained using the reward value and the number of iterations obtained from each loop.

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma max_{a'} Q(s^{'}, a^{'}) - Q(s,a) + \tau_{ij}] \tag{3}$$

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \Delta\tau_{ij}(t) \tag{4}$$

$$\Delta\tau_{ij}(t) = \frac{rewards}{counts} \tag{5}$$

Algorithm 3 Improved *Q-Learning* algorithm.

1: Initialize $Q(s,a)$, the number of iterations is N, the learning rate is α, the discount factor is γ, the exploration factor ϵ, and the maximum number of single iterations is T
2: Initialize the pheromone matrix.
3: The reward value and number of each iteration, and the converged $Q(s,a)$ matrix.
4: **for** $n = 1 \rightarrow N$ **do**
5: Get the initial state s.
6: **for** $t = 1 \rightarrow T$ **do**
7: Select the action a in the current state s based on Q and pheromone matrix using the $\epsilon - greedy$ strategy.
8: Get the feedback r, $s^{'}$ from the environment, and then update Q according to Equation (3).
9: **end for**
10: Update the pheromone matrix.
11: **end for**

3.2. Ant Colony Q-Learning Algorithm

In practical counter scenarios, it is difficult for a single jammer to have multi-antenna, wide range and high power jamming capabilities due to size and power limitations. The jammer is usually in a single orientation and frequency band during the countermeasure against the target radar, so it is difficult for it to perform jamming in multiple directions and in multiple frequency bands simultaneously. Additionally, the decision algorithm process for a single jammer means that no matter how it is optimized, a single jammer has a high chance of being detected and destroyed by the target radar compared with cooperative multi-machine jamming decision-making. Compared with traditional methods, the improved *Q-Learning* algorithm can converge to the optimal point faster and smoother and can effectively avoid becoming trapped in a local optimum. We can consider the *Q-Learning* algorithm with a pheromone mechanism as a single optimality-seeking intelligence (ant); then, the algorithm, when extended to a cluster, can be considered as a multi-intelligence (ant colony) algorithm. Multiple agents perform separate adversarial and decision-making processes in multiple working directions of the target radar; then, the radar jamming decision-making efficiency of the multi-agent remains relatively low. In this paper, we propose a multi-agent collaborative decision-making method based on information sharing. The method becomes simple in terms of hardware and power requirements and only requires the jammer to be able to cooperate in a single direction to achieve high-power suppression jamming. The target radar is usually considered to be able to perform a simultaneous operation in multiple dimensions and multiple frequency bands, but its state transition matrix is all consistent in the face of jamming. This is the key to the ability of multiple agents to cause effective cooperative jamming in decision-making.

The flow of the cooperative jamming decision-making algorithm for multiple jammers is shown in Figure 8. The initialization process includes the parameters of the *Q-Learning* algorithm, the pheromone matrices and the number of agents. Here, the agents represent the jammers, and the number of agents is also the number of jammers. After all jammers complete one round of iterations, the increments in Q matrices (pheromone matrices) of some jammers with better effects are selected as shared information. The Q-matrices (pheromone matrices) of the less effective jammers are zeroed out and set as shared information. There are two differences in multi-machine jamming decision-making compared with single-machine jamming decision-making. The first is that the pheromone increase is calculated independently for each jammer, as shown in Equations (6) and (7). The evaporation mechanism of a pheromone needs to be considered in multi-machine jamming decision-making, and ρ in Equation (6) is the volatilization factor. After several iterations, the algorithm can reach convergence faster and more stably. It should be noted that multiple jammers have different exploration steps at each round of iterations. During the simulation experiments, the Q matrix and the pheromone matrix stop updating after the jammer with a small number of exploration steps reaches the termination state. The jammer uses the existing Q matrix and pheromone matrix as a guide to interact with the environment. After all the jammers reach the termination state in this round, the algorithm selects the optimal agent in this round.

$$\tau_{ij}^{k}(t+1) = (1-\rho)\tau_{ij}^{k}(t) + \Delta\tau_{ij}^{k}(t) \tag{6}$$

$$\Delta\tau_{ij}^{k}(t) = \frac{rewards^{k}}{counts^{k}} \tag{7}$$

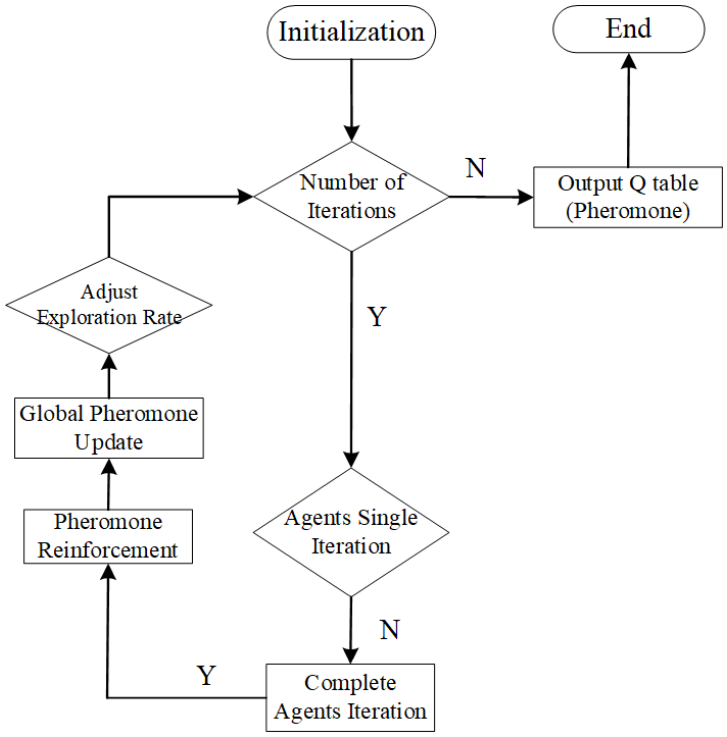


Figure 8. *Ant-QL* algorithm flow chart.

4. Simulation Experiments and Results Analysis

There will be various states of the radar during operation, such as search, tracking, identification, guidance, etc. There are also various jamming actions that can be performed by the jammer, such as noise FM jamming, false target jamming and distance dragging jamming. The state of a radar operation is denoted as $si, i = 0, 1, \ldots, 15$. The state of a jamming operation is denoted as $ai, i = 0, 1, \ldots, 9$. States $s1$, $s2$ and $s3$ represent the three search states. States $s4$, $s5$, $s6$ and $s7$ represent the four tracking states. States $s8$, $s9$, $s10$ and $s11$ represent the four identification states. States $s12$, $s13$ and $s14$ represent the three guidance states. For the convenience of modeling, state $s0$ is set as the shutdown state and state $s15$ is set as the destroyed state in this paper. The jamming mode $a0$ represents noise amplitude modulation jamming. The jamming mode $a1$ represents noise FM jamming. The jamming mode $a2$ represents noise modulation jamming. The jamming mode $a3$ represents dexterous noise jamming. The jamming mode $a4$ represents dense dummy target suppression jamming. The jamming mode $a5$ represents distance deception jamming. The jamming mode $a6$ represents velocity deception jamming. The jamming mode $a7$ represents angle deception jamming. The jamming mode $a8$ represents distance–velocity deception jamming. The jamming mode $a9$ represents distance–angle deception jamming. The state transfer matrix is the working core of the target radar. Similar to Refs. [42,43], the state transfer matrices in this paper are set to empirical values. Additionally, the state of the radar and the threat level are positively correlated. During the simulation, the state transfer matrices are set with small fluctuations. When the target radar is subjected to some kind of suppression jamming, the table of transfer probability of radar operating state is shown in Table 1. When the target radar is subjected to false target jamming, the state transfer matrix is shown in Table 2. The state transfer matrices of the target radar when subjected to other jamming are not listed in this paper.

Table 1. State transfer matrix of the radar under some suppression jamming.

	s0	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12	s13	s14	s15
s0	0.18	0.28	0.25	0.19	0.1	0	0	0	0	0	0	0	0	0	0	0
s1	0.12	0.28	0.25	0.15	0.1	0.1	0	0	0	0	0	0	0	0	0	0
s2	0.1	0.2	0.15	0.25	0.2	0.1	0	0	0	0	0	0	0	0	0	0
s3	0.1	0.2	0.15	0.25	0.2	0.1	0	0	0	0	0	0	0	0	0	0
s4	0	0	0	0.05	0.2	0.25	0.3	0.2	0	0	0	0	0	0	0	0
s5	0	0	0	0.05	0.15	0.2	0.35	0.25	0	0	0	0	0	0	0	0
s6	0	0	0	0.03	0.1	0.15	0.2	0.48	0	0	0	0	0	0	0	0
s7	0	0	0	0	0.02	0.15	0.2	0.1	0.33	0.2	0	0	0	0	0	0
s8	0	0	0	0	0	0	0.02	0.08	0.15	0	0.3	0.3	0.15	0	0	0
s9	0	0	0	0	0	0	0	0.05	0.15	0	0	0.3	0.3	0.2	0	0
s10	0	0	0	0	0	0	0	0	0.2	0	0	0.3	0.25	0.25	0	0
s11	0	0	0	0	0	0	0	0	0.15	0	0.2	0.2	0.25	0.25	0.05	0
s12	0	0	0	0	0	0	0	0	0	0	0	0.2	0.15	0.3	0.2	0.15
s13	0	0	0	0	0	0	0	0	0	0	0	0.1	0.15	0.25	0.2	0.3
s14	0	0	0	0	0	0	0	0	0	0	0	0	0.1	0.2	0.2	0.5
s15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

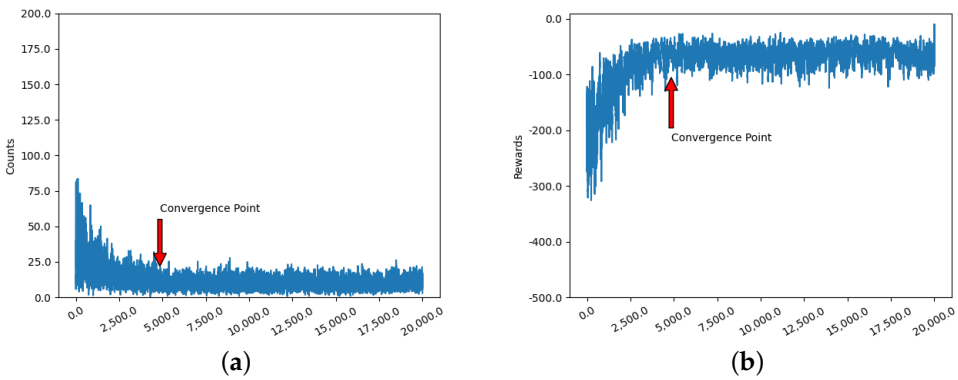
Table 2. State transfer matrix of the radar under false-target jamming.

	s0	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12	s13	s14	s15
s0	0.3	0.4	0.2	0.1	0	0	0	0	0	0	0	0	0	0	0	0
s1	0.1	0.1	0.3	0.2	0.2	0.1	0	0	0	0	0	0	0	0	0	0
s2	0.1	0.1	0.1	0.3	0.2	0.2	0	0	0	0	0	0	0	0	0	0
s3	0	0.1	0.1	0.2	0.4	0.2	0	0	0	0	0	0	0	0	0	0
s4	0	0	0	0.2	0.2	0.1	0.4	0.1	0	0	0	0	0	0	0	0
s5	0	0	0	0	0.1	0.2	0.3	0.4	0	0	0	0	0	0	0	0
s6	0	0	0	0	0.1	0.1	0.2	0.4	0.2	0	0	0	0	0	0	0
s7	0	0	0	0	0	0.2	0.2	0.2	0.2	0.2	0	0	0	0	0	0
s8	0	0	0	0	0	0	0	0	0.1	0.2	0.3	0.3	0.1	0	0	0
s9	0	0	0	0	0	0	0	0	0.1	0.2	0.3	0.3	0.1	0.2	0	0
s10	0	0	0	0	0	0	0	0	0	0.1	0.2	0.3	0.2	0.2	0	0
s11	0	0	0	0	0	0.05	0	0	0.15	0	0.1	0.2	0.2	0.25	0.05	0
s12	0	0	0	0	0	0	0.1	0	0	0.2	0	0.1	0	0.3	0.15	0.15
s13	0	0	0	0.2	0	0	0.15	0	0	0.15	0	0	0	0.15	0.2	0.15
s14	0	0	0	0	0	0	0.15	0	0	0.25	0	0	0.1	0.2	0.2	0.1
s15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

The parameters of the jammer with the *Q-Learning* algorithm are set as follows: the learning rate α is 0.05, the discount factor γ is 0.7 and the initial value of the exploration factor ϵ is 0.99 and decreases with the number of simulation steps at a decay rate of 0.0003. The parameters for the *DQN* algorithm are set as follows: the learning rate α is 0.01, the discount factor γ is 0.9, the initial value of exploration factor ϵ is 0.99,and the decreasing decay rate is 0.0003 with the number of simulation steps. The network in the *DQN* is a three-layer fully connected network, and the capacity of the replay buffer is set to 2000. The improved *Q-Learning* algorithm adds the pheromone-related parameters. The number of jammers for cluster jamming decision-making is assumed to be 3, i.e., three jammers collaborate with each other for jamming decision-making. All simulations are performed for 20,000 iterations. In this paper, the condition to end each round of iterations is that the radar state is transferred to the termination state or the number of explorations reaches 200. In the simulation experiments, we use the number of state transfers and the reward value in a single iteration as the quantitative evaluation criteria. It can be seen from the simulation results of multiple times that the convergence processes of both are completely synchronized and can confirm the convergence process with each other. The criterion for determining convergence in this paper is to divide every 10 iterations into groups. The average of the cumulative reward value and the number of state transfers is calculated for each group. The mean values between two adjacent groups are compared separately. We determine that the agent has reached convergence when the mean value of reward values in each group is less than -120 and the difference between the means of two adjacent groups is not greater than 15.

According to the convergence assessment criterion in this paper, we obtained the following results through simulation experiments. The simulation results of the *Q-Learning* algorithm are shown in Figure 9. As can be seen in Figure 9a,b, they converge at the 4800th iteration. Figure 10 shows the simulation results of the *DQN* algorithm, which converges at the 3600th iteration. Figure 11 shows the simulation results of the improved *Q-Learning* algorithm, which converges at the 2500th iteration. Figure 12 shows the simulation results of the *Ant-QL* algorithm, which converges at the 700th iteration. The convergence results of the four different algorithms are shown in Table 3.

After the simulation, it can be seen that the *DQN* algorithm accelerates the convergence speed and stability compared with the *Q-Learning* algorithm. However, the *DQN* algorithm suffers from the computationally time-consuming feature, which makes it difficult to achieve more desirable results. Comparing *Q-Learning* and *DQN*, the convergence speed of the improved *Q-Learning* algorithm is improved by 48% and 31%, respectively. The convergence speed of the *Ant-QL* algorithm is improved by 85.4%, 80.56% and 72% compared with the convergence speed of the remaining three algorithms. In addition, we can also learn from the simulation results of these algorithms that the overall stability of the *Ant-QL* algorithm is very high and the difference between groups is very small.



(**a**) (**b**)

Figure 9. Convergence process of *Q-Learning* algorithm. (**a**) Convergence process of counts; (**b**) convergence process of rewards.

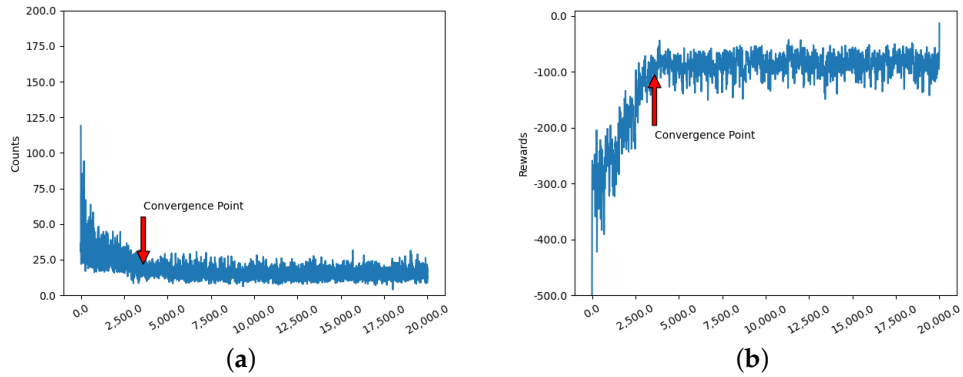


Figure 10. Convergence process of *DQN* algorithm. (**a**) Convergence process of counts; (**b**) convergence process of rewards.

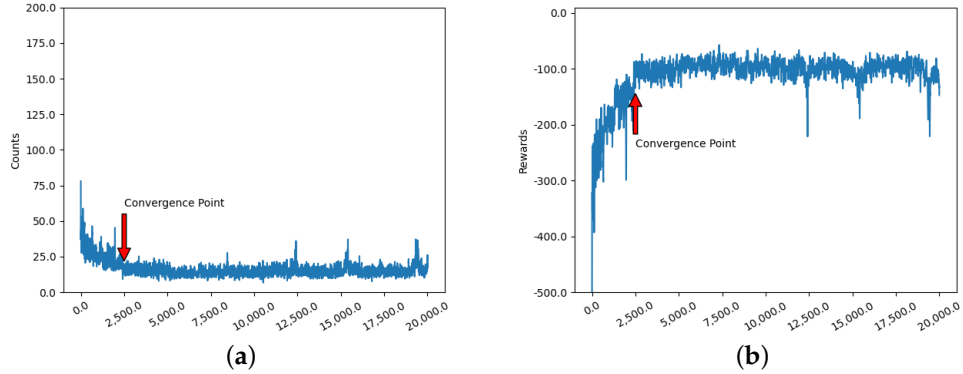


Figure 11. Convergence process of improved *Q-Learning*. (**a**) Convergence process of counts; (**b**) convergence process of rewards.

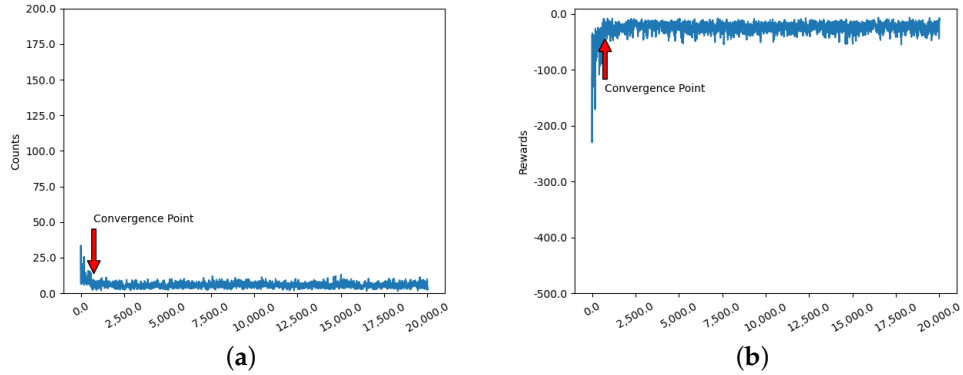


Figure 12. Convergence process of *Ant-QL*. (**a**) Convergence process of counts; (**b**) convergence process of rewards.

Table 3. Simulation results of algorithm convergence.

	Q-Learning	*DQN*	Improve-*QL*	*Ant-QL*
Counts	4800	3600	2500	700
Rewards	4800	3600	2500	700

The performance of each algorithm depends not only on the number of rounds of convergence but also on its average response time. After the algorithms converged, we

tested each algorithm 50 times and obtained their average response times. Table 4 shows the average response times of the four algorithms after convergence. The *DQN* algorithm has the longest average response time due to the time-consuming nature of training the network. Compared with the *Q-Learning* algorithm, the improved *Q-Learning* algorithm reduced the average response time by 68.38%. The *Ant-QL* algorithm has the fastest average response time, which proves that the algorithm is more adaptable in electronic countermeasures.

Table 4. Simulation results of algorithm response time.

	Q-Learning	***DQN***	**Improve-*QL***	***Ant-QL***
Response Time (s)	6.99×10^{-4}	2.234×10^{-3}	2.21×10^{-4}	1.7×10^{-4}

The simulation results prove the superiority of the *Ant-QL* algorithm. Compared with *Q-Learning*, *Ant-QL* makes the iterative process of the jammer faster and more stable by introducing pheromones while avoiding the problem of a local optimum that can occur in the *Q-Learning* algorithm. Compared with *DQN*, *Ant-QL* does not overestimate during the iterative process, and the jammer has the path guidance provided by the pheromone as the number of iterations increases. *Ant-QL* is also considerably less complex than the *DQN* algorithm when it comes to algorithm debugging and application. After the algorithms converge, the average response time of the algorithms has a large impact on the performance of the jammer. The average response time of the improved *Q-Learning* algorithm is faster than that of the *Q-Learning* algorithm due to the advantage of pheromones in the merit-seeking process. The average response time of the *Ant-QL* algorithm is faster than that of the improved *Q-Learning* algorithm due to the addition of multi-machine collaboration. These advantages not only reduce the hardware requirements of the jammer but also increase its advantages in a real battlefield.

5. Conclusions

In this paper, we proposed an improved method combining the pheromone mechanism and the *Q-Learning* algorithm and applied it to radar jamming decision-making. The method not only performs Q-table updating and learning during the iterative process but also continuously optimizes the search range of the jammer to find the optimal state transfer and purposefully improves the convergence speed and stability. Through the simulation results, it can be analyzed that the improved *Q-Learning* algorithm can avoid the local optimum and reach convergence faster during the iterative process. The complexity and training time of the algorithm have significant advantages over *DQN*, which can effectively reduce the probability of being detected and destroyed in the actual battlefield. To better adapt to the trend of cluster confrontation in future warfare and to effectively overcome the hardware and power limitations of a single jammer, we extend the improved *Q-Learning* algorithm with multiple machines to cope with these trends. Under the condition of information sharing, the method introduces the idea of an ant colony and shares the information by selecting the better jammer during each iteration. The *Ant-QL* algorithm for clustering reduces the hardware and power requirements of the jammer. The simulation results show very good convergence and stability, which will help to improve the survivability and adaptability of these jammers. Additionally, after the convergence of each algorithm, their average response times are discussed in this paper. The simulation results show that the improved *Q-Learning* algorithm and *Ant-QL* algorithm have shorter average response times compared with traditional methods.

CEW is receiving more and more attention. For cognitive radar jamming decision-making, there is also a continuous need to explore better algorithms. RL and metaheuristic algorithms are still a hot topic for many researchers, and new and various algorithms are emerging. This paper concludes with a few possible research points that will hopefully be useful to future researchers.

(1) Due to the increasing complexity of the battlefield situation, multi-UAV cluster reconnaissance positioning has gained tremendous attention and development [50,51]. The advantages of UAV clusters in reconnaissance and positioning are enormous, and there are still many technical difficulties waiting to be broken through.
(2) The research method in this paper has achieved better results, but there are still some problems. To address these problems, we also propose some new research plans. For example, if we increase the anti-jamming capability of the radar, the cooperative decision-making method of multiple jammers can be improved, for example, how to communicate efficiently between multiple jammers. It should also be noted that the *Ant-QL* method proposed in this paper discards information matrices that do not work well in each round. In the next stage of research, we need to consider how to utilize all information matrices of multiple jammers more effectively.
(3) For radar jamming decision-making, the jammers use hierarchical RL algorithms for radar jamming decisions in response to the hierarchical structure of the target radar. For increasingly urgent cluster confrontation, multi-jammers use multi-intelligence RL algorithms for radar jamming decision-making. There are limitations in using a single algorithm to solve this problem, and we hope that more researchers will combine algorithms such as metaheuristics with RL to solve this problem more effectively.
(4) Research on various new jamming methods is the key to gaining battlefield advantages. Radars of different regimes have many counterpart anti-jamming measures to traditional jamming methods. We note that the use of many new types of jamming usually achieves unexpected results [52–55].
(5) With the development of a cognitive radar, radar jamming decisions based on fixed jamming methods still pose a significant battlefield threat. A future jammer should be centered on radar jamming decision-making and have an adaptive jamming waveform optimization capability. These features will greatly enhance the flexibility and adaptability of jammers in the actual battlefield. Numerous meta-heuristic optimization algorithms are currently available for jamming waveform optimization [56,57].
(6) As battlefield complexity rises, electronic countermeasures in various clusters will proliferate. The jammer itself is limited by the resources of the applicable platform and often does not have the software, hardware and power resources to match the target radar. For the jamming side, the rational allocation of jamming resources to maximize operational effectiveness is the difficulty. Despite jamming waveform optimization, the jammer of the cluster needs to have a strong jamming resource scheduling algorithm. The algorithms currently applied to jamming resource scheduling include the Hungarian algorithm, the dynamic planning algorithm, etc. [58,59]. We believe that the fusion algorithm of a metaheuristic algorithm and RL is an effective method to realize cognitive electronic countermeasures in the future.

Author Contributions: Conceptualization, C.Z. and S.X.; methodology, C.Z.; software, C.Z. and Y.S.; validation, C.Z., Y.S. and R.J.; formal analysis, C.Z.; investigation, C.Z.; resources, J.H.; data curation, C.Z.; writing—original draft preparation, C.Z.; writing—review and editing, C.Z., Y.S. and R.J.; visualization, C.Z.; supervision, S.X. and J.H.; project administration, S.X. and J.H.; funding acquisition, S.X. and J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This paper is partly supported by the National Key R&D Program of China under grant 2021YFA0716600, the Shenzhen Fundamental Research Program under grant No. JCYJ20180307151430655 and the Shenzhen Science and Technology Program under grant No. KQTD20190929172704911.

Data Availability Statement: The data are available to readers by contacting the corresponding author.

Acknowledgments: The authors thank the Software Defined Radar Lab of Sun Yat-sen University.

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

Abbreviations

The following abbreviations are used in this manuscript:

EW	Electronic Warfare
AI	Artificial Intelligence
RL	Reinforcement Learning
CEW	Cognitive Electronic Warfare
IDSS	Intelligent Decision-Making Support System
OODA	Observe, Orient, Decide and Act
Ant-QL	Ant Colony Q-Learning

References

1. Haigh, K.; Andrusenko, J. *Cognitive Electronic Warfare: An Artificial Intelligence Approach*; Artech House: London, UK, 2021.
2. Haykin, S. New generation of radar systems enabled with cognition. In Proceedings of the 2010 IEEE Radar Conference, Arlington, VA, USA, 10–14 May 2010.
3. Haykin, S. Cognitive radar: A way of the future. *IEEE Signal Process. Mag.* **2006**, *23*, 30–40. [CrossRef]
4. Darpa, A. *Behavioral Learning for Adaptive Electronic Warfare*; Darpa-BAA-10-79; Defense Advanced Research Projects Agency: Arlington, TX, USA, 2010.
5. Haystead, J. DARPA seeks proposals for adaptive radar countermeasures J. *J. Electron. Def.* **2012**, *2012*, 16–18.
6. Gurbuz, S.Z.; Griffiths, H.D.; Charlish, A.; Rangaswamy, M.; Greco, M.S.; Bell, K. An overview of cognitive radar: Past, present, and future. *IEEE Aerosp. Electron. Syst. Mag.* **2019**, *34*, 6–18. [CrossRef]
7. Zhou, H. An introduction of cognitive electronic warfare system. In *Communications, Signal Processing, and Systems, Proceedings of the 2018 CSPS, Dalian, China, 14–16 July 2018*; Volume III: Systems 7th; Springer: Singapore, 2020; pp. 1202–1210.
8. du Plessis, W.P.; Osner, N.R. Cognitive electronic warfare (EW) systems as a training aid. In Proceedings of the Electronic Warfare International Conference (EWCI), Bangalore, India, 13–16 February 2018; pp. 1–7.
9. Shafei, W.; Yanfei, B.; Yan, L. Cognitive Electronic Warfare Architecture and Technology. *China Sci. Inf. Sci.* **2018**, *48*, 1603–1613.
10. Kun, Z.B.; Weizang, Z.; Wei, L.; Ying, Y.; Tianhao, G. A Review of Reinforcement Learning Based Radar Jamming Decision Making Techniques. *Electro-Opt. Control.* **2022**, *29*, 52.
11. Songtao, L.; Chenshuo, L.; Yang, G.; Zhenming, W. Review of electronic countermeasure jamming effect evaluation techniques. *J. Chin. Acad. Electron. Sci.* **2020**, *15*, 306–317.
12. Gong, H. New field of electronic warfare-AI. *Aerosp. Shanghai* **1986**, *2*, 36–42.
13. Li, Z. Application of AI technology in EW. *Electron. Warf. Technol.* **1988**, *2*, 27–39.
14. Wang, X.; Zhu, M.; Cheng, Y. *The Principle of Reinforcement Learning and Its Application*; Science Press: Beijing, China, 2014.
15. Thrun, S.; Littman, M.L. Reinforcement learning: An introduction. *AI Mag.* **2000**, *21*, 103.
16. Yang, Z.; Guangya, S.; Yanzheng, W.; Rongxiang, L. Modeling and simulation of cognitive electronic attack operations under system countermeasures. *J. Chin. Acad. Electron. Sci.* **2019**, *5*, 10–14.
17. Wonderley, D.; Selee, T.; Chakravarthy, V. Game theoretic decision support framework for electronic warfare applications. In Proceedings of the 2016 IEEE Radar Conference (RadarConf), Philadelphia, PA, USA, 2–6 May 2016; pp. 1–5.
18. Gao, Y.; Xiao, Y.; Wu, M.; Xiao, M.; Shao, J. Game theory-based anti-jamming strategies for frequency hopping wireless communications. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 5314–5326. [CrossRef]
19. Hongwei, S.; Ningning, T.; Fujun, S. Electronic interference pattern selection based on DS evidence theory. *J. Ballist. Arrow Guid.* **2003** , 218–220.
20. Gong, L.W.; Hua, W.C. Research on self-learning model of intelligent radar jamming system. *Mod. Def. Technol.* **2009** , *1*, 83–86.
21. Ye, F.; Che, F.; Gao, L. Multiobjective cognitive cooperative jamming decision-making method based on Tabu search-artificial bee colony algorithm. *Int. J. Aerosp. Eng.* **2018**, *2018*, 7490895. [CrossRef]
22. Ye, F.; Che, F.; Tian, H. Cognitive cooperative-jamming decision method based on bee colony algorithm. In Proceedings of the 2017 Progress in Electromagnetics Research Symposium-Fall (PIERS-FALL), Singapore, 19–22 November 2017; pp. 531–537.
23. Pan, W.; Jin, X.; Xie, H.; Xia, Y. Radar jamming strategy allocation algorithm based on improved chaos genetic algorithm. In Proceedings of the 2020 Chinese Control And Decision Conference (CCDC), Hefei, China, 22–24 August 2020; pp. 4478–4483.
24. Tan, Y.; Pan, W.; Han, Y.; Xu, S. Research on force assignment of radar jamming system based on chaos genetic algorithm. In Proceedings of the 2019 Chinese Control And Decision Conference (CCDC), Nanchang, China, 3–5 June 2019; pp. 1193–1197.
25. Gu, W.; Zhu, L.; Bu, Y.; Yue, W.; Cai, X.; Fan, Y. Collaborative jamming decision-making mechanism using ant colony algorithm in electromagnetic antagonism. In Proceedings of the 2020 IEEE 20th International Conference on Communication Technology (ICCT), Nanning, China, 28–31 October 2020; pp. 1645–1649.
26. Liu, W.L.; Gong, Y.J.; Chen, W.N.; Liu, Z.; Wang, H.; Zhang, J. Coordinated charging scheduling of electric vehicles: A mixed-variable differential evolution approach. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 5094–5109. [CrossRef]

27. Zhou, S.; Xing, L.; Zheng, X.; Du, N.; Wang, L.; Zhang, Q. A self-adaptive differential evolution algorithm for scheduling a single batch-processing machine with arbitrary job sizes and release times. *IEEE Trans. Cybern.* **2019**, *51*, 1430–1442. [CrossRef] [PubMed]
28. Zhao, F.; Zhao, L.; Wang, L.; Song, H. An ensemble discrete differential evolution for the distributed blocking flowshop scheduling with minimizing makespan criterion. *Expert Syst. Appl.* **2020**, *160*, 113678. [CrossRef]
29. Zhao, F.; Zhang, L.; Zhang, Y.; Ma, W.; Zhang, C.; Song, H. A hybrid discrete water wave optimization algorithm for the no-idle flowshop scheduling problem with total tardiness criterion. *Expert Syst. Appl.* **2020**, *146*, 113166. [CrossRef]
30. Safatly, L.; Bkassiny, M.; Al-Husseini, M.; El-Hajj, A. Cognitive radio transceivers: RF, spectrum sensing, and learning algorithms review. *Int. J. Antennas Propag.* **2014**, *2014*, 548473. [CrossRef]
31. Akanksha, E.; Sharma, N.; Gulati, K. Review on reinforcement learning, research evolution and scope of application. In Proceedings of the 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 8–10 April 2021; pp. 1416–1423.
32. Li, H.; Li, Y.; He, C.; Zhan, J.; Zhang, H. Cognitive electronic jamming decision-making method based on improved Q-learning algorithm. *Int. J. Aerosp. Eng.* **2021**, *2021*, 8647386. [CrossRef]
33. Sadhu, A.K.; Konar, A. An efficient computing of correlated equilibrium for cooperative Q-learning-based multi-robot planning. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *50*, 2779–2794.
34. Kontoudis, G.P.; Vamvoudakis, K.G. Kinodynamic motion planning with continuous-time Q-learning: An online, model-free, and safe navigation framework. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3803–3817. [CrossRef]
35. Li, J.; Chai, T.; Lewis, F.L.; Ding, Z.; Jiang, Y. Off-policy interleaved Q-learning: Optimal control for affine nonlinear discrete-time systems. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *30*, 1308–1320. [CrossRef] [PubMed]
36. Peng, Y.; Chen, Q.; Sun, W. Reinforcement Q-learning algorithm for H_∞ tracking control of unknown discrete-time linear systems. *IEEE Trans. Syst. Man, Cybern. Syst.* **2019**, *50*, 4109–4122. [CrossRef]
37. Wang, D.; Zhang, W.; He, H.; Tian, Y.C. Efficient hybrid central processing unit/input–output resource scheduling for virtual machines. *IEEE Trans. Ind. Electron.* **2020**, *68*, 2714–2724. [CrossRef]
38. Tang, F.; Zhou, Y.; Kato, N. Deep reinforcement learning for dynamic uplink/downlink resource allocation in high mobility 5G HetNet. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 2773–2782. [CrossRef]
39. Qiang, X.; Wei-gang, Z.; Xin, J. Intelligent countermeasure design of radar working-modes unknown. In Proceedings of the 2017 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Xiamen, China, 22–25 October 2017; pp. 1–5.
40. Qiang, X.; Weigang, Z.; Xin, J. Research on method of intelligent radar confrontation based on reinforcement learning. In Proceedings of the 2017 2nd IEEE International Conference on Computational Intelligence and Applications (ICCIA), Beijing, China, 8–11 September 2017; pp. 471–475.
41. Kang, L.; Bo, J.; Hongwei, L.; Siyuan, L. Reinforcement learning based anti-jamming frequency hopping strategies design for cognitive radar. In Proceedings of the 2018 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Qingdao, China, 14–16 September 2018; pp. 1–5.
42. Kun, Z.B.; Weizang, Z.; Wei, L.; Ying, Y.; Tianhao, G. Markov-based multifunctional radar cognitive jamming decision modeling study. *Syst. Eng. Electron. Technol.* **2022**, *44*, 2488–2497.
43. Kun, Z.B.; Weizang, Z.; Wei, L.; Ying, Y.; Tianhao, G. A priori knowledge-based intelligent jamming decision method for multifunctional radar. *Syst. Eng. Electron. Technol.* **2022**, *44*, 3685–3695.
44. Yunjie, L.; Yunpeng, Z.; Meiguo, G. Q-learning algorithm-based design of cognitive radar countermeasure process. *J. Beijing Univ. Technol.* **2015**, *35*, 1194–1199.
45. Kai, Z.B.; Weizang, Z. DQN cognitive interference decision-making approach for multifunctional radars. *Syst. Eng. Electron. Technol.* **2020**, *42*, 819–825.
46. Zhang, B.; Zhu, W. A cognitive jamming decision method for multi-functional radar based on Q-learning. *Telecommun. Eng.* **2020**, *60*, 129–136.
47. Visnevski, N.; Krishnamurthy, V.; Haykin, S.; Currie, B.; Dilkes, F.; Lavoie, P. Multi-function radar emitter modelling: A stochastic discrete event system approach. In Proceedings of the 42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475), Maui, HI, USA, 9–12 December 2003; Volume 6, pp. 6295–6300.
48. Visnevski, N.; Krishnamurthy, V.; Wang, A.; Haykin, S. Syntactic modeling and signal processing of multifunction radars: A stochastic context-free grammar approach. *Proc. IEEE* **2007**, *95*, 1000–1025. [CrossRef]
49. Zou, W.; Niu, C.; Liu, W.; Gao, O.; Zhang, H. A3C-based multifunctional radar cognitive jamming decision method. *Syst. Eng. Electron. Technol.* **2023**, *45*, 86–92.
50. Zhu, L.; Zhang, J.; Xiao, Z.; Xia, X.G.; Zhang, R. Multi-UAV aided millimeter-wave networks: Positioning, clustering, and beamforming. *IEEE Trans. Wirel. Commun.* **2021**, *21*, 4637–4653. [CrossRef]
51. Wang, R.; Gu, Y.; Zhou, Z.; Wang, Z.; Xu, F.; Luo, J.; Ma, L.; Qiu, H. A Dynamic Task Assignment Strategy for Emitter Reconnaissance and Positioning through Use of UAV Swarms. In Proceedings of the 11th International Conference on Computer Engineering and Networks, Hechi, China, 21–25 October 2021; Springer: Singapore, 2022; pp. 1654–1663.
52. Han, B.; Qu, X.; Yang, X.; Li, W.; Zhang, Z. DRFM-Based Repeater Jamming Reconstruction and Cancellation Method with Accurate Edge Detection. *Remote Sens.* **2023**, *15*, 1759. [CrossRef]

53. Wang, X.; Liu, J.; Zhang, W.; Fu, Q.; Liu, Z.; Xie, X. Mathematic principles of interrupted-sampling repeater jamming (ISRJ). *Sci. China Ser. F Inf. Sci.* **2007**, *50*, 113–123. [CrossRef]
54. Sun, Z.; Quan, Y.; Liu, Z. A Non-Uniform Interrupted-Sampling Repeater Jamming Method for Intra-Pulse Frequency Agile Radar. *Remote Sens.* **2023**, *15*, 1851. [CrossRef]
55. Aldimashki, O.; Serbes, A. Performance of chirp parameter estimation in the fractional Fourier domains and an algorithm for fast chirp-rate estimation. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 3685–3700. [CrossRef]
56. Wang, Y.; Zhang, T.; Kong, L.; Ma, Z. A stochastic simulation optimization based range gate pull-off jamming method. *IEEE Trans. Evol. Comput.* **2022**, *27* , 580–594. [CrossRef]
57. Jia, R.; Zhang, T.; Wang, Y.; Deng, Y.; Kong, L. An intelligent range gate pull-off (RGPO) jamming method. In Proceedings of the 2020 International Conference on UK-China Emerging Technologies (UCET), Glasgow, UK, 20–21 August 2020; pp. 1–4.
58. Li, K.; Jiu, B.; Wang, P.; Liu, H.; Shi, Y. Radar active antagonism through deep reinforcement learning: A way to address the challenge of mainlobe jamming. *Signal Process.* **2021**, *186*, 108130. [CrossRef]
59. Geng, J.; Jiu, B.; Li, K.; Zhao, Y.; Liu, H.; Li, H. Radar and Jammer Intelligent Game under Jamming Power Dynamic Allocation. *Remote Sens.* **2023**, *15*, 581. [CrossRef]