electronics

MDPI

Article

Intelligent Frequency Decision Communication with Two-Agent Deep Reinforcement Learning

Xin Liu, Mengqi Shi * and Mei Wang

School of Information Science and Engineering, Guilin University of Technology, Guilin 541000, China
* Correspondence: 1020210999@glut.edu.cn

Abstract: Traditional intelligent frequency-hopping anti-jamming technologies typically assume the presence of an ideal control channel. However, achieving this ideal condition in real-world confrontational environments, where the control channel can also be jammed, proves to be challenging. Regrettably, in the absence of a reliable control channel, the autonomous synchronization of frequency decisions becomes a formidable task, primarily due to the dynamic and heterogeneous nature of the transmitter and receiver's spectral states. To address this issue, a novel communication framework for intelligent frequency decision is introduced, which operates without the need for negotiations. Furthermore, the frequency decision challenge between two communication terminals is formulated as a stochastic game, with each terminal's utility designed to meet the requirements of a potential game. Subsequently, a two-agent deep reinforcement learning algorithm for best-response policy learning is devised to enable both terminals to achieve synchronization while avoiding jamming signals. Simulation results demonstrate that once the proposed algorithm converges, both communication terminals can effectively evade jamming signals. In comparison to existing similar algorithms, the throughput performance of this approach remains largely unaffected, with only a slightly extended convergence time. Notably, this performance is achieved without the need for negotiations, making the presented algorithm better suited for realistic scenarios.

Keywords: stochastic game; ordinal potential game; multi-agent reinforcement learning; intelligent frequency decision; no control channel; deep reinforcement learning

Citation: Liu, X.; Shi, M.; Wang, M. Intelligent Frequency Decision Communication with Two-Agent Deep Reinforcement Learning. *Electronics* **2023**, *12*, 4529. https://doi.org/10.3390/electronics12214529

Academic Editor: Franco Cicirelli

Received: 11 October 2023
Revised: 30 October 2023
Accepted: 1 November 2023
Published: 3 November 2023

1. Introduction

In wireless communication networks, the wireless channel serves as the transmission path between the transmitter and receiver. It is profound to focus on their properties in real-life situations despite the lack of tangible connections. However, wireless channels are vulnerable to malicious jamming attacks [1], posing a nonnegligible challenge to reliable communication. To address these challenges, spread spectrum technology [2], including direct-sequence spread spectrum (DSSS), frequency-hopping (FH), and time-hopping (TH), has emerged as an effective anti-jamming measure. The control competition over the electromagnetic spectrum between communication terminals requires a decision-making scheme that can assist or predict such struggles. Game theory [3] has become a valuable mathematical tool to tackle anti-jamming issues. It enables the system to select the optimal policies in confrontational or conflict situations. However, with the increasing complexity and variability of the jamming environment in wireless communication channels, coupled with advancements in artificial intelligence software and hardware, jamming technology has become more intelligent and dynamic. Consequently, traditional anti-jamming techniques struggle to effectively combat these challenges. Integrating machine learning with anti-jamming technology offers a promising solution. By developing intelligent anti-jamming systems, we can cater to the specific demands of real-world scenarios. The real-world scenario of this paper, that is, in the actual confrontation scenario, the reliability and security of the communication are guaranteed, and the control channel is not affected

by jamming. Therefore, the development of efficient and flexible intelligent anti-jamming schemes has become a significant challenge [4,5].

The adversarial dynamics between jammers and communicators can be effectively analyzed as a game. It has led to numerous research efforts in this domain. Jia et al. [6] investigated anti-jamming communication in wireless networks from the perspective of Stackelberg games. Jia et al. [7] explored the framework and challenges of game theory in learning anti-jamming, providing an analysis of various anti-jamming game models. Reinforcement Learning (RL) has also gained significant traction as an effective method for addressing anti-jamming in communication. A common approach involves leveraging the Q-Learning algorithm proposed in [8,9] to obtain the optimal anti-jamming policy by querying the Q-value table. However, the traditional Q-Learning algorithm faces challenges in solving high-dimensional spectrum state problems due to the complexity of the spectrum environment. To overcome this issue, Liu et al. [10] introduced Deep Reinforcement Learning (DRL) techniques in communication anti-jamming. The Recurrent Convolutional Neural Network (RCNN) is involved in processing infinite spectrum environmental states and extract relevant features. Compared with traditional RL methods, the optimized RCNN significantly improves convergence speed, offering a more effective solution.

To combat unknown jamming threats and develop optimal anti-jamming policies, intelligent frequency decision anti-jamming technology has been garnering significant attention. In the pursuit of enhancing communication's anti-jamming capabilities within dynamic adversarial environments, Chang et al. [11] introduced an improved anti-jamming method that leverages feature engineering and DRL. This method exhibits superior performance and reduces computational complexity. Liu et al. [12] proposed a sequential DRL algorithm without prior information to tackle the anti-jamming communication issues in a dynamic, intelligent jamming environment. This algorithm enables the rapid and effective selection of anti-jamming channels. Additionally, Li et al. [13] presented an anti-jamming algorithm utilizing Parallel Policy Networks based on Deep Q-Networks (DQN). This algorithm adjusts power levels and accesses idle channels simultaneously, aiming to resist intense jamming attacks. Furthermore, it explores dynamic spectrum anti-jamming access. Han et al. [14] investigated the application of DRL. With the help of a trained channel decision network, the secondary users are expected to be guided, and they can flexibly access the spectrum environment in the presence of jamming. In addition, Li et al. [15] delved into the frequency selection problem in a jamming environment with a vast number of available frequencies. They proposed a hierarchical DRL algorithm that effectively addresses various jamming scenarios, even without prior knowledge of the jamming mode and channel model. Collectively, these studies present various approaches that utilize DRL to enhance intelligent anti-jamming techniques. They demonstrate promising policies that adapt to dynamic jamming environments, optimize channel selection, and successfully resist jamming attacks without prior knowledge of the jamming type or channel model.

While the intelligent frequency decision anti-jamming algorithms discussed in [11–15] have shown promise in combating jamming, they share a common assumption: the existence of an unjammed control channel for direct information transmission. However, in unknown and dynamic environments where information is incomplete, maintaining a reliable control channel is often impractical. It becomes difficult for communication terminals to exchange policy information without a predetermined frequency change sequence. Consequently, achieving FH synchronization autonomously becomes a significant challenge in the absence of a control channel.

In consideration of the limitations associated with the ideal control channel in existing intelligent frequency decision anti-jamming techniques, this paper focuses on studying an intelligent frequency decision scheme that does not rely on an additional control channel. The research objectives of this study are as follows:

- How to design a communication system without communication negotiation?
- How to design a game model to ensure that the two terminals involved in the communication can achieve convergence and converge to the optimal outcome?

- How to design an algorithm that allows the two terminals to learn synchronously and find an equilibrium?

The rest of the paper is organized as follows. In Section 2, we provide a review of the related work in the field, discussing the existing research on intelligent frequency decision communication. In Section 3, a framework for intelligent frequency decision communication is introduced, which does not rely on a control channel. In Section 4, an OPG model is constructed and analyzed. The NE existence of the model is proved. In Section 5, a two-agent frequency decision learning algorithm based on the best-response policy is proposed. The algorithm is designed to converge to the NE. In Section 6, the simulation results are presented and thoroughly analyzed. In Section 7, conclusions are shown. All the frequent abbreviations used in our work are shown in Abbreviations.

2. Related Work

The problem of anti-jamming, based on game theory, has been extensively researched. Game theory provides a useful framework for modeling, implementing jamming countermeasures, and determining optimal anti-jamming policies. Xu et al. [16] took a bird's eye view of the issue of multi-user anti-jamming spectrum access and developed a game model to prove the existence of an NE. Jia et al. [17] presented a dynamic game approach to dealing with the problem of anti-jamming channel selection. They propose a distributed algorithm that converges to the NE of the game in a dynamic environment. However, applying game theory methods to the intricate and ever-changing electromagnetic environment often requires access to upgrading precise jamming parameters and environmental information, which can be challenging to obtain in real-world scenarios.

With the rapid development of artificial intelligence technology, the practical value of RL is becoming more evident. In the dynamic jamming wireless channel, the jamming parameters are no longer necessary. Instead, the agent interacts directly with the environment to learn how to implement counter-jamming and gain an advantage position in real-world combat scenarios. In [18], Xiao et al. examined the problem of anti-jamming power control for secondary users in large-scale cooperative cognitive radio networks. They utilized RL methods, such as Q-learning [19], to achieve optimal anti-jamming power. However, as the spectrum environment grows more complex, the curse of dimensionality may bring extra difficulties for the RL algorithm. DRL leverages the powerful perception capabilities of deep learning in vision and other areas. It is combined with the decision-making abilities of RL and can enable end-to-end learning. This approach partially alleviates the issue of dimensionality. Despite the increasing complexity of dynamic jamming in wireless channels, it has been successfully applied in the field of communication to handle intelligent anti-jamming decision-making in high-dimensional, complex, and dynamic environments, yielding significant results. Specifically, the literature [11–13] demonstrated how the agent can effectively achieve intelligent anti-jamming communication by utilizing the DRL algorithm to perceive environmental spectrum information.

However, as the number of wireless communication devices continues to grow, it becomes increasingly challenging for agents to develop proficient policies, understand tactics, and effectively collaborate in multi-user environments. Solely relying on a single agent is insufficient to solve the coordination problem among multiple agents. Therefore, many existing intelligent anti-jamming technologies have been extended to include research on multi-agent systems. In [20], Yao et al. examined the problem of anti-jamming defense in multi-user scenarios. They utilized a Markov Game framework for modeling and analysis and proposed a cooperative multi-agent anti-jamming algorithm (CMAA) to obtain optimal anti-jamming policies. In [21], a distributed multi-agent reinforcement learning (MARL) anti-jamming algorithm was proposed to address the challenge of relying on a reliable control channel for information exchange between users in multi-agent cooperative learning algorithms. While some other works [20,21] have conducted research on the cooperative anti-jamming problem in multi-user scenarios, they all rely on the control channel to facilitate information interaction among users. However, it is often arduous to achieve an

ideal control channel in real-world environments, and the decisions made by one agent can greatly impact other agents. Consequently, studying an intelligent frequency decision communication method without the strong limitations on a control channel. They are of great importance of theoretical and practical value.

In response to the above challenges, an OPG model is introduced, and it is deeply analyzed to prove the existence of NE in the game. To verify the model's capability to converge to an NE, a two-agent DRL algorithm is proposed based on the best-response policy. The proposed method eliminates the need for information exchange among users, allowing them to achieve FH synchronization via self-learning.

3. Intelligent Frequency Decision Communication Framework without Control Channel Assistance

As illustrated in Figure 1, an intelligent frequency decision communication system without a control channel is introduced. The system comprises a pair of intelligent nodes and one or more jamming nodes. In this scenario, the communication terminals have not prearranged a FH sequence and are all functioning within a jamming environment. Each node in the system is equipped with a transmitter, receiver, agent, and sensing device. Notably, node B utilizes a multi-channel receiver, enabling it to simultaneously receive signals across multiple channels. Therefore, they increase the chances of capturing the transmission frequency. The data transmission process between node A and node B is illustrated in Figure 2. Node A serves as the primary node and is responsible for initiating communication, while node B acts as a secondary node. During a round of data transmission, the primary node A selects a communication frequency based on the sensing information to initiate communication, while the secondary node B selects a group based on the sensing information to await reception. If node B scans and identifies node A's transmission frequency within its selected group of frequencies, a successful frequency is matched. If node B successfully receives the data information, it is expected a rapid response by sending confirming information back to node A using the same frequency. If node A also successfully receives the confirming information, it signifies the completion of a normal round of communication under this circumstance.

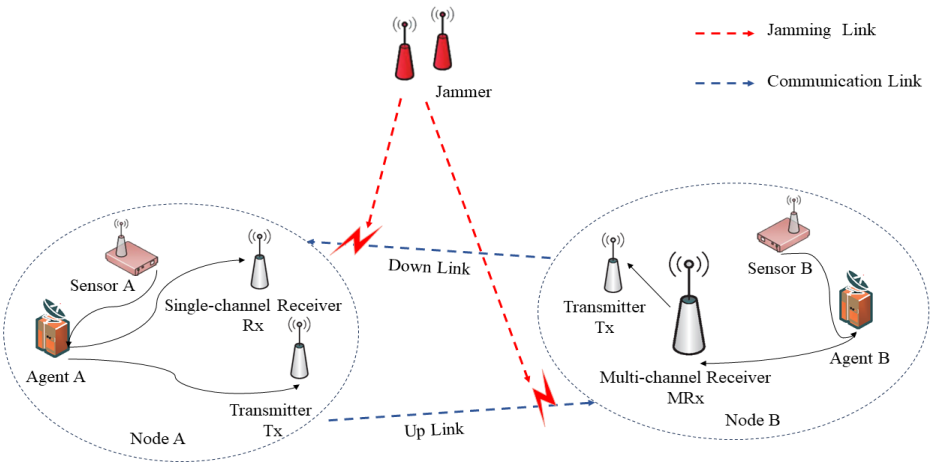

Figure 1. Model of intelligent frequency decision communication system without control channel assistance.

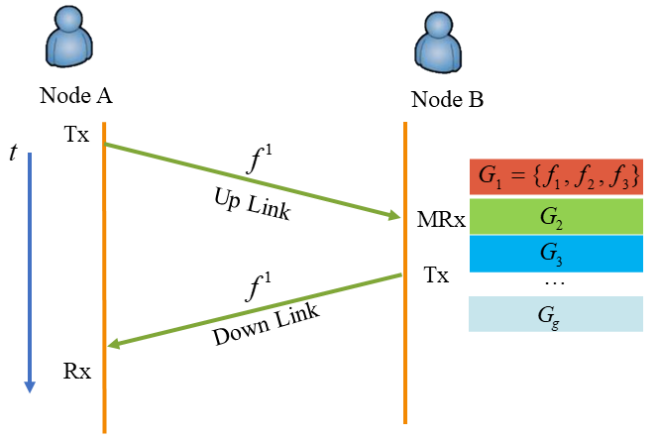

Figure 2. Data transfer process.

To facilitate modeling the transmission process, the continuous time is divided into a series of equal-length time slots $\{t_1, t_2, \cdots, t_i\}$. The process within the whole simulation time can be regarded as an iterative process, and the two communication terminals conduct a round of communication in a time slot. As is shown in Figure 3, Each time slot $t_i = \left\{t_i^u, t_i^d\right\}$ is further divided into uplink and downlink time slots of equal duration. $C = \{1, 2, \cdots, C\}$ is considered the number of wireless transmission channels that are available, with each channel having a bandwidth of w. The set of communication frequencies are $f = \{f_1, f_2, \cdots, f_c\}$, and the entire communication frequency band that is divided into distinct groups $G = \{G_1, G_2, \cdots, G_g\}$. In each group G_g, some frequencies are arranged in a group, such as $G_1 = \{f_1, f_2, f_3\}$. As mentioned earlier, during a single round of data transmission, node A selects a frequency for transmission. Therefore, we define the set of communication frequencies selected by node A as $G^1 = \left\{f_1^1, f_2^1, \cdots, f_c^1\right\}$. However, node B utilizes a multi-channel receiver and can only select one group for reception at a time; we define the set of groups selected by node B as $G^2 = \left\{G_1^2, G_2^2, \cdots, G_g^2\right\}$. Simultaneously, the jammer selects a jamming frequency to disrupt data transmission. We define the set of jamming frequencies as $f^J = \left\{f_1^J, f_2^J, \cdots, f_C^J\right\}$. P^J is considered as a parameter representing the power of the jamming signal. Assuming that in the time slot t, the received Signal-to-Interference-plus-Noise Ratio (SINR) of the receiver is expressed as follows:

$$SINR_t = \frac{P \times g}{P^J \times \delta(f = f^J) + N_0} \tag{1}$$

P is used as the transmit power, the link gain from transmitter to receiver is g; the background noise power is N_0; and the function $\delta(x)$ represents an indicator that is equal to one if the condition is true $\delta(x) = 1$ and zero otherwise.

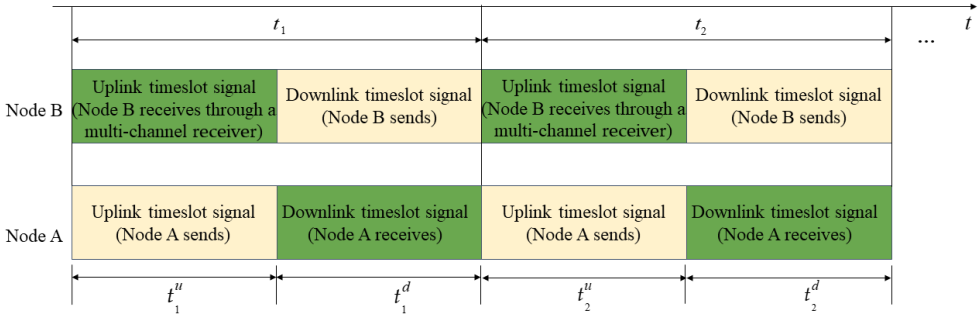

Figure 3. Timeslot structure.

In addition, $SINR_{th}$ represents the minimum detectable SINR threshold. If the output SINR of the receiver is greater than or equal to this threshold, it indicates successful data reception; otherwise, it denotes unsuccessful data reception. The reward function r_t is defined as follows:

$$r(f_t) = \begin{cases} 1, SINR_t \geq SINR_{th} \\ 0, SINR_t < SINR_{th} \end{cases} \quad (2)$$

Due to the dynamic properties of jamming, the agent is unable to determine the current jamming state. To record the current jamming state, we make the assumption that the spectrum vector detected by the sensing device in the time slot t is as follows:

$$o_t = [o_{t,1}, o_{t,2}, \cdots, o_{t,C}] \quad (3)$$

In the above Equation (3), $o_{t,C} = P_t \times g_t \times \delta(f_t = f_{t,C})$.

4. Potential Game Model of Frequency Decision without Information Interaction

The Materials and Methods should be described with sufficient details to allow others to understand that Stochastic games (SGs) are a combination of Markov Decision Processes (MDP) and game theory. They provide a framework to describe dynamic game processes where multiple decision-makers interact and make decisions repeatedly in various states. MDPs are primarily utilized to address decision problems in uncertain environments, while game theory offers tools to analyze the interactions among decision-makers. In our intelligent frequency decision communication system, the decision is influenced by the two intelligent nodes. They can be vividly illustrated and stimulated by using an SG model [22]. This model captures the frequency decision problem for both the receiver and the transmitter, taking into account the uncertainties and interactions involved in the communication process.

Definition 1. *The decision process of the two intelligent nodes is formulated as an SG, which is defined by a quintuple* $\langle N, S, A, R, P, \gamma \rangle$.

- $N = \{1, 2, \ldots, n\}$*denotes the set of decision-makers participating in the game. In this paper, specifically whenn* $= 2$*, the decision-makers involved in the game are node A and node B.*
- $S = \left\{ \boldsymbol{o}^1, \boldsymbol{o}^2, \cdots, \boldsymbol{o}^n \right\}$ *denotes the state space. The global spaceSconsists of two different state spaces of node A and node B. At timet*, the set of global states formed by the corresponding states of A and B is denoted as $S_t = \left\{ \boldsymbol{o}_t^1, \boldsymbol{o}_t^2 \right\}$.
- $A = \left\{ \boldsymbol{a}^1, \boldsymbol{a}^2, \cdots, \boldsymbol{a}^n \right\}$ *denotes the joint selection policy anda* $= \left\{ a^1, a^2 \right\}$ *denotes the joint anti-jamming policy of node A and node B.*$a^1 = \left\{ a_1^1, a_2^1, \cdots, a_C^1 \right\}$ *represents the* set of *policies available to node A, while* $a^2 = \left\{ a_1^2, a_2^2, \cdots, a_g^2 \right\}$ *represents the set of policies available to node B.*
- $R = \left\{ r^1, r^2, \cdots, r^n \right\}$ *denotes the reward function.*
- $P(S'|S, a) : S \times A \times S \to [0, 1]$ *denotes the state transition probability function of the SG.*
- $\gamma \in [0, 1]$ *is the discount factor.*

To analyze the evolution of the jamming state, we consider the Wth time slots and define it as the state of a single agent represented by $\boldsymbol{o}_t = [o_t, o_{t-1}, \ldots, o_{t-W+1}]$, where it represents the length of the historical time slot [10]. In Figure 4, the non-convergence of spectrum waterfall $\boldsymbol{o}_t^1$ of node A, denoted as $\boldsymbol{o}_t$, is a two-dimensional matrix $C \times W$ consisting of channels and time slots. Not only is the distribution of the signal in the time and frequency domains shown by this figure, but also the intensity of the signal is shown by the color depth. At time t, the agent based on the current state $\boldsymbol{o}_t$, selects a policy from the action set $\boldsymbol{a}_t$, and receives a reward r_t from the environment. Subsequently, the agent transitions to the next state $\boldsymbol{o}_{t+1}$ according to the state transition probability P.

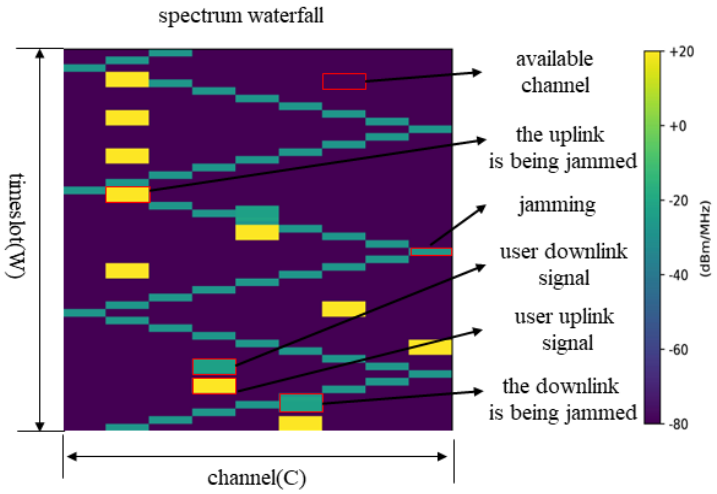

Figure 4. Sample signal distribution.

In potential games, the change in the utility function of each decision-maker resulting from a policy is proportionally mapped to the global potential function. In this paper, for the intelligent frequency decision communication scenario without a control channel, the utility of each node is designed to make it suitable for a potential game [23]. This ensures that the independent decisions of the communication terminals can ultimately converge to the optimal joint anti-jamming policy. In this context, the utility function is formulated as an indicator function. Its value only depends on the reward function. Specifically, the reward is 1 when data is successfully received and -1 when it is not. The utility function U^n is defined as follows:

$$U^n(a^n, a_{-n}) = r_t^n(a) \tag{4}$$

In the above Equation (4), a_{-n} represents the combination of policies of other decision-makers, excluding the decision-maker n. The decision-maker can adjust its own policy by analyzing the policies a_{-n} of other decision-makers, aiming to maximize the value of its utility function U^n:

$$\underset{a^n}{\text{argmax}} U^n(a^n, a_{-n}), \forall n \in N \tag{5}$$

The decision-maker's policy will be iteratively adjusted, and the value of the utility function will demonstrate a monotonic change as the policy is coordinated. Via a finite number of iterations, it will ultimately converge to a stable state known as NE.

Definition 2. *As previously defined, the policy of node A is denoted as* a^1 *and the policy of node B is denoted as* a^2*. In the non-cooperative game model, the policies* $a^* = \{a^{1*}, a^{2*}\}$ *exist where neither player in the satisfaction game can maximize their utility or payoff by unilaterally changing their policies:*

$$U^n(a^{n*}, a^*_{-n}) \geq U^n(a^n, a^*_{-n}), \forall a^n \in A^n, n \in N \tag{6}$$

The set of policies a^* *is adopted to represent the NE of the game.*

In the intelligent frequency decision communication system, for node A and node B, the utility function is represented as $U = \{U^1, U^2\}$, $U^1 = r^1$, $U^2 = r^2$. The magnitude of the utility function value is influenced by both policy a^1 and policy a^2. As the existence of NE cannot be guaranteed, potential games are introduced to analyze the NE problem.

Definition 3. *If there exists an ordinal potential function* $P : A \to R$ *for* $\forall n, \forall a_{-n} \in A_{-n}$ *and* $\forall a^n, a^{n'} \in A^n$*:*

$$P(a^{n'}, a_{-n}) - P(a^n, a_{-n}) > 0 \tag{7}$$

then

$$U^n(a^{n'}, a_{-n}) - U^n(a^n, a_{-n}) > 0 \tag{8}$$

If such conditions are guaranteed, then it can be classified as an OPG.

A potential function is constructed to prove that the problem of intelligent frequency decision communication without a control channel is an OPG [24] problem. It is ensured that it can converge to an NE within a finite number of iterations.

Theorem 1. *The intelligent frequency decision communication problem without a control channel is an OPGP*(a^n, a_{-n})*. An ordinal potential function is defined as the sum of the utility values*U^n*of all nodes:*

$$P(a^n, a_{-n}) = \sum_{n \in N} U^n(a^n, a_{-n}) \tag{9}$$

To establish that the aforementioned problem is an OPG, it is crucial to demonstrate that the utility value increases when a player n updates their policy from a^n to $a^{n'}$, consequently elevating the overall situation function $P(a^{n'}, a_{-n}) > P(a^n, a_{-n})$.

A proof of Theorem 1 is provided in Appendix A. By constructing the ordinal potential function P, it is proven that the intelligent frequency decision communication problem without a control channel is an OPG. Theorem 1 guarantees the existence of at least one NE solution. Moreover, this equilibrium that maximizes OPG is also the Pareto optimal solution [25,26]. Pareto optimality [27] refers to a combination of policies that maximizes the utility of all players involved, thereby constituting the global optimal solution to the problem.

5. Two-Agent Frequency Decision Learning Algorithm

5.1. Q-Learning and DQN Algorithm

Q-learning algorithm is a classical RL algorithm. Traditional Q-learning is presented in the form of a Q-value table to store the Q-value of each state-action pair. The goal of RL is to maximize long-term future rewards. The environment gives the agent a reward R_{t+1}after taking an action a_t in each state s_t. The cumulative discounted return is defined as

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \tag{10}$$

In MDP, the agent takes an action a_t according to the policy π at the state s_t, and the expected reward is defined as the state action value function:

$$Q_\pi(s_t, a_t) = E_\pi[G_t | S_t = s_t, A_t = a_t] \tag{11}$$

The optimal state action value function is to select the function with the largest state action value from the state action value functions generated by all policies:

$$Q^*(s_t, a_t) = \max_\pi Q_\pi(s_t, a_t) \tag{12}$$

In practice, however, Q-learning in tabular form is difficult to implement. Because in practical applications, the number of states and actions is often very large, it requires a lot of storage space and computing resources to maintain the Q-value table. To solve this problem, neural networks emerged instead of Q-value tables. DQN uses deep neural networks to approximate the optimal action-value function and successfully solves the complex problem of huge states and action spaces. The update procedure for the q-function can be defined as follows:

$$Q_t(s_t, a_t) \leftarrow (1 - \alpha) Q_t(s_t, a_t) + \alpha(r_t + \gamma max Q_{t+1}(s_{t+1}, a)) \tag{13}$$

Here, the function $Q(s,a)$ fits the long-term payoff evaluation of a state-action pair. s_t and a_t are denoted as the agent's current state and action, respectively. $Q_t(s_t,a_t)$ denotes the value of Q corresponding to the action performed in the current state s_t. $maxQ_{t+1}(s_{t+1},a)$ is the maximum value of all Q values in the state at the next time. α is the learning rate of the neural network and γ is the discount factor.

In DQN, the idea of function approximation is used to find a set of optimal parameters and thus determine the best-fitting function. The whole training process can be viewed as a process in which the Q-value is constantly approaching the target Q-value. The Q-value represents the current Q-value, and the target Q-value is the score that the agent obtains by interacting with the environment. During training, learning is performed by using the target Q-value to update the original Q-value. The loss function can be defined as follows:

$$L(\theta_t) = [Q_{t_{target}} - Q_t(s_t,a_t,\theta_t)]^2 \tag{14}$$

In the above Equation (14), $Q_{ttarget} = r_t + \gamma maxQ_{t+1}(s_{t+1},a_{t+1},\theta_t^-)$. θ_t denotes the parameters of the prediction network and θ_t^- denotes the parameters of the target network. After the weight parameters of the prediction network are updated, the target network replicates to update its weight parameters. In DQN, experience replay is used to train the agent. The agent is allowed to explore the environment first, after which the experience value is stored. After the experience value has accumulated to a certain extent, another batch of experience is randomly selected for training.

In practice, agents often become stuck in local optima, resulting in poor learning. To solve this problem, the ε-greedy policy is often used to make the agent fully explore the environment to find the optimal decision. Each time the agent selects an action, it will select the action with the largest Q-value with some probability and select a random action with some probability. The value decreases as the agent explores. The greedy policy is defined as follows:

$$a_t = \begin{cases} \underset{a}{\text{argmax}} Q(s_t,a;\theta), 1-\varepsilon \\ a_{\text{random}}, \varepsilon \end{cases} \tag{15}$$

5.2. Two-Agent Frequency Decision Learning Algorithm Flow

Multi-agent learning algorithms, such as MARL, provide a convenient approach to verify the convergence of a game model toward NE. Therefore, this paper proposes a two-agent RL algorithm based on best response (Algorithm 1). Our novel method aims to investigate the optimal anti-jamming policy and equilibrium in the dynamic game between the two terminals of communication. The fundamental concept of this algorithm is to address the challenge that the two terminals of communication cannot acquire the global state. Each node serves as an individual agent that interacts with the jamming environment over multiple iterations. The ε-greedy policy was used to adjust the policy according to the perceived environmental state information, and the corresponding channel was selected to increase the utility value, and finally, the intelligent frequency decision was realized. The policy is chosen randomly for the first I time slots, after which the ε-greedy policy is adopted.

During the training process of the anti-jamming algorithm, each node $n(n=1,2)$ is equipped with two neural networks. An online Q network with parameters θ_t^n is utilized for action reasoning, while a target Q network with parameters θ_t^{n-} is employed for parameter update learning [28]. To enhance learning efficiency, an experience replay technique is employed, which allows for the reuse of previous experiences stored in the experience replay pool. This approach effectively breaks the correlation among experience data and maximizes data utilization. During each iteration, a batch of experience values are randomly sampled from the experience replay pool, and these experiences are used to update the parameters of the neural network. The Q-value update is performed as follows:

$$Q_t^n(o_t^n,a_t^n) \leftarrow (1-\alpha)Q_t^n(o_t^n,a_t^n) + \alpha\left(r_t + \gamma \text{max} Q_{t+1}^n\left(o_{t+1}^n,a_{t+1}^n\right)\right) \tag{16}$$

The loss function of the online Q-network is minimized as follows:

$$L(\theta_t^n) = [Q_{t\,\text{target}}^n - Q_t^n(o_t^n, a_t^n, \theta_t^n)]^2 \tag{17}$$

In the above Equation (17), θ_t^n represents the weight parameter of the online Q-network, and the target Q-network $Q_{t\,\text{target}}^n$ prediction is compared to the true observation in order to determine the reward r_t.

$$Q_{t\,\text{target}}^n = r_t + \gamma \text{max} Q_{t+1}^n(o_{t+1}^n, a_{t+1}^n, \theta_t^{n-}) \tag{18}$$

θ_t^{n-} represents the weight parameter of the target Q-network.

Stochastic Gradient Descent (SGD) is utilized to train the parameters of the online Q-network. The gradient of the loss function L with respect to θ is calculated as follows:

$$\nabla_{\theta_t^n} L(\theta_t^n) = [r_t + \gamma \text{max} Q_{t+1}^n(o_{t+1}^n, a_{t+1}^n, \theta_t^{n-}) - Q_t^n(o_t^n, a_t^n, \theta_t^n)] \nabla_{\theta_t^n} Q(o_t^n, a_t^n, \theta_t^n) \tag{19}$$

$\tau \in [0, 1]$ is adopted as the update rate to adjust the parameters in the target Q-network. After enough iterations, the weight parameters of the target Q-network are updated by applying a soft update, which is between the current target Q-network parameters and the online Q-network parameters:

$$\theta_t^{n-} \leftarrow (1 - \tau)\theta_t^{n-} + \tau\theta_t^n \tag{20}$$

After the training process, the parameters θ_t^n of the neural network are saved. In a complex and dynamic jamming environment, both terminals involved in the communication only need to load their neural network parameters locally after the training process. This allows them to respond to the observed local environment state without rapid extensive training.

Algorithm 1: Two-agent DRL Algorithm Based on best-response (Training Phase)

Input: Experience replay pool D^n, network parameters θ_t^n, θ_t^{n-}, state o_t^n, $n = 1, 2$.
Output: Parameters of the trained network θ_t^n, θ_t^{n-}.

```
for t = 1 to T do
    if t ≤ I
        Nodes n(n = 1,2) randomly select an anti-jamming policy a_t^n
        based on the current state o_t^n of the game;
    else
    The anti-jamming policy o_t^n is selected according to
    the ε-greedy policy;
Node A performs the action a_t^1, node A performs the action a_t^2, the utility value U^1, U^2 is
obtained according to Equation (4) and transferred to the next game state o_{t+1}^n;
The experience values (o_t^n, a_t^n, r_t^n, o_{t+1}^n) are put into the experience replay pool D^n of node n,
respectively;
    for n = 1,2 do
        Batch experience values (o_t^n, a_t^n, r_t^n, o_{t+1}^n) are randomly sampled
        from the experience replay pool D^n;
        Gradient descent is performed on the loss function L(θ_t^n) to update θ_t^n
        according to Equation (19);
        The target network parameters are updated at intervals θ_t^{n-} = θ_t^n;
    end for
end for
until reaching a NE
```

5.3. Computational Complexity Analysis

Considering the memory limitation and computational power scheduling, the algorithm complexity of the network model is thoroughly analyzed. The online Q-network and

the target Q-network are set to have the same structure, consisting of one convolutional layer and two fully connected layers. In the convolutional layer, 16 convolution kernels of size 10×4 are utilized and the stride is set to 2. The first fully connected layer consists of 512 neurons, while the number of neurons in the second fully connected layer matches the size of the action space for a single agent. The overall computational complexity for one forward propagation is represented as $O(Time)$ and can be calculated as follows [29]:

$$\text{Time} \sim O(\sum_{l=1}^{L} M_l^2 \times K_l^2 \times H_{l-1} \times H_l + \sum_{v=1}^{V} 2 \times Y_v \times Z_v - 1) \tag{21}$$

where L represents the number of convolutional layers in the neural network; l represents the lth convolutional layer; the time complexity of a single convolutional layer is determined by the dimensions of the output feature map M^2; the dimensions of the convolution kernel is represented by K^2; the number of input feature map channels is denoted as H_{l-1}; and output feature map channels is denoted as H_l. Similarly, V represents the number of fully connected layers in the neural network, v represents the v-th fully connected layer, and the time complexity of each fully connected layer is determined by the number of input neurons Y and the number of output neurons Z.

When updating the network parameters, *minibatch* samples can be selected for node n randomly from the experience replay pool D^n to update the neural network parameters. During the gradient descent process using the loss function $L(\theta_t^n)$, it requires $2 \cdot minibatch$ forward propagations and backward propagations. As a result, each round of the game involves a computational complexity of $O(2 \cdot (minibatch + 1) \cdot \text{Time})$.

5.4. Description of Simulation Parameters

In this paper, the simulation experiment utilizes the tensorflow2.8.0 deep learning development framework for building the network model. Python 3.9 is used in Windows 11 for our simulations. The simulation is run on a PyCharm platform with a GeForce RTX 4060 GPU. The CPU used in the experiments is Intel Core i5-13500HX. Adam is chosen as the optimizer during the training process of the neural network. During the simulation, the number of nodes N is 2, and both the nodes and jammers deployed in the same environment transmit data in each channel. The bandwidth w is 1 MHz. The transmit power of the two nodes is 100 dBm, the length of each time slot t is 10 ms, the history duration W of the spectral waterfall is 100 ms, the transmit power of the jammer is 1000 dBm, the SINR threshold $SINR_{th}$ is 6 dB, and the background noise power is -80 dBm. The learning rate α of the algorithm is set to 0.5, the discount factor γ is set to 0.8, the greedy factor ε is set to 1.0, and the soft update coefficient τ is set to 0.01. When updating the network parameters, each node is trained by randomly sampling a batch of 128 samples from the experience replay pool D, which has a capacity of 5000. Different simulation conditions are established by adjusting the simulation time T, the element combination of a single group G_g in the group set G, the number of groups g, and the number of available channels C. The parameters for the rest part of the simulation experiment are shown in Table 1.

Table 1. Simulation parameter setting.

Parameter Name	Symbol	Parameter Value
Number of nodes	N	2
Grouping collections	G	$\{[1,2,3],[4,5,6],[7,8,9]\}$
Group numbers	g	3
Bandwidth	w	1 MHz
Duration of history	W	100 MHz
Transmit power	P	100 ms
SINR threshold	$SINR_{th}$	6 dB
Time slot length	t	10 ms
Jammer power	P^J	1000 dBm

Table 1. *Cont.*

Parameter Name	Symbol	Parameter Value
Background noise power	N_0	−80 dBm
Discount factor	γ	0.8
Greedy factor	ε	1.0
Soft update coefficient	τ	0.01
Learning rate	α	0.5
Sampling batch size	*minibatch*	128
Experience pool capacity	D	5000

6. Simulation Results and Analysis

To verify the effectiveness of the proposed algorithm, simulation experiments are conducted in three typical jamming environments: sweeping jamming, comb jamming, and dynamic jamming:

(1) Sweeping jamming

In this scenario, the jammer performs the sweeping jamming at a specific frequency. The frequency sweep rate is 1 MHz/ms.

(2) Comb jamming

In this scenario, the jammer simultaneously implements jamming at multiple frequencies. The specific frequencies 2 MHz, 6 MHz, and 9 MHz for jamming are selected.

(3) Dynamic jamming

In this scenario, the jamming environment is dynamic. It periodically alternates between comb jamming and sweep jamming. Once selected, a jamming mode will remain unchanged within a certain duration of time, which is 10 ms. If switching to comb jamming, the specific frequencies 1 MHz, 5 MHz, and 9 MHz for jamming are selected.

Firstly, Figures 5 and 6 illustrate the spectrum waterfall in the initial and convergent states, respectively, for a channel number of 9 and a group number of 3, under different jamming environments. The figures reveal that the sweeping jamming technique executes a periodic linear scan of each channel, resulting in narrowband jamming with the jamming frequency exhibiting a linear variation. During the uplink time slot, node A transmits the signal, while node B acts as the receiver. Conversely, during the downlink time slot, node B becomes the transmitter, and node A takes on the role of the receiver.

It can be observed that two figures exist for each jamming environment. The left figure represents the spectral waterfall of node A, while the right figure represents the spectral waterfall of node B. The horizontal axis of each node's spectrum waterfall represents the time slot, while the vertical axis represents the channel. The figures provide a clear visualization of the available idle channels at each node, as well as the changes in the user's uplink signal, downlink signal, and jamming signal over time. The user signal is represented by a pair of yellow-green rectangular color blocks. The yellow rectangular block indicates that the node is transmitting a signal during that time slot, while the green rectangular block represents the node receiving a signal. If the color blocks overlap, it indicates jamming with the user signal. Figure 5 illustrates the initial state, where fewer pairs of yellow-green blocks are observed, suggesting that the two communicating nodes have not yet synchronized or have been disrupted by jamming during the communication process. In contrast, Figure 6 displays the convergence state, where all the yellow and green blocks appear in pairs without overlapping with other color blocks. This indicates that both node A and node B have achieved frequency decision synchronization in various dynamic jamming environments. Although the two terminals of communication are not aware of the specific changes in the jamming environment, they can effectively avoid jamming via the convergence decision after learning, which demonstrates the effectiveness of the algorithm.

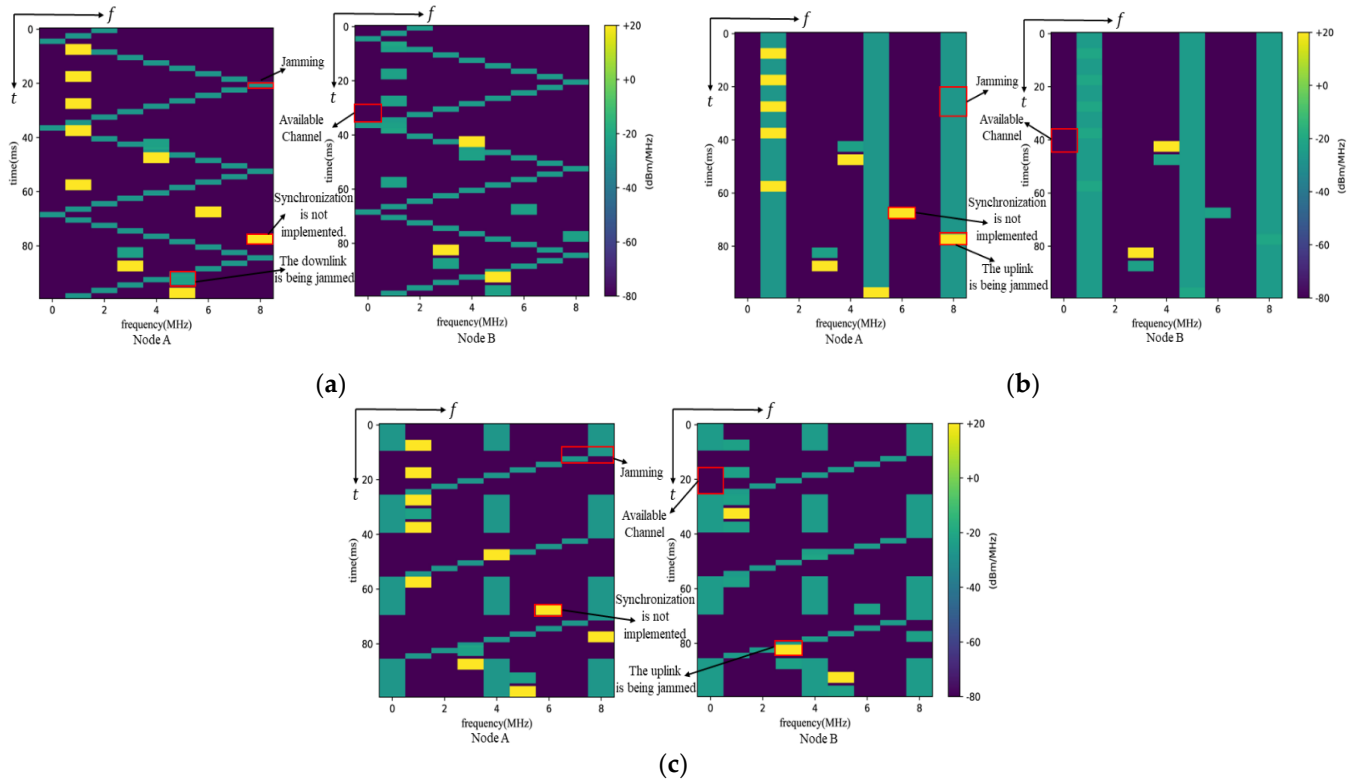

Figure 5. Time –frequency information of initial state. (**a**) Sweeping jamming; (**b**) Comb jamming; (**c**) Dynamic jamming.

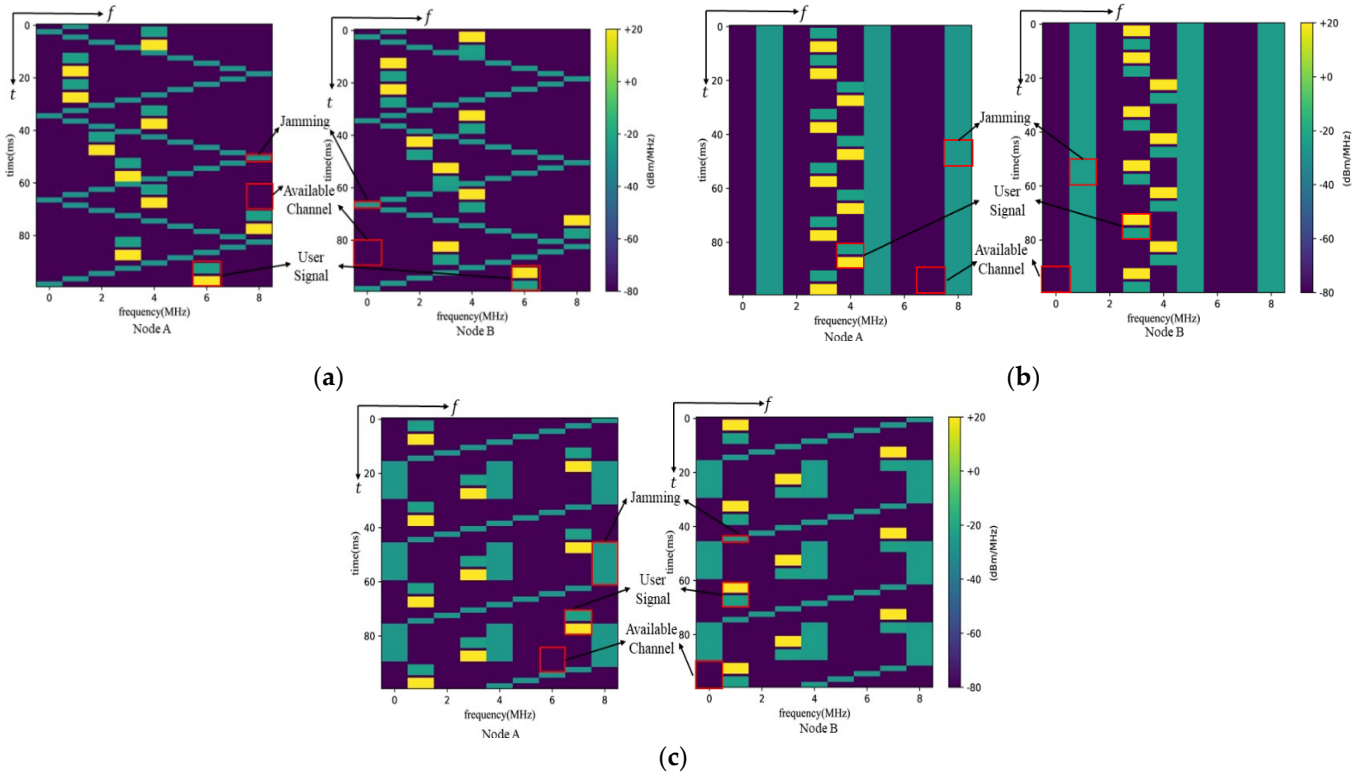

Figure 6. Time –frequency information of convergence state. (**a**) Sweeping jamming; (**b**) Comb jamming; (**c**) Dynamic jamming.

Secondly, in order to evaluate the influence of the number of available channels on the proposed algorithm, the number of channels was increased to 12, 15, and 18, respectively, while keeping the number of groups unchanged, and comparative experiments were carried out. By conducting these experiments, we aimed to evaluate the performance of the algorithm in scenarios with an increased number of channels. The results of the experiments are presented in Figures 7–10. From the figures, it can be observed that in all cases, the convergence state is achieved after a certain number of iterations (4000, 10,000, and 12,000). As the number of channels increases, the decision-making complexity for the two terminals of communication also increases. The environment becomes more intricate, and the probability of achieving synchronization decreases, resulting in a slower convergence speed. Nevertheless, despite these challenges, it is worth noting that the two terminals of communication are still able to converge to the optimal decision even with the increased number of channels. This result demonstrates the effectiveness and applicability of the proposed algorithm in scenarios with a higher number of channels. In conclusion, our experiments indicate that the algorithm proposed in this paper can effectively handle scenarios with different numbers of channels, maintaining its superior performance and showing its potential in real-world applications.

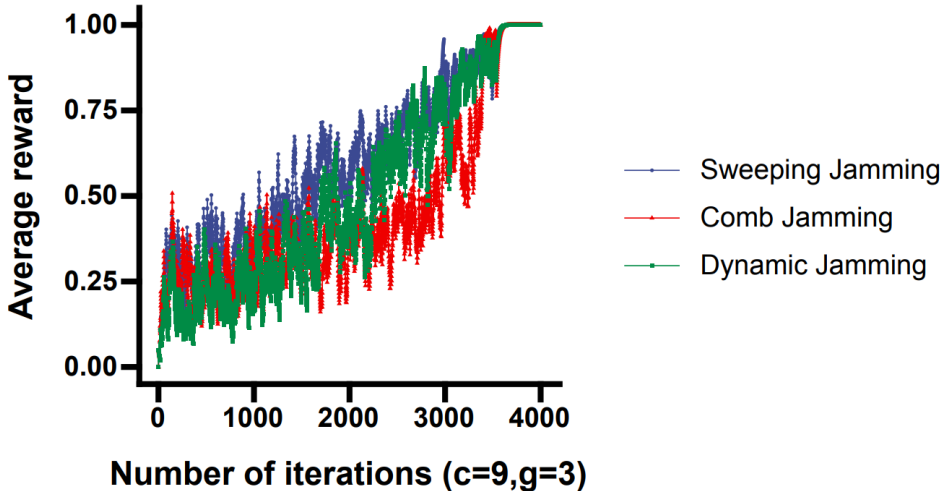

Figure 7. The average reward varies with the number of iterations in different jamming environments (c = 9, g = 3).

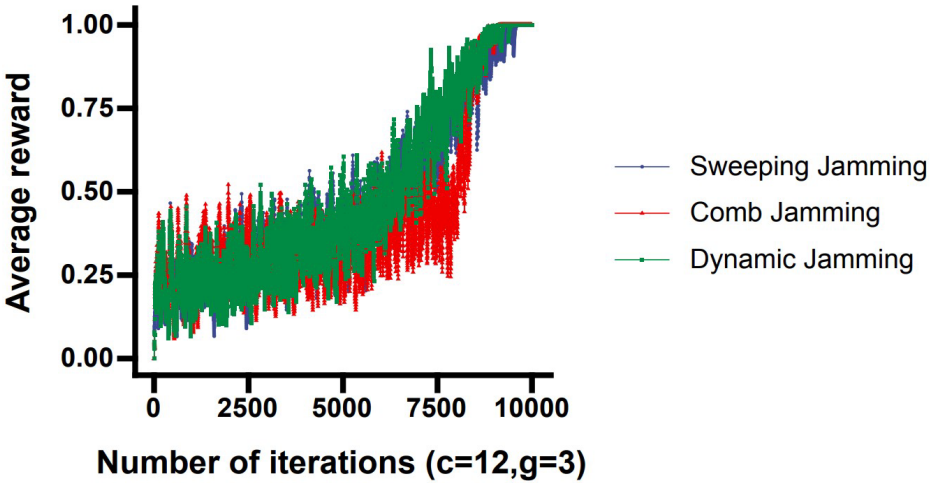

Figure 8. The average reward varies with the number of iterations in different jamming environments (c = 12, g = 3).

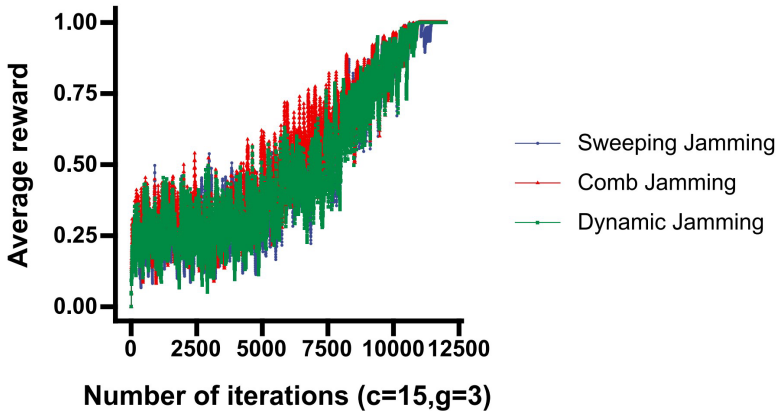

Figure 9. The average reward varies with the number of iterations in different jamming environments (c = 15, g = 3).

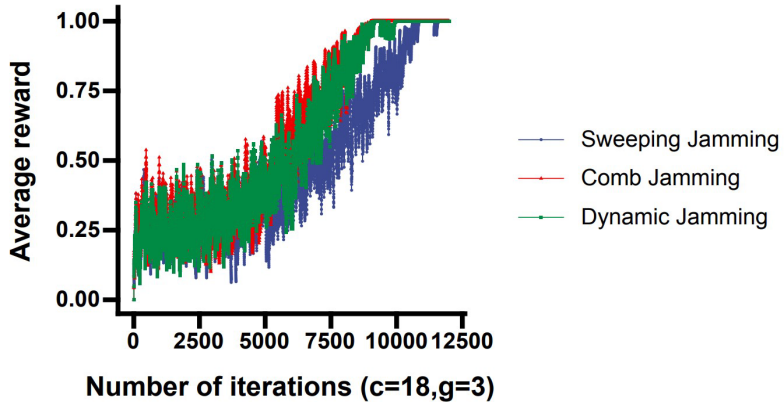

Figure 10. The average reward varies with the number of iterations in different jamming environments (c = 18, g = 3).

Finally, the performance of the proposed algorithm in three different jamming environments is compared with that of the method using the presence of control channels:

(1) PPQN-AJ [13]: An anti-jamming algorithm based on a DQN parallel policy network is proposed, which adaptively selects power and channel.
(2) ADRLA [10]: The proposed method takes the spectrum environment as the input state and uses DQN to continuously try different actions and sense the spectrum environment in order to learn the optimal anti-jamming policy.

In Table 2, it can be seen that the normalized throughput performance of different methods under different jamming environments. In the comb-jamming environment, the throughput of the communication system is relatively high. This is because the pattern of comb jamming is relatively stable, and if the uplink is not jammed, then the downlink is necessarily not jammed either. Meanwhile, Table 2 also provides the number of iterations required to reach the convergence state. It can be observed that there is no significant difference between the proposed algorithm and the method with control channels in terms of performance. The convergence speed of the proposed algorithm is relatively slow. This is mainly because, in the proposed method, both terminals of communication cannot exchange information and need to realize tacit communication via continuous learning and mutual cooperation. Nevertheless, the strength of the proposed method lies in abandoning the conventional approach of relying on control channels for interaction to achieve synchronization and instead realizing a more independent decision-making process.

Table 2. The performance comparison between the proposed algorithm and the existing control channel method.

Method	Sweeping Jamming		Comb Jamming		Dynamic Jamming	
	Throughput	Iterations	Throughput	Iterations	Throughput	Iterations
Ours	0.74	4000	0.92	4000	0.78	4000
PPQN-AJ	0.80	1200	0.94	1200	0.75	1200
ADRLA	0.79	1200	0.93	1200	0.75	1200

Experimental results show the efficiency and feasibility of this method. The proposed algorithm exhibits strong robustness and adaptability in typical jamming environments, including sweeping jamming, comb jamming, and dynamic jamming. It effectively mitigates the influence of various jamming signals, ensuring reliable and stable communication.

7. Discussion and Conclusions

In this paper, we focus on the problem of intelligent frequency decisions in the absence of control channels. Our proposed framework avoids the need for a control channel by enabling continuous iteration and adjustment within the communication system. By transforming the frequency decision problem between the transmitter and receiver, we introduce an SG model. Additionally, we design the utility of each node to meet the requirements of OPG, ensuring the attainment of global optimal decision and equilibrium in the dynamic game. To solve the equilibrium policy of a two-agent game, a novel DRL algorithm based on the best response is proposed. Via iterative updates of the policies of both terminals of communication, the algorithm can eventually converge to an NE, even in scenarios where the jamming parameters are unknown. The purpose of introducing RL into the algorithm is to search for the optimal decision. Due to the dynamically changing environment of the two communication terminals, directly determining the optimal decisions that both terminals can reach is challenging. RL is well-suited for exploring unknown environments. Hence, the DRL method is employed to enable the two communication terminals to eventually converge to the same policy consistently via trial and error. Simulation experiments are conducted to verify the anti-jamming performance of the proposed scheme in various jamming scenarios. The results confirm the effectiveness of the scheme.

In future research, there will be ongoing efforts to study and develop a more efficient and flexible intelligent frequency decision communication scheme. This will enable both terminals of communication to rapidly select the optimal joint anti-jamming policy. At the same time, the more complex situation will be considered, that is, how to realize the frequency decision synchronization in the frequency decision network with three nodes or even multiple nodes participating.

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

Funding: This research work was supported by the National Natural Science Foundation of China under Grant 61961010,62071135, the Key Laboratory Found of Cognitive Radio and Information Processing, Ministry of Education (Guilin University of Electronic Technology) under Grant No. CRKL200204, No. CRKL220204, RZ18103102, the 'Ba Gui Scholars' program of the provincial government of Guangxi.

Data Availability Statement: All the grants for this manuscript are still in the research phase, and some research data or key codes are currently limited to disclosure within the project team. However, the datasets used and/or analyzed during the current study are available via the email (1020210999@glut.edu.cn) upon reasonable request.

Acknowledgments: We are very grateful to volunteers from GLUT for their assistance in the experimental part of this manuscript.

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

Abbreviations

Abbreviations	Descriptions
FH	Frequency-Hopping
RL	Reinforcement Learning
DRL	Deep Reinforcement Learning
DQN	Deep Q-Networks
MARL	Multi-agent reinforcement learning
NE	Nash equilibrium
SG	Stochastic game
OPG	Ordinal-potential game
MDP	Markov Decision Process

Appendix A

Proof of Theorem 1. By combining Equations (2) and (4) in this paper, we can deduce the relationship between the utility values of node A and node B:

$$U^1(a^1, a^2) = \begin{cases} 1, U^2(a^2, a^1) = 1 \\ 0, U^2(a^2, a^1) = 0 \text{ or } 1 \end{cases} \tag{A1}$$

In Equation (A1), $U^1(a^1, a^2)$ represents the utility value of node A, while $U^2(a^2, a^1)$ represents the utility value of node B. This equation implies that if node A has a utility value of 1, then node B must also have a utility value of 1. On the other hand, when node A has a utility value of 0, node B can have a utility value of either 0 or 1. This indicates that for node A to successfully receive feedback data information, node B must also successfully receive data and feedback data information at the same frequency. Otherwise, there is no possibility for node A to receive the feedback information. Furthermore, if node A fails to receive the feedback, two cases can arise:

(1) Node B fails to receive in the uplink, resulting in no further feedback data information.
(2) In the downlink, node A suffers from jamming, resulting in a failure to receive the feedback information.

$$U^2(a^2, a^1) = \begin{cases} 1, U^1(a^1, a^2) = 0 \text{ or } 1 \\ 0, U^1(a^1, a^2) = 0 \end{cases} \tag{A2}$$

In contrast, Equation (A2) indicates that when the utility value of node B is 0, the utility value of node A must be 0. Node A can have a utility value of either 0 or 1, whereas node B has a utility value of 1. This means that if node A fails to receive in the uplink, it will not provide feedback data information. On the other hand, when node B successfully receives the feedback, it ensures the success of the uplink transmission. However, it is difficult to guarantee whether node A can successfully receive the feedback information in the downlink.

In summary, in the OPG, when the policy of any node changes, the policies of the other nodes must remain unchanged. Based on this, we can deduce the conditions for the changes in the utility values of node A and node B, as well as the existence of these conditions:

$$\begin{cases} \text{node A}: \underset{0}{U^1}(a^1, a^2) \to \underset{1}{U^1}(a^{1'}, a^2) \begin{cases} \underset{0}{U^2}(a^2, a^1) \to \underset{0}{U^2}(a^2, a^{1'}), \text{ dosn't exist} \\ \underset{1}{U^2}(a^2, a^1) \to \underset{1}{U^2}(a^2, a^{1'}), \text{ exist} \end{cases} \\ \text{node B}: \underset{0}{U^2}(a^2, a^1) \to \underset{1}{U^2}(a^{2\prime}, a^1) \begin{cases} \underset{1}{U^1}(a^1, a^2) \to \underset{1}{U^1}(a^1, a^{2'}), \text{ dosn't exist} \\ \underset{0}{U^1}(a^1, a^2) \to \underset{0}{U^1}(a^1, a^{2'}), \text{ exist} \end{cases} \end{cases} \tag{A3}$$

For node A, assuming that the policy of node B remains unchanged when the utility value of node A increases from 0 to 1, the utility value of node B also remains unchanged. There are two cases: 0 to 0 or 1 to 1. This means that node A transforms from failure to success in receiving feedback information to success in receiving feedback data. However, when node A succeeds in receiving feedback information, there is no possibility for node B to fail in receiving data. Therefore, it is impossible for node B to have utility values 0 to 0. The same is true for node B. Assuming that the policy of node A does not change when the utility value of node B is increased from 0 to 1, the utility value of node A can go from 0 to 0 or from 1 to 1. This means that node B has changed from receiving data information failure to receiving data information success. However, when node B fails to receive data information, there is no possibility for node A to receive feedback information. Therefore, it is impossible for node A to have utility values from 1 to 1.

The same reasoning applies to node B. Assuming the policy of node A remains unchanged when the utility value of node B increases from 0 to 1, the utility value of node A can transition from 0 to 0 or from 1 to 1. This signifies that node B has transitioned from a failure to receive data information to a successful reception of data information. However, when node B fails to receive data information, it implies that node A will not receive feedback information. Therefore, it is impossible for node A to have utility values of 1 to 1.

Therefore, two cases are supposed to be discussed separately:

(1) When node A $a^1 \to a^{1'}, U^1(a^1, a^2) \to U^1(a^{1'}, a^2)$, a^2 remains unchanged.

The policy of node B is given and denoted as a^2, and the utility value $U^2(a^2, a^1)$ is 1. If we update the policy of node A from a^1 to $a^{1'}$, the utility value of node A increases from 0 to 1. However, the policy of node B remains unchanged, and its utility value remains the same: $U^1(a^{1'}, a^2) - U^1(a^1, a^2) > 0$.

$$\begin{aligned} &P(a^{1'}, a^2) - P(a^1, a^2) \\ &= (U^1(a^{1'}, a^2) + U^2(a^2, a^{1'})) - (U^1(a^1, a^2) + U^2(a^2, a^1)) \\ &= (1+1) - (0+1) \\ &= 2 - 1 \\ &= 1 > 0 \end{aligned} \tag{A4}$$

(2) When node B $a^2 \to a^{2\prime}, U^2(a^2, a^1) \to U^2(a^{2\prime}, a^1))$, a^1 remains unchanged.

The policy of node A is given and denoted as a^1 and the utility value $U^1(a^1, a^2)$ is 0. If we update the policy of node B from a^2 to $a^{2\prime}$, the utility value of node B increases from 0 to 1. However, the policy a^1 of node A remains unchanged, and the utility value is unchanged: $U^2(a^{2\prime}, a^1) - U^2(a^2, a^1) > 0$.

$$\begin{aligned} &P(a^{2\prime}, a^1) - P(a^2, a^1) \\ &= (U^1(a^1, a^{2\prime}) + U^2(a^{2\prime}, a^1)) - (U^1(a^1, a^2) + U^2(a^2, a^1)) \\ &= (0+1) - (0+0) \\ &= 1 - 0 \\ &= 1 > 0 \end{aligned} \tag{A5}$$

□

References

1. Pirayesh, H.; Zeng, H.C. Jamming Attacks and Anti-Jamming Strategies in Wireless Networks: A Comprehensive Survey. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 767–809. [CrossRef]
2. Sharma, H.; Kumar, N.; Tekchandani, R. Mitigating Jamming Attack in 5G Heterogeneous Networks: A Federated Deep Reinforcement Learning Approach. *IEEE Trans. Veh. Technol.* **2023**, *72*, 2439–2452. [CrossRef]
3. Li, W.; Xu, Y.H.; Chen, J.; Yuan, H.C.; Han, H.; Xu, Y.F.; Feng, Z.B. Know Thy Enemy: An Opponent Modeling-Based Anti-Intelligent Jamming Strategy Beyond Equilibrium Solutions. *IEEE Wirel. Commun. Lett.* **2023**, *12*, 217–221. [CrossRef]
4. Aref, M.; Jayaweera, S.; Yepez, E. Survey on cognitive anti-jamming communications. *IET Commun.* **2020**, *14*, 3110–3127. [CrossRef]
5. Zhu, J.W.; Wang, A.Q.; Wu, W.; Zhao, Z.J.; Xu, Y.T.; Lei, R.; Yue, K.Q. Deep-Learning-Based Recovery of Frequency-Hopping Sequences for Anti-Jamming Applications. *Electronics* **2023**, *12*, 496. [CrossRef]
6. Jia, L.L.; Xu, Y.H.; Sun, Y.M.; Feng, S.; Anpalagan, A. Stackelberg Game Approaches for Anti-Jamming Defence in Wireless Networks. *IEEE Wirel. Commun.* **2018**, *25*, 120–128. [CrossRef]
7. Jia, L.L.; Qi, N.; Chu, F.H.; Fang, S.L.; Wang, X.M.; Ma, S.L.; Feng, S. Game-Theoretic Learning Anti-Jamming Approaches in Wireless Networks. *IEEE Commun. Mag.* **2022**, *60*, 60–66. [CrossRef]
8. Kong, L.J.; Xu, Y.H.; Zhang, Y.L.; Pei, X.F.; Ke, M.X.; Wang, X.M.; Bai, W.; Feng, Z.B. A reinforcement learning approach for dynamic spectrum anti-jamming in fading environment. In Proceedings of the International Conference on Communication Technology, Chongqing, China, 8–11 October 2018.
9. Pei, X.F.; Wang, X.M.; Yao, J.N.; Yao, C.H.; Ge, J.C.; Huang, L.Y.; Liu, D.X. Joint time-frequency anti-jamming communications: A reinforcement learning approach. In Proceedings of the International Conference on Wireless Communications and Signal Processing, Xi'an, China, 23–25 October 2019.
10. Liu, X.; Xu, Y.H.; Jia, L.L.; Wu, Q.H.; Anpalagan, A. Anti-Jamming Communications Using Spectrum Waterfall: A Deep Reinforcement Learning Approach. *IEEE Wirel. Commun. Lett.* **2018**, *22*, 998–1001. [CrossRef]
11. Chang, X.; Li, Y.B.; Zhao, Y.; Du, Y.F.; Liu, D.H. An Improved Anti-Jamming Method Based on Deep Reinforcement Learning and Feature Engineering. *IEEE Access* **2022**, *10*, 69992–70000. [CrossRef]
12. Liu, S.Y.; Xu, Y.F.; Chen, X.Q.; Wang, X.M.; Wang, M.; Li, W.; Li, Y.Y.; Xu, Y.H. Pattern-Aware Intelligent Anti-Jamming Communication: A Sequential Deep Reinforcement Learning Approach. *IEEE Access* **2019**, *7*, 169204–169216. [CrossRef]
13. Li, W.; Qin, Y.; Feng, Y.B.; Han, H.; Chen, J.; Xu, Y.H. "Advancing Secretly by an Unknown Path": A Reinforcement Learning-Based Hidden Strategy for Combating Intelligent Reactive Jammer. *IEEE Wirel. Commun. Lett.* **2022**, *11*, 1320–1324. [CrossRef]
14. Han, H.; Xu, Y.F.; Jin, Z.; Li, W.; Chen, X.Q.; Fang, G.; Xu, Y.H. Primary-User-Friendly Dynamic Spectrum Anti-Jamming Access: A GAN-Enhanced Deep Reinforcement Learning Approach. *IEEE Wirel. Commun. Lett.* **2022**, *11*, 258–262. [CrossRef]
15. Li, Y.Y.; Xu, Y.H.; Xu, Y.T.; Liu, X.; Wang, X.M.; Li, W.; Anpalagan, A. Dynamic Spectrum Anti-Jamming in Broadband Communications: A Hierarchical Deep Reinforcement Learning Approach. *IEEE Wirel. Commun. Lett.* **2020**, *9*, 1616–1619. [CrossRef]
16. Xu, Y.F.; Xu, Y.H.; Dong, X.; Ren, G.C.; Chen, J.; Wang, X.M.; Jia, L.L.; Ruan, L. Convert Harm into Benefit: A Coordination-Learning Based Dynamic Spectrum Anti-Jamming Approach. *IEEE Trans. Veh. Technol.* **2020**, *69*, 13018–13032. [CrossRef]
17. Jia, L.L.; Xu, Y.H.; Sun, Y.M.; Feng, S.; Yu, L.; Anpalagan, A. A Game-Theoretic Learning Approach for Anti-Jamming Dynamic Spectrum Access in Dense Wireless Networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 1646–1656. [CrossRef]
18. Xiao, L.; Li, Y.; Liu, J.J.; Zhao, Y.F. Power control with reinforcement learning in cooperative cognitive radio networks against jamming. *J. Supercomput.* **2015**, *71*, 3237–3257. [CrossRef]
19. Ibrahim, K.; Ng, S.X.; Qureshi, I.M.; Malik, A.N.; Muhaidat, S. Anti-Jamming Game to Combat Intelligent Jamming for Cognitive Radio Networks. *IEEE Access* **2021**, *9*, 137941–137956. [CrossRef]
20. Yao, F.Q.; Jia, L.L. A Collaborative Multi-Agent Reinforcement Learning Anti-Jamming Algorithm in Wireless Networks. *IEEE Wirel. Commun. Lett.* **2019**, *8*, 1024–1027. [CrossRef]
21. Elleuch, I.; Pourranjbar, A.; Kaddoum, G. A Novel Distributed Multi-Agent Reinforcement Learning Algorithm Against Jamming Attacks. *IEEE Commun. Lett.* **2021**, *25*, 3204–3208. [CrossRef]
22. Ororbia, M.E.; Warn, G.P. Design Synthesis Through a Markov Decision Process and Reinforcement Learning Framework. *J. Comput. Inf. Sci. Eng.* **2021**, *22*, 021002. [CrossRef]
23. Zhang, Y.L.; Xu, Y.H.; Xu, Y.T.; Yang, Y.; Luo, Y.P.; Wu, Q.H.; Liu, X. A Multi-Leader One-Follower Stackelberg Game Approach for Cooperative Anti-Jamming: No Pains, No Gains. *IEEE Commun. Lett.* **2018**, *22*, 1680–1683. [CrossRef]
24. Cao, H.J.; Cai, J. Distributed Opportunistic Spectrum Access in an Unknown and Dynamic Environment: A Stochastic Learning Approach. *IEEE Trans. Veh. Technol.* **2018**, *67*, 4454–4465. [CrossRef]
25. Zheng, J.C.; Zhang, H.G.; Cai, Y.M.; Li, R.P.; Anpalagan, A. Game-Theoretic Multi-Channel Multi-Access in Energy Harvesting Wireless Sensor Networks. *IEEE Sens. J.* **2016**, *16*, 4587–4594. [CrossRef]
26. Du, Y.W.; Gong, J.H.; Wang, Z.M.; Xu, N. A Distributed Energy-Balanced Topology Control Algorithm Based on a Noncooperative Game for Wireless Sensor Networks. *Sensors* **2018**, *18*, 4454. [CrossRef]
27. Shen, L.H.; Feng, K.T.; Hanzo, L. Five Facets of 6G: Research Challenges and Opportunities. *ACM Comput. Surv.* **2023**, *55*, 1–39. [CrossRef]

28. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.; Veness, J.; Bellemare, M.; Graves, A.; Riedmiller, M.; Fidjeland, A.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
29. He, K.M.; Sun, J. Convolutional neural networks at constrained time cost. In Proceedings of the International Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.