IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 16 July 2024, accepted 4 August 2024, date of publication 8 August 2024, date of current version 19 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3440324

RESEARCH ARTICLE

Intelligent Jamming Strategy for Wireless Communications Based on Game Theory

YONGCHENG LI[1], WEIJIAN MIAO[2], ZHENZHEN GAO[2], AND GANGMING LV[2]
[1]State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System (CEMEE), Luoyang 471003, China
[2]School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an 710049, China

Corresponding author: Zhenzhen Gao (zhenzhengao@xjtu.edu.cn)

This work was supported in part by the Open Research Found of CEMEE under Grant 2023K0201, and in part by the National Natural Science Foundation of China under Grant 62071367.

ABSTRACT This paper explores the scenario that a jammer attacks an intelligent transmitter which can sense and adapt to the jamming environment. Given the non-cooperative relationship between the transmitter and the jammer, two main challenges are addressed in this paper: how to model their interactions and how to devise a jamming strategy without prior knowledge of the transmitter. A non-zero-sum game is used to model and analyze such non-cooperative interactions. An approximate mixed-strategy Nash equilibrium (NE) under complete information is derived to serve as a benchmark for comparison. According to the non-zero-sum game model, a Deep Q-Network (DQN) approach is proposed to determine jamming strategies by exploiting the detection results of the legitimate signals, such as Acknowledgements (ACKs) feedback and the modulation recognition results obtained by the jammer. Simulation results demonstrate that, without requiring complete information about the transmitter, the proposed DQN approach can achieve a similar utility as the benchmark strategy using complete information. Compared to other learning-based jamming schemes and random jamming strategy, the proposed DQN approach achieves a higher packet error rate for the communication transceiver with reduced jamming power consumption.

INDEX TERMS Intelligent jamming, game theory, deep Q-network, wireless communication.

I. INTRODUCTION

Due to broadcast nature, wireless communications are susceptible to adversary jamming attacks. In electronic warfare scenarios, the development of advanced wireless technologies, such as cognitive radio [1], have increased the difficulty of successful jamming attacks. Traditional jamming techniques often relied on expert experience for strategy-making or directly used high power to suppress victim communication systems in order to block communication transmissions [2], [3]. However, with the development of anti-jamming technologies, these traditional techniques have become less effective and even raise the risk of the jammer's position being detected by the opponent.

In previous studies, optimal power allocation strategies for a jammer with an average power constrained operating on additive noise channels have been investigated [4]. Based on this, the optimal jamming signal against digital modulation has been studied without considering the transmitter's anti-jamming techniques. In [5] and [6], the authors explored the jamming performance when the modulation scheme and power of the communication signals were given. An optimization problem was formulated to maximize the average probability of error at the receiver. These studies investigated the jamming optimization problem for some given communication signals. Obviously, in the aforementioned studies, the transmitter does not possess intelligence.

If the transmitter can adaptively adjust its transmission strategies to combat the jammer, Game theory is a powerful and natural framework to represent such interactions between them, where each of them seeks to maximize its own utility. In the event that the sum of their utilities is not equal to zero, the game is designated as a non-zero-sum game. In this scenario, the transmitter seeks to maximize transmission performance of its own while the jammer aims to degrade transmission performance of communication [7],

The associate editor coordinating the review of this manuscript and approving it for publication was Moussa Ayyash.

[8], [9]. Nash equilibria (NE) have been derived based on the assumption of complete information, where players know each other's objectives and actions, including channel characteristics, power levels, and system parameters [10], [11]. In the cases of incomplete information, such as uncertainties about player actions, Bayesian games provide a framework for modeling player interactions to maximize expected payoffs [12], [13]. However, Bayesian games face challenges when players in the jamming game lack a prior distribution over the parameters of interest.

Given the non-cooperative nature of the jammer and the legitimate transceiver, prior information about the opponent is usually not available. How to formulate the jamming utility and how to solve the non-cooperative jamming game in such situations are challenging problems. Therefore, this study aims to explore the problem of jamming strategy-making for non-cooperative games without prior information.

A. RELATED WORK

Many studies have been done about anti-jamming [14], [15], [16], [17], where intelligent transmitters are considered and complete or partial information about the jamming strategy is assumed to be available. Compared to abundant research on anti-jamming schemes, fewer studies have been done on jamming strategies from the jammer's perspective. We have listed existing related work on jamming strategies in Table 1.

Specifically, with complete information or knowledge of the prior distribution, the authors in [7] and [18] build a jamming game with the goal of minimizing the utility of the communication. Some authors consider the establishment of jamming games under incomplete information, where the utility function is constructed using the SNR at the receiver [10], [19], [20]. Then, the strategy-making problem of the jamming game is solved by optimizing the channel selection and power allocation. Finally, the equilibrium strategy of the jammer in the game can be derived using some specific methodologies. However, these jamming strategies are solved by requiring complete information or some prior knowledge about the communication transceiver or signal processing results at the receiver, such as channel state information and SINR. These are difficult to obtain in a non-cooperative scenario. Therefore, the existing game-based methods are difficult to cope with unknown environments with non-cooperative players.

Reinforcement learning (RL) theory, which requires no prior knowledge, has been applied in electronic warfare scenarios recently. As a branch of machine learning, RL uses a trial-error mechanism to learn actions in unknown environments and has been extensively studied in jamming strategy-making [21], [22], [23], [24], [25], [26], [27], [28], [29]. A considerable part of the existing research focuses on the field of radar jamming strategy-making [21], [22], [23]. In the field of communication jamming, a deep competitive double-Q jamming strategy-making network integrating action elimination and a jamming strategy-making method based on Q-Learning (QL) have been proposed in [24] and [25] respectively. Authors in [26] used a strategy-making algorithm based on clustering and RL to learn the optimal jamming strategy through continuous interaction with the environment. In [27], a learning-based algorithm has been used to explore the frequency changes of the communication transmitter and then implement precise jamming. Based on [5] and [6], authors in [28] used a multi-armed gambling machine to find the optimal jamming parameters for digital communications when the transmitter adopts a fixed transmission strategy. For a wireless network with multiple legitimate nodes, the authors proposed a beamforming attack strategy based on RL to optimize the direction and angle width of the jamming beam [29]. However, these learning-based jamming strategy-making methods mainly consider the intelligence and learning ability of the jammer.

In general, if the intelligence of the transmitter is considered, existing jamming schemes typically rely on certain prior information about the opponent, such as the SNR feedback, channel characteristics, or transmitter's strategies, to formulate the game model. When no prior information is available, RL is used in existing schemes to enable the intelligent jammer to learn the unknown environment, but the transmitter's intelligence has not been not considered yet.

B. CONTRIBUTIONS

Different from the existing learning-based jamming schemes in [5], [6], [24], [25], [26], [27], [28], and [29], the transmitter is considered intelligent in this paper. When the intelligent transmitter can adjust its transmission adaptively and no prior information can be obtained, how to optimize the jamming strategy? We try to find a solution to this problem in this paper. Considering the non-cooperation of the jammer and the transmitter, who both have the ability to adjust their transmission strategies, we formulate their interactions as a non-zero-sum game. By exploiting the sensing results of the communication signals, the jammer's utility is formulated without prior information of the legitimate communications. The contributions of the paper are as follows.

- The jamming game is modeled as a non-zero-sum game. In the game, the transmitter can adaptively adjust the signal modulation scheme and transmission power to transmit more bits successfully per unit of power. Simultaneously, the jammer endeavors to adjust its jamming signal modulation, jamming power, and jamming time ratio to hinder the receiver from accurately receiving bits while minimizing the jamming power.
- A DQN-based jamming strategy-making method is proposed to solve the jamming game without prior information about the opponent. Based on the game model, a reward is designed for the DQN approach by exploiting the detection of Acknowledgements (ACKs) feedback.
- The proposed intelligent method can approach the jamming utility provided by the approximate mixed-strategy

TABLE 1. Existing related work on jamming strategies.

Ref.	Methods	Decision Domains	Require Priori Information of Communication?	Is Transmitter Intelligent?
[2]	Expert Experience	Frequency	No. Expert information is required	Yes
[3]	Expert Experience	UAVs allocation	No. Expert information is required	Yes
[4]	Optimization theory	Power	Yes. Probability density function of signal is required	No
[5], [6]	Optimization theory	Power; Time	No. But need power and modulation information	No
[7]	Game theory	Power; Frequency	Yes. Distribution probability of the transmitter's type and SINR are required	Yes
[10]	Game theory	Frequency	Yes. Channel information and SINR are required	Yes
[18]	Game theory	Frequency	Yes. Channel information are required	Yes
[19]	Game theory	Power	Yes. Channel information and SINR are required	Yes
[20]	Game theory	Power	Yes. Channel information and SINR are required	Yes
[24]	Reinforcement Learning	Frequency	No. But need feedback: ACKs	No
[25]	Reinforcement Learning	Frequency	No. But need feedback: ACKs	No
[26]	Reinforcement Learning	Spatial	Yes. Locations of communication nodes are required	No
[27]	Reinforcement Learning	Frequency	No. But need feedback: ACKs	No
[28]	Reinforcement Learning	Power; Time	No. But need feedback: ACKs	No
[29]	Reinforcement Learning	Spatial	No. But need the change in the observed channel busy times or ACKs	No
This Work	Game theory and Reinforcement Learning	Power, Time, Modulation	No. But need the feedback: ACKs	Yes

NE under complete information. Compared to existing schemes, the proposed method achieves a higher packet error rate with lower jamming power consumption.

C. ORGANIZATION

The rest of this paper is organized as follows. Section II introduces the system model and game model. The approximate mixed-strategy NE of the non-zero-sum game is solved under the assumption of complete information using the Particle Swarm Optimization (PSO) algorithm in Section III. In Section IV, the DQN-based jamming strategy-making scheme is proposed. Section V shows the simulation parameters, comparison schemes, simulation results, and discussions. Finally, Section VI concludes this paper. The abbreviations used in this paper are listed in Table 2.

II. SYSTEM MODEL AND GAME FORMULATION

A. SYSTEM MODEL

Considering a communication transceiver that is faced with an intentional jammer, as shown in Figure 1, where the transmissions from the transmitter (S) to the receiver (D) are jammed by the jammer (J). In [28], the optimal jamming signal has been designed when the transmitter is unaware of the jammer, but the jammer knows the modulation scheme of the transmitted signals. In this paper, we consider the case where both S and J are aware of each other's existence. When J is able to adaptively adjust the jamming power, the modulation scheme, and the jamming time ratio, S can also use different digital modulation schemes and adjust the transmit power to combat the jamming.

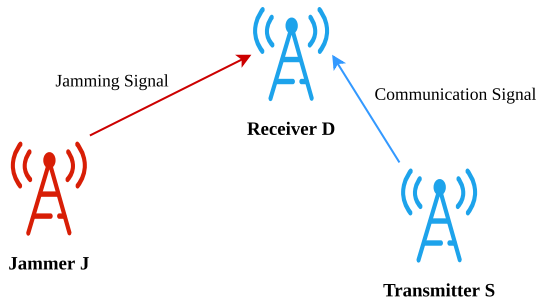


FIGURE 1. System model with an intentional jammer.

The set of modulation schemes used at S is represented by $\mathcal{C} = \{C_1, C_2, \cdots, C_M\}$ and M is the total number of modulation schemes. The available transmit power set of S

TABLE 2. List of abbreviations.

Abbreviations	Description
DQN	Deep Q-Network
ACK/NACK	Acknowledgement/Negative Acknowledgement
PER	Packet Error Rate
SER	Symbol Error Rate
SINR	Signal-to-Interference plus Noise Ratio
SNR	Signal-to-Noise Ratio
JNR	Jamming-to-Noise Ratio
NE	Nash Equilibrium
PSO	Particle Swarm Optimization
BPSK	Binary Phase Shift Keying
QPSK	Quadrature Phase Shift Keying
M-QAN	M-ary Quadrature Amplitude Modulation
DRL	Deep Reinforcement Learning
MDP	Markov Decision Process
QL	Q Learning
CSI	Channel State Information
LTE	Long Term Evolution
PSS	Primary Synchronization Signal
SSS	Secondary Synchronization Signal
FCN	Fully Connected Neural Network
PERT	Prioritized Experience Replay Technique
RERT	Random Experience Replay Technique
QiERT	Quantum-inspired Experience Replay Technique

is $\mathcal{P}_S = \{P_S^1, P_S^2, \cdots, P_S^{L_S}\}$ and L_S is the maximum level of transmit power. For J, the set of modulation schemes is $\mathcal{J} = \{J_1, J_2, \cdots, J_N\}$ and N is the total number of jamming modulation schemes. The available jamming power set is $\mathcal{P}_\mathcal{J} = \{P_J^1, P_J^2, \cdots, P_J^{L_J}\}$ and L_J is the maximum level of jamming power. The set of jamming time ratios is denoted as $\Lambda = \{\lambda_1, \lambda_2, \cdots, \lambda_R\}$, where R is the maximum level of the jamming time ratio and $\lambda_r \in (0, 1], r = 1, 2, \cdots, R$. The discretization of transmit power, jamming power, and jamming time ratio is intended to accommodate practical strategy-making methods for rapid implementation of effective strategies [5], [6], [24], [25], [28]. During the process of electronic countermeasures, S and J adjust their strategies independently.

B. JAMMING SIGNAL

Referring to the signal model proposed in [5], the low pass equivalent of the transmission signal $s(t)$ can be represent as $s(t) = \sum_{-\infty}^{\infty} \sqrt{P_S} s_k g(t - kT)$, where $P_S \in \mathcal{P}_S$ is the average signal power, $g(t)$ is the real valued pulse shape, and T is the symbol interval. s_k denotes the symbol coming from one of the possible constellations in $\mathcal{C}$.

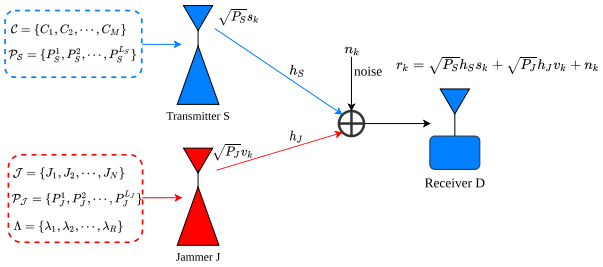


FIGURE 2. Signal model of the jamming process.

The jamming signal process is clearly shown in Figure 2. It is assumed that $s(t)$ passes through the wireless channel while being attacked by a jamming signal represented as $v(t) = \sum_{-\infty}^{\infty} \sqrt{P_J} v_k g(t - kT)$, where P_J is the jamming signal power and v_k denotes the jamming symbol coming from one of the possible constellations in $\mathcal{J}$. If there is a coherent receiver and perfect synchronization, the received signal after matched filtering and sampling at the symbol intervals is given by $r_k = \sqrt{P_S} h_S s_k + \sqrt{P_J} h_J v_k + n_k$, where h_S and h_J denote the wireless channels from D to S and J, respectively. n_k is the zero-mean additive white Gaussian noise with variance σ^2. Let $\lambda \in \Lambda$ denote the jamming time ratio utilized by J. It has been proved in [5] that, J will concentrate its energy to jam the communication signal for a portion of the time to achieve better jamming effects when J has limited jamming energy, i.e., given a jamming time ratio λ and power P_J, J transmits the jamming signal with power P_J/λ.

C. GAME FORMULATION

When both S and J are intelligent, meaning that they can sense the environment and adaptively adjust their strategies to maximize their own utility, it is crucial to model the confrontation between them. Game theory, with its powerful concept of equilibrium, is a natural tool to explore such a problem. The equilibrium, if it exists, represents a strategic point where both S and J would like to stick to, as deviating from this equilibrium strategy would result in diminished utility for either party [30], [31].

The confrontation between S and J can be modeled as a non-cooperative game. In the game, S focuses on successfully transmitting more bits per unit of power by choosing an appropriate modulation scheme and power level. On the contrary, J aims to reduce the number of correct bits received by D while reducing its own energy consumption. Specifically, J selects the appropriate modulation scheme, power level, and jamming time ratio to achieve its intent.

When J knows the modulation scheme and signal power of S, the existing literature [5] provides the corresponding formula to calculate the error probability at the receiver. Assuming that the victim communication signal is an M-QAM signal, the error probability of the M-QAM victim signal affected by the jamming signal v along any signal

dimension is given by Eq. (1), as shown at the bottom of the page. d_{min} is the minimum distance of the underlying modulation scheme [32], M is the order of the constellation. SNR is the received signal-to-noise ratio at D, which can be calculated as $\frac{G_{h_S}P_S}{\sigma^2}$, where $G_{h_S} = |h_S|^2$. JNR is the received jamming-to-noise ratio at D, which can be calculated as $\frac{G_{h_J}P_J}{\sigma^2}$, where $G_{h_J} = |h_J|^2$.

Considering the influence of jamming time ratio λ, according to [5] and [6], the overall symbol error rate ser is given as follows.

$$ser = \lambda p_e(j, \text{SNR}, \frac{\text{JNR}}{\lambda}) + (1-\lambda)p_e(j, \text{SNR}, 0) \quad (2)$$

After calculating the SER of the jammed communication signal, the packet error rate per can be further obtained according to the relationship between per and ser which is shown in Eq. (3), where N_{sym} is the number of symbols in one packet.

$$per = 1 - (1-ser)^{N_{sym}} \quad (3)$$

In actual communication systems, certain error correction coding schemes have been used to enhance the reliability of the system. Considering the error correction capability of the communication system, the tolerable SER threshold α is set. It means that a packet is considered to be received correctly if at most αN_{sym} symbols in a packet are received in error at D. Thus, Eq. (3) is rewritten as follows.

$$per = 1 - \sum_{k=1}^{N_{th}} \binom{N_{sym}}{k} ser^k (1-ser)^{N_{sym}-k} \quad (4)$$

where $N_{th} = \alpha N_{sym}$ indicates the number of incorrect symbols. The summation term in the expression of per represents the probability that the packet is received correctly.

Then, the number of accurately decoded symbols in the correctly transmitted packets is $N(1-per)(1-ser)N_{sym}$. If the number of bits contained in a symbol is denoted by n_{bit}, the number of corrected decoded bits of the N packets is $N(1-per)(1-ser)N_{sym}n_{bit}$. Taking the power cost into consideration, the utility of S can be defined as

$$\mu_S = \frac{N(1-per)(1-ser)N_{sym}n_{bit}}{P_S^i} \quad (5)$$

where P_S^i is the signal power used by S and $i \in \{1, \cdots, L_S\}$. The power consumption of S is normalized by the minimum transmit power P_S^1. Besides, the number of successfully transmitted bits is normalized by the total number of symbols inside the N packets, which is NN_{sym}. Therefore, the utility of S defined in Eq.(5) can be expressed as

$$\mu_S = \frac{(1-per)(1-ser)n_{bit}}{f(P_S^i, P_S^1)} \quad (6)$$

where $f(P_S^i, P_S^1) = \frac{P_S^i}{P_S^1}$. Once the transmission strategy of S is selected, the values of per and ser are affected by the jamming strategy of J.

Given the modulation scheme set $\mathcal{C}$ and the transmit power set $\mathcal{P}_S$, S will choose an action pair (C_m, P_S^i), where $m \in \{1, \cdots, M\}$ and $i \in \{1, \cdots, L_S\}$, to maximize its utility μ_S.

$$\max_{C_m \in \mathcal{C}, P_s^i \in \mathcal{P}_S} \mu_S \quad (7)$$

On the contrary, J tries to minimize the number of successfully transmitted bits with less jamming power, thus the utility of J can be defined as $-P_J^j N(1-per)(1-ser)N_{sym}n_{bit}$, where $P_J^j \in \mathcal{P}_\mathcal{J}$ is the jamming signal power and $j \in \{1, L_J\}$. Similarly, the utility is normalized by the total number of symbols NN_{sym} and the maximum jamming power $P_J^{L_J}$. So, the utility of J can be written as

$$\mu_J = \frac{-(1-per)(1-ser)n_{bit}}{f(P_J^{L_J}, P_J^j)} \quad (8)$$

where $f(P_J^{L_J}, P_J^j) = \frac{P_J^{L_J}}{P_J^j}$ represents the normalized jamming power. Given the strategy of S, J can influence per and ser by choosing jamming strategies to increase its utility.

Given the modulation scheme set $\mathcal{J}$, the jamming power set $\mathcal{P}_\mathcal{J}$, and the jamming time ratio set Λ, J can choose an action triple (J_n, P_J^j, λ_r), where $n \in \{1, \cdots, N\}$, $j \in \{1, \cdots, L_J\}$, and $r \in \{1, \cdots, R\}$, to maximize its utility μ_J.

$$\max_{J_n \in \mathcal{J}, P_J^j \in \mathcal{P}_\mathcal{J}, \lambda_r \in \Lambda} \mu_J \quad (9)$$

The confrontation between S and J is modeled as a non-zero-sum game in a non-cooperative situation. The utilities μ_S and μ_J depend on the actions of J and S, i.e., power, modulation scheme, and jamming time ratio. They can proactively adjust their strategies to increase their utilities. If S and J can obtain perfect information about each other, a mixed-strategy NE can be achieved. Although S and J can hardly obtain complete information about the opponent in an electronic warfare scenario, we will solve the mixed-strategy NE under the assumption of complete information so that it can work as a benchmark.

III. MIXED STRATEGY NASH EQUILIBRIUM UNDER COMPLETE INFORMATION

In the complete information game, the strategic information of both players is public knowledge. In this case, S and J need to know each other's perfect information, such as signal power and CSI [33]. When opponents' information is available, in a game with a finite number of players and a finite-size set of strategies, there exists at least one mixed-strategy NE [30], [31]. The specific mixed strategy can be

$$p_e(v, \text{SNR}, \text{JNR}) \approx \frac{1}{2}\left(1 - \frac{1}{\sqrt{M}}\right)\left[\text{erfc}\left(\sqrt{\text{SNR}}\frac{d_{min}}{2} + \sqrt{\text{JNR}}v\right) + \text{erfc}\left(\sqrt{\text{SNR}}\frac{d_{min}}{2} - \sqrt{\text{JNR}}v\right)\right] \quad (1)$$

obtained through linear programming, numerical iteration, or optimization algorithms. According to the adjustable action sets of J and S in Section II, the mixed strategy of J is $\boldsymbol{p}$ which is expressed as

$$\boldsymbol{p} = (p_1, p_2, \cdots, p_j, \cdots, p_{N \times L_J \times R}) \tag{11}$$

where p_j represents the probability that J chooses the j-th action from the action space $\mathcal{J} \times \mathcal{P}_{\mathcal{J}} \times \Lambda$ with $\times$ representing the Cartesian product. Similarly, the mixed strategy $\boldsymbol{q}$ of S can be given as

$$\boldsymbol{q} = (q_1, q_2, \cdots, q_i, \cdots, q_{M \times L_S}) \tag{12}$$

where q_i represents the probability that S chooses the i-th action selected from the action space $\mathcal{C} \times \mathcal{P}_{\mathcal{S}}$.

Among all mixed strategies, there is a pair of mixed strategies $(\boldsymbol{p}^*, \boldsymbol{q}^*)$ forming a NE, then the mathematical expression of the NE can be written as

$$\mu_J(\boldsymbol{p}^*, \boldsymbol{q}^*) \geq \mu_J(\boldsymbol{p}, \boldsymbol{q}^*) \tag{13}$$

$$\mu_S(\boldsymbol{p}^*, \boldsymbol{q}^*) \geq \mu_S(\boldsymbol{p}^*, \boldsymbol{q}) \tag{14}$$

According to the above description, both J and S have the perfect CSI and strategy information of each other. By referring to the payoff matrix of the confrontation, the mixed strategy NE of the game can be found. To address the game problem, we utilize the PSO method to find the NE [34]. To attain the NE in the optimization outcome, it is essential to appropriately design the fitness function of the PSO algorithm. In the case of complete information, the payoff matrices of J and S can be represented as $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively. The fitness function can be set as Eq. (10), shown at the bottom of the page, based on [34]. Similar designs can also be found in [35] and [36]. Theoretically speaking, if the fitness function achieves a minimum value of 0, the PSO algorithm can provide the mixed strategy NE solution. The steps of the PSO algorithm are briefly outlined below.

1) INITIALIZE PARTICLE SWARM

The size of the particle swarm is N_p and the maximum number of iterations is g_{max}. Then, the position and velocity of the particle swarm are randomly initialized. The position vector $(\boldsymbol{p}, \boldsymbol{q})$ of each particle satisfies the condition in Eq. (15) and each component of the velocity is located in $[-1, 1]$. Then, we make the current number of iterations $t = 1$.

$$\begin{cases} \sum p_j = 1, & p_j \geq 0 \\ \sum q_i = 1, & q_i \geq 0 \end{cases} \tag{15}$$

2) FITNESS FUNCTION CALCULATION

The fitness function value of the particle can be calculated according to Eq. (10). The individual extreme value $P_{best}(i)$ and global extreme value G_{best} of the particle can be found based on the fitness function value of the particle.

3) UPDATE AND NORMALIZATION

Firstly, the inertia weight ω is calculated according to Eq. (16) for subsequent needs.

$$\omega = \omega_{max} - t\frac{\omega_{max} - \omega_{min}}{g_{max}} \tag{16}$$

Then, we can update the particle velocity and position vector according to Eq. (17) and (18). The position vector of each particle is normalized in turn, so that the position of each particle can be always in the feasible mixed strategy combination space.

$$V_i^{t+1} = \omega V_i^t + c_1 r_1 (P_{best}(i) - Z_i^t) + c_2 r_2 (G_{best} - Z_i^t) \tag{17}$$

$$Z_i^{t+1} = Z_i^t + V_i^{t+1} \tag{18}$$

Among them, V_i^t is the speed of the i-th particle at the t-th iteration, Z_i^t is the position of the i-th particle at the t-th iteration. $P_{best}(i)$ is the optimal solution currently found by the i-th particle. G_{best} is the optimal solution currently found by the entire population. ω is the inertia weight, r_1, r_2 are random numbers between 0 and 1 and c_1, c_2 are learning factors.

After updating the particle speed and position, the position vector of each particle needs to be normalized in turn so that the position of each particle belongs to the feasible mixed strategy space.

4) TERMINATION AND OUTPUT

The particle's individual extreme value $P_{best}(i)$ and global extreme value G_{best} are updated. If the number of iterations reaches g_{max} or G_{best} reaches the accuracy requirement that the fitness value function value $f(G_{best})$ is less than the specified threshold, the algorithm will stop and the optimal particle G_{best} is output, which is the approximate NE solution. Otherwise, return to Step 2 and calculate the value of fitness function.

IV. INTELLIGENT JAMMING STRATEGY-MAKING SCHEME WITHOUT PRIOR INFORMATION

A. PROCESS ANALYSIS

Due to the non-cooperative nature of J and S, it is challenging for S and J to obtain each other's information. Many existing anti-jamming schemes formulate the utility based on the worst-case assumption that the received SNR/JNR at the legitimate receiver is accessible to J. However, it is problematic for J to acquire SNR/JNR at D when attempting to optimize its jamming strategy. The utilities designed in Eq. (6) and (8) cannot work directly in this case.

$$f(\boldsymbol{p}, \boldsymbol{q}) = \max(\max_i(\boldsymbol{A}(i,:)\boldsymbol{p}^{\boldsymbol{T}} - \boldsymbol{q}\boldsymbol{A}\boldsymbol{p}^{\boldsymbol{T}}), 0) + \max(\max_j(\boldsymbol{q}\boldsymbol{B}(:,j) - \boldsymbol{q}\boldsymbol{B}\boldsymbol{p}^{\boldsymbol{T}}), 0) \tag{10}$$

In the considered system of Section II, the reliability of the communication between S and D is ensured by the ACKs feedback mechanism. The ACKs feedback can be received by both S and J. Although per and ser in the utilities cannot be derived from p_e in Eq. (1) without prior information about S, J can monitor the ACKs feedback sent by D and make an approximate estimation of per. Furthermore, J can derive ser by using Eq. (4). In this way, it is possible for J to derive the correctly transmitted bits in Eq. (8). Similarly, S can also estimate per and ser through the ACKs feedback from D. A decrease in the number of ACKs indicates a higher error rate at D. In response, S needs to increase the transmit power or reduce the modulation order to ensure reliable communication. Otherwise, J needs to increase the jamming power and adjust the modulation or jamming time ratio to enhance jamming effectiveness.

Both J and S use the ACKs feedback to estimate their utilities and then adjust their strategies. It means that they do not need to know each other's perfect information, such as signal power and CSI. During the process, they engage in continuous interaction to get better strategies. RL happens to be a method that learns by interacting with the environment and adjusts strategy according to the feedback signals or rewards from the environment. QL and DQN are classical algorithms in RL. However, QL converges slowly when learning in high-dimensional complex space. Compared with QL, DQN uses neural networks to estimate the Q-value function, which has more advantages in dealing with strategy-making problems of high-dimensional complex spaces. So far, the DQN algorithm has been used to solve the anti-jamming problem [37], [38], [39].

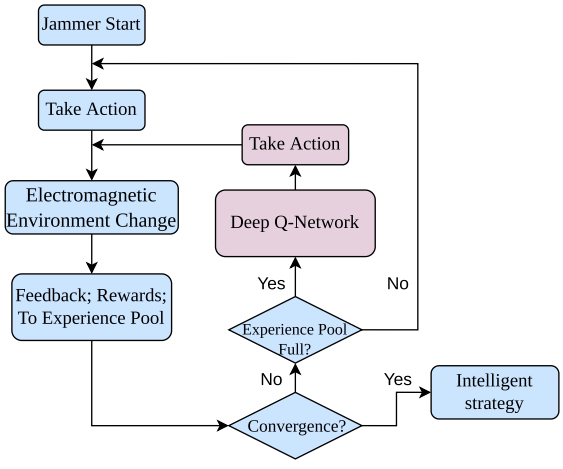


FIGURE 3. Jamming strategy-making process.

On this basis, an intelligent jamming strategy-making method is proposed, its specific execution process is shown in Figure 3. Both the transmitter and the jammer use the DQN to learn better strategies when the experience pool is full. This solution is based on the non-zero-sum game in Section II and the rewards of the strategy-making network originates from the utilities designed in Eq. (6) and (8). The details of the proposed method are given as follows.

B. THE MDP FORMULATION AND RL ELEMENT SETTINGS

According to the system model described in the previous sections, the strategy-making in the non-cooperative confrontation between S and J is analogous to the state transition process. To move to the next state, both S and J choose an action in the current state. Next state is only associated with the current state and action. Therefore, the DQN method is designed on an MDP, which is defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R})$, where $\mathcal{S}$, $\mathcal{A}$, and $\mathcal{R}$ are the state space, the action space, and the immediate reward of the network, respectively. In the specific process, the system time is divided into K time slots and each slot is indexed by $k \in \{1, \cdots, k, \cdots, K\}$. A time slot is the smallest unit for the transmitter and the jammer to take an action.

- **State** $\mathcal{S}$: The states encompass the historical actions and environmental observations of J and S. We define the states of them in the k-th time slot as s_J^k, s_S^k, which mainly describe the current environment of J and S. Specifically, the state s_J^k of J can be recorded as

$$s_J^k = (N_{ACKs}^{k-w}, C^{k-w}, a_J^{k-w}, \cdots, N_{ACKs}^{k-1}, C^{k-1}, a_J^{k-1}) \tag{19}$$

  where N_{ACKs}^k and C^k are the number of ACKs and the modulation scheme of the communication signal, which can be detected and identified by J. a_J^k denotes the action of J at time slot k.
  Similarly, the state s_S^k of S can be written as

$$s_S^k = (N_{ACKs}^{k-w}, J^{k-w}, a_S^{k-w}, \cdots, N_{ACKs}^{k-1}, J^{k-1}, a_S^{k-1}) \tag{20}$$

  where J^k, a_S^k represents the modulation scheme of the jamming signal and the action of S at time slot k. There is a parameter w in Eq. (19) and (20), which denotes the length of previous observations or actions. Different values of w have a certain impact on the results, which will be discussed in the subsequent content.
- **Action** $\mathcal{A}$: As mentioned in Section II-A, J tries to select the jamming modulation scheme, jamming power and jamming time ratio to degrade the communication between S and D. In the k-th time slot, the action of J is denoted as $a_J^k = (J_n, P_J^j, \lambda_r)$, which belongs to the action space $\mathcal{J} \times \mathcal{P}_{\mathcal{J}} \times \Lambda$. The dimension of a_J^k is 3 and the size of the action space of J is RNL_J, it means that there are RNL_J choices. In the same way, S adjusts the modulation scheme and the transmit power to maintain its quality of communication. In the k-th time slot, the action of S is denoted as $a_S^k = (C_m, P_S^i)$, which belongs to the action space $\mathcal{C} \times \mathcal{P}_{\mathcal{S}}$. The dimension of a_S^k is 2 and the size of the action space of S is ML_S, namely, S has ML_S choices to send the signal.
- **Reward** $\mathcal{R}$: In an electronic warfare scenario, in order to solve the jamming game in Section II-C, the utility

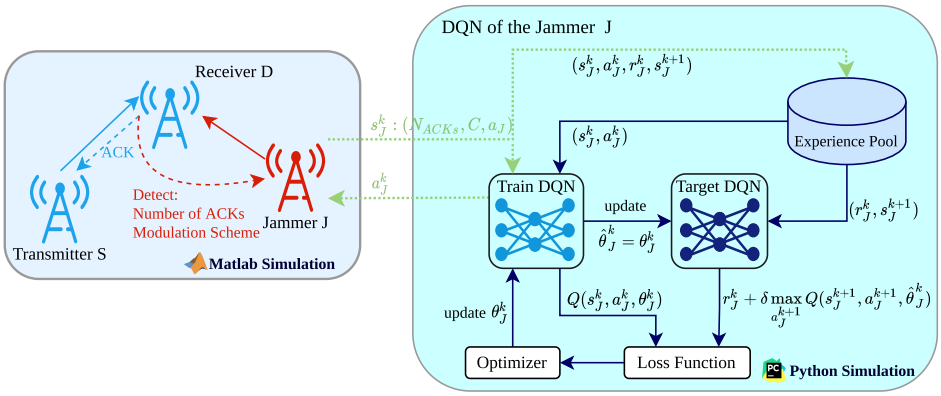


FIGURE 4. The DQN-based jamming strategy-making process in k-th time slot.

of J is used as the reward. The DQN algorithm is used to find the optimal jamming strategy to maximize long-term average utility of J. Therefore, the reward in the k-th time slot of J is defined as

$$r_J^k = \frac{-(1-p\hat{e}r_k)(1-s\hat{e}r_k)\hat{n}_{bit}^k}{f(P_J^{L_J}, P_J^j)} \tag{21}$$

where $p\hat{e}r$ and $s\hat{e}r$ are calculated based on ACKs from D in the k-th time slot. $\hat{n}_{bit}^k$ is the estimated number of bits in a symbol, which can be calculated based on the jammer's sensing result about the modulation scheme used by S. The denominator term represents the power cost of J. The reward defined here only includes the information of J and its detected information.
Similarly, the reward r_S^k of S can be written as

$$r_S^k = \frac{(1-p\hat{e}r_k)(1-s\hat{e}r_k)n_{bit}^k}{f(P_S^i, P_S^1)} \tag{22}$$

where n_{bit}^k is the number of bits in one symbol. The rewards for J and S are designed based on the utility functions of the game. At the conclusion of each time slot, the rewards for the actions can be calculated based on the received ACKs immediately.

C. DQN-BASED INTELLIGENT JAMMING

In the considered case, both S and J are aware of each other's existence, but they do not share any information. However, they can calculate their rewards based on the feedback ACKs using Eq. (21) and (22).

In the proposed DQN-based strategy-making method, J and S individually train their own strategy-making networks and adjust strategies to maximize their cumulative rewards through learning from current states and historical data. This real-time learning and strategy adaptation process occurs autonomously. A synchronous time-slotted system is considered, i.e., the time slots of the jammer and the intelligent transmitter are aligned [40]. The structure and parameter configuration of both DQNs are identical. Hence, only the structure and parameters of the jammer's DQN will be described in detail thereafter.

In the specific process, at each time slot, S and J perform actions to transmit communication signals and jamming signals, respectively. The state of the agent of J in the k-th time slot is s_J^k, which mainly describes the current environment state of the agent of J. The agent of J adopts a greedy strategy, and a specific action a_J^k will be performed by learning current and partial historical information. The greedy strategy implies a trade-off between exploration and exploitation, aiming to strike a balance between exploring new actions and exploiting the best action known at the moment. The actions of J and S are made simultaneously. In order to calculate the reward, J estimates the SER and PER as $s\hat{e}r$ and $p\hat{e}r$ through the received ACKs sent by D. Then, the jamming reward r_J^k can be obtained. After one time slot, the agent of J moves to the next state s_J^{k+1}, and it can get an experience of $e_J = (s_J^k, a_J^k, r_J^k, s_J^{k+1})$.

Next, the agent of J will store its experiences in the experience pool $\mathcal{M}_J$ of its own. When the experience pool is full, it will take a mini-batch from $\mathcal{M}_J$ to update the network. This technique is defined as experience replay and the data correlation can be reduced using this technique [41].

The DQN-based jamming strategy-making process in k-th time slot is shown in Figure 4. The Q-function $Q(s, a, \theta)$ is the estimated long-term reward of J after executing the action a_J^k under the state s_J^k, and θ is the weight vector of the DQN. Refer to [41], the DQN-based strategy-making scheme adopts a dual neural network structure. The purpose is to obtain better and more stable performance during the process of training.

In this scheme, both the agent of J and the agent of S have the train DQN and the target DQN with the weights vector θ_J^k, θ_S^k and $\hat{\theta}_J^k$, $\hat{\theta}_S^k$, respectively. In the k-th time slot, the agent of J selects a mini-batch $\mathcal{M}_J^k$ with K experiences randomly sampled from the experience pool $\mathcal{M}_J$ and uses the stochastic gradient descent algorithm to minimize the estimation error between the training DQN and the target DQN. The expression of the prediction error as the loss

function is shown in Eq. (23), as shown at the bottom of the page, where δ is the discount factor for future expectations. The gradient to the weight vector of the updated train-DQN is shown in Eq. (24), as shown at the bottom of the page. Repeat the above process until the long-term average benefit becomes stable.

The whole process is also carried out simultaneously at the agent of S. On the whole, in the k-th time slot, the transmitter and the jammer perform actions to transmit communication signals and jamming signals, respectively. Upon successful packet reception, the receiver sends an ACK signal to the transmitter, which can also be received by the jammer. Based on ACKs, the transmitter and the jammer can calculate their rewards r_S^k, r_J^k respectively. They are rewarded differently for changes in their actions. In this manner, the DQNs of the transmitter and the jammer learn through continuous interactions to develop transmission strategies that result in greater rewards for S and J. The specific execution process can be found in Algorithm 1.

Algorithm 1 The Proposed DQN-Based Scheme

1:Construct DQNs at J and S, set their experience pools $\mathcal{M}_J$ and $\mathcal{M}_S$, respectively
2:Initialize target-DQNs and train-DQNs for both S and J randomly with DQNs' weights $\hat{\theta}_J^k = \theta_J^k$, $\hat{\theta}_S^k = \theta_S^k$, respectively
3: J and S choose actions randomly, store their experience e_J, e_S into $\mathcal{M}_J$, $\mathcal{M}_S$, respectively
4:Repeat
  4.1:J, S observe their own states s_J^k, s_S^k, respectively
  4.2:J, S choose their own actions a_J^k, a_S^k, respectively
  4.3:J, S calculate their own rewards r_J^k, r_S^k, respectively
  4.4:J, S get their next states s_J^{k+1}, s_S^{k+1}, respectively
  4.5:J, S store their experience e_J, e_S into $\mathcal{M}_J$, $\mathcal{M}_S$, respectively
  4.6:J, S randomly sample experience from $\mathcal{M}_J$, $\mathcal{M}_S$ to update the weights θ_J^k, θ_S^k of train-DQN, respectively
  4.7:J, S respectively update the weights $\hat{\theta}_J^k$, $\hat{\theta}_S^k$ of target-DQN using the weights θ_J^k, θ_S^k per slot
Until convergence

D. DQN ARCHITECTURE AND COMPLEXITY ANALYSIS

The design of the DQN is critical because an excessive number of neurons can cause problems such as high computational complexity, slow convergence, and overfitting, while too few neurons will degrade the learning performance. For the DQN structure in this work, it contains a FCN. The input and output layers of the network contain N_s, N_a neurons, respectively. They represent the number of elements contained in the state of the DQN and the size of the action space. There are two hidden layers in the network, which contain 256 and 128 neurons, respectively. The numbers of the hidden layers are designed to avoid overfitting and optimize the performance as well as the convergence speed. For each time slot, one forward propagation and one backward propagation are required. The complexity of the proposed scheme is analyzed by calculating the computational complexity of forward propagation and backward propagation. The computational complexity of the forward propagation is mainly related to the number and the size of the hidden layers, and the complexity of the backward propagation is the same as that of the forward propagation. Therefore, the complexity of the DQN scheme is about $O(2^8 N_s + 2^7 N_a + 2^{15})$.

V. SIMULATION RESULTS AND DISCUSSIONS

A. SIMULATION SETUP

In the following simulation, the widely used path loss model is used to model the channel gain. According to [42], the specific channel gain G_h can be written as

$$G_h = \left(\frac{c}{4\pi F_0 d_0}\right)^2 \left(\frac{d_0}{d}\right)^{\beta} \tag{25}$$

where d represents the distance between J and D or the distance between S and D, c is the speed of light, F_0 is the center frequency of the wireless signal, d_0 is the far-field reference distance of the antenna, and β is the path loss exponent.

It is proven in [5] and [6] that the jamming signal using BPSK or QPSK has the optimal jamming effect for various digital communication signals. Therefore, in the following simulations, the modulation sets for S and J are set as {BPSK, QPSK, 16QAM} and {BPSK, QPSK}, respectively. To accommodate the DQN method, we discretize the signal power and jamming time ratio. Additionally, considering the interleaving technique in wireless communications, a jamming time ratio that is too small cannot achieve effective jamming. Therefore, the jamming time ratio is set to different discrete values, with a minimum value of 0.25. More specific and detailed parameter settings are given in Table 3.

The details of the simulations: The modeling of S, D, and J is constructed by Matlab R2020b. Based on Matlab, the digital communication link is programmed and

$$L(\theta_J^k) = \frac{1}{2K} \sum_{e_J \in \mathcal{M}_J^k} (r_J^k + \delta \max_{a_J^{k+1}} Q(s_J^{k+1}, a_J^{k+1}, \hat{\theta}_J^k) - Q(s_J^k, a_J^k, \theta_J^k))^2 \tag{23}$$

$$\frac{\partial L(\theta_J^k)}{\partial \theta_J^k} = \frac{1}{K} \sum_{e_J \in \mathcal{M}_J^k} (r_J^k + \delta \max_{a_J^{k+1}} Q(s_J^{k+1}, a_J^{k+1}, \hat{\theta}_J^k) - Q(s_J^k, a_J^k, \theta_J^k)) \nabla Q(s_J^k, a_J^k, \theta_J^k) \tag{24}$$

TABLE 3. Simulation parameters and numerical settings.

Parameter	Value or Configuration
the distance between S and D	500m
the distance between J and D	700m
the power set of S, $\mathcal{P}_S$	{5W,10W,15W,20W,25W,30W}
the power set of J, $\mathcal{P}_J$	{10W,20W,30W}
the modulation set of S, $\mathcal{C}$	{BPSK, QPSK,16QAM}
the modulation set of J, $\mathcal{J}$	{BPSK, QPSK}
the jamming time ratios, Λ	{0.25,0.5,0.75,1}
central frequency, F_0	900MHz
noise power, σ^2	-114dBw
far-field reference distance, d_0	20m
path-loss exponent, β	3
tolerable SER threshold, α	0.1

implemented, as well as the generation of the jamming signal, as illustrated in the left part of Figure 4. This component is responsible for generating the communication results (i.e., N^k_{ACKs}, modulation schemes) under different transmission strategies of S and J. The proposed intelligent jamming strategy-making scheme is implemented by Python (Version 3.9) and TensorFlow (Version 2.2.0). This component is responsible for receiving communication results, calculating rewards, learning, and generating output actions accordingly. These are illustrated in the right part of Figure 4. The ReLU activation function is applied to each hidden layer and the RMSprop optimizer is used. The simulation programs are executed with a personal computer with a single CPU (AMD R7-4800H) inside.

Hyperparameters have a significant influence on the performance of the intelligent strategy-making scheme. The primary hyperparameters are the learning rate lr, the update interval T_{step}, and the exploration probability ε of ε-greedy algorithm. The learning rate determines how much the neural network adjusts the weights as the gradient is updated. An inappropriate choice of learning rate can make the training process difficult or too slow to converge. The target network is used to stabilize the training process, and a suitable update interval between the train network and the target network can make the training process stable and have good convergence. ε is an important parameter used to balance the exploration and exploitation in the learning process, and its inappropriate value will lead to a local optimum or failure to learn an effective strategy. Accordingly, after continuous trials, we empirically set the learning rate, update frequency and ε to 0.01, 100, and 0.6, respectively. The learning rate and the parameter ε for balancing exploration and exploiting decrease by a coefficient of $\frac{1}{1+2e-4}$ during the training process of the network. The sizes of experience pool and mini-batch are 500 and 32. The simulation is conducted for 4×10^4 time slots.

B. COMPARISON SCHEMES

In order to verify the performance of the proposed jamming strategy-making scheme based on DQN algorithm, the following learning-based or basic strategy-making schemes are set up for comparison.

- QL scheme: QL is a common method that has been widely used earlier. It estimates the value of each action and makes decisions by recording the Q-values of all actions in a locally stored table.
- JB scheme: JB-based jamming strategy scheme was proposed in [28]. By constantly evaluating and updating the benefits of each choice, it tends to select the action which leads to relatively high benefits based on the evaluation of the action benefits in the previous step.
- Random strategy: This scheme means that J randomly selects action from the whole action space with the same probability. This is a classic jamming method.

In the following, the mixed-strategy NE solved by PSO algorithm under the assumption of complete information will be introduced, which can be regarded as a benchmark for subsequent simulations. Then, the value of w and experience replay techniques are discussed for the proposed DQN scheme. Finally, the average utilities of J and the jamming effects are compared for various jamming schemes.

C. APPROXIMATE MIXED STRATEGIES WITH COMPLETE INFORMATION

When both parties have each other's information, the non-zero-sum game in Section II can be solved by the PSO algorithm.

Considering the larger dimensions of action space, the population size N_p is set to 500. According to the relevant literature [34], [35], [36], the fitness function accuracy threshold is set as 10^{-4}. The values of ω_{max} and ω_{min} are set to 0.9 and 0.4, respectively. After several simulations, it has been found that the threshold accuracy requirement of the fitness function can be satisfied when the number of iterations exceeds 800. Therefore, we empirically set the maximum number of iterations to 1000 to ensure that the accuracy meets the requirements.

Under such a parameterization, the simulation is conducted in Python and repeated 10 times so that an average value of the results can be obtained as a benchmark for the following simulations. During the process of solving the game equilibrium, the values of the fitness function in ten simulations have similar trends and all converge to 0. For convenience of observation, Figure 5 displays the fitness function curves. The lines with different colors are the fitness function curves of PSO algorithm for 10 simulations.

After ten executions of PSO algorithm, we can find that the value fitness always meets the accuracy threshold after 800 iterations. With the value of fitness function reaching the threshold requirement, the simulation results of PSO algorithm can be considered as an approximate NE. In this case, the average value of utilities obtained by 10 simulations

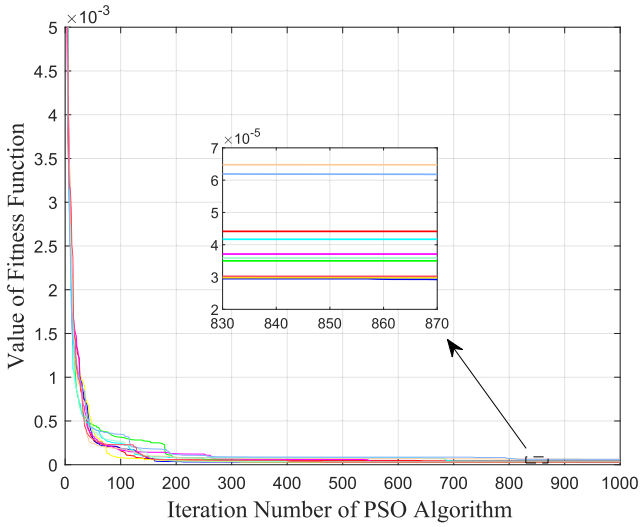
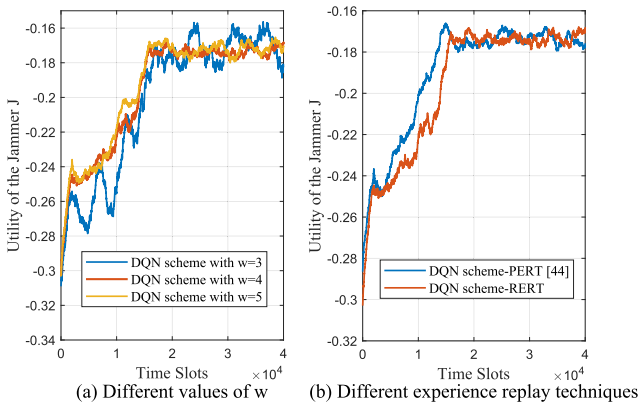


FIGURE 5. Values of fitness function of the PSO algorithm.

is taken as the benchmark for the utility of the jammer. After calculating, the benchmark of the jammer's utility is −0.1736.

D. EFFECTIVENESS OF THE DQN SCHEME

When S and J are aware of each other's existence but do not share any information, they can independently adjust strategies using the proposed DQN scheme. In the design of the rewards, the number of previous actions or observations that is determined by w in Eq. (21) and (22), has a certain impact on the performance. Better learning effect can be obtained for the DQN scheme by using larger values of w. However, the increase of w will increase the computational complexity, so it is necessary to select an appropriate w to strike a balance between the complexity and the performance gain.



(a) Different values of w (b) Different experience replay techniques

FIGURE 6. Influence of w and experience replay techniques.

Subgraph (a) of Figure 6 shows the influence of different values of w on the convergence of the jamming utility. The number of time slots for the simulation is long enough so that curves with different values of w are converged. We can see that the values of w do not influence the converged value of the utility. However, the jammer's utility grows faster at $w = 4$ or 5 compared to $w = 3$. Additionally, we have recorded the running time during the simulation of 4×10^4 time slots for different values of w. The average running time of the program for $w = 3$, 4, and 5 is 5612s, 6457s, and 7340s, respectively. It can be found that larger value of w brings faster convergence of the utility, but also results in higher computation complexity. To balance the convergence speed and the computation complexity, we set $w = 4$ in the following simulations.

Another way to speed up the convergence of the DQN based scheme is to use advanced experience replay methods, such as QiERT, PERT [43], [44]. These advanced replay techniques can improve the convergence at the cost of computational complexity for problems with large action space, unstable environments, or high data acquisition costs. In subgraph (b) of Figure 6, the utility curve using PERT is shown to compare with RERT used in this paper [44]. We can see that using PERT does speed up the learning process. Compared to RERT, the jamming utility can converge about 1500 time slots earlier. However, PERT requires ranking and selection of the experiences, which increases the complexity of the algorithm. The running time during the simulation of 4×10^4 time slots using PERT and RERT is 7924s and 6457s, respectively. Specifically, time costs of PERT have increased by 22.72%. To balance the learning speed and the computational cost, the basic RERT is used in the following simulations.

To show the effectiveness of the proposed DQN scheme, various jamming schemes are compared in Figure 7. The dashed line represents the jamming utility with complete information. After convergence, the DQN scheme can achieve similar utility values as the complete information case. Specifically, the converged utility of the DQN scheme is only 0.12% less than the jamming utility achieved by the approximate mixed-strategy NE, which is the ideal utility under complete information. The jamming utility values of QL scheme and JB scheme are respectively 6.28% and 28.00% less than the jamming utility of the approximate mixed-strategy NE.

Compared to the learning-based schemes, the proposed DQN scheme not only achieves better jamming utility but also has a faster convergence speed. In terms of convergence speed, the DQN scheme converges after 16000 time slots and remains near the NE utility. The QL scheme does not reach convergence until 20000 time slots. The reason is that the size of the Q-value table of the QL scheme is related to the number of possible actions and the state space, resulting in a larger size of the Q-value table and slower convergence speed. Compared to random strategy and JB scheme, the proposed DQN scheme shows a significant performance advantage in terms of jamming utility after convergence. It's important to note that DQN uses neural networks to estimate Q-value, which increases the amount of computation per time slot. However, it is worthwhile to pay

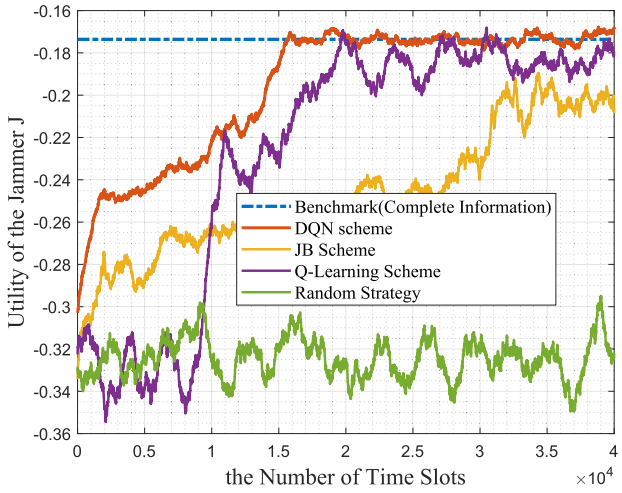


FIGURE 7. Comparisons of various schemes in terms of jammer's utility.

the corresponding computational costs in order to improve the jamming performance.

In order to verify the jamming effects of various jamming schemes, the average utility of J, the power consumption of J, and PER at D after the 20000-th time slot are calculated. The results are shown in Table 4.

TABLE 4. Utility, PER and power consumption of various schemes.

Jamming Schemes	Utility	PER	P_J(W)
DQN	-0.1738	0.7493	16.6915
Q-Learning	-0.1845	0.7071	17.5498
Jamming-Bandits	-0.2222	0.6571	17.5885
Random	-0.3257	0.5967	20.0000

It is evident that, compared to the random jamming strategy, all three intelligent strategy-making schemes all achieve better jamming effects using lower power consumption. Compared with the other two learning-based methods, the DQN scheme has a higher PER with lower jamming power. It achieves the best jamming effect with the lowest power consumption among all the jamming strategy-making schemes.

E. ANALYSIS OF ASYNCHRONOUS JAMMING

The aforementioned simulations and results are based on the precondition that J and S are synchronized [28]. This is possible in practical wireless systems such as LTE because they use signals like PSS and SSS for synchronization between the victim receiver and the transmitter. When the jammer encounters these signals, it can also synchronize with the victim [28]. In Section III of complete information which works as a benchmark, SER is used in the utilities of S and J, so the jammer is assumed to be symbol-synchronous with S. This assumption is strong. However, these results are only shown as a benchmark to verify the effectiveness of the proposed DQN scheme. In Section IV of the proposed DQN scheme, PER is used to calculate the rewards in Eq. (21) and (22), where SER is estimated based on PER which is evaluated using ACKs. In this case, J only needs to be synchronous with S on a per-packet basis. Similar assumptions have been made in [5], [25], [28], [38], [40], [45], and [46]. However, if the jammer is not synchronized, then the jamming performance could be degraded.

Here, we consider the jamming performance when J and S are not synchronized with each other. In general, phase offset and symbol timing offset can occur together in practical wireless communication systems, here we do not consider both these non-idealises together due to the complexity involved in simulations. However, the framework developed thus far is still applicable and can be extended to such complex scenarios. Below, we focus on the effects of phase offset on the jamming performance.

If there is a random phase offset between the jamming signal and the communication signal, so that the signal at the receiver can be equivalently represented as

$$r_k = \sqrt{P_S} h_S s_k + \sqrt{P_J} h_J v_k e^{i\phi} + n_k \tag{26}$$

where ϕ represents the phase offset at D and is regarded as a uniform random variable between 0 and 2π, and $i = \sqrt{-1}$.

The parameter settings remain consistent with those in Section V-A. Only a random phase offset is added between the jamming signal and the communication signal. The utility of the jammer is shown in subgraph (b) of Figure 8.

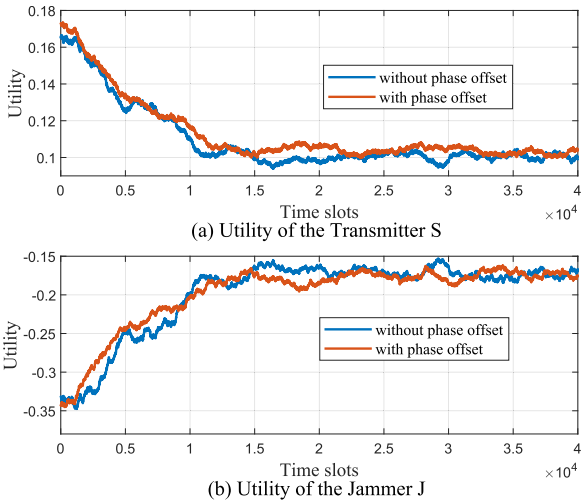


FIGURE 8. Utilities of S and J without/with phase offset.

It can be observed that, considering the phase offset, the average utility of J after convergence is slightly lower than the synchronized case. Particularly, let $(0.1034, -0.1771)$ represent the utilities of S and J in the presence of phase offset. Compared to the utilities $(0.1009, -0.1736)$ without phase offset, the utility of J decreases by 2.02% while the utility of S increases by 2.51%. The numerical change in utility is subtle in the case of considering phase offset. The

corresponding research issues also exist in [6], showing that the effects of the asynchronous problem are relatively minor.

VI. CONCLUSION

In this paper, we model the interaction between the jammer and the transmitter in an electronic warfare scenario as a non-zero-sum game, where both the jammer and the transmitter can proactively adjust their strategies. Due to the non-cooperative nature of the jammer and the transmitter, a DQN-based jamming strategy-making method is proposed to learn the intelligent jamming strategies without complete or partial information about the transmitter. The rewards are designed to utilize the ACKs feedback mechanism of the communication system. Compared to the benchmark of approximate mixed strategies NE under the complete information, the proposed DQN method can quickly converge and stabilize near the jamming utility provided by the benchmark. Compared with other classical learning-based methods, the proposed DQN method can provide intelligent jamming strategies that result in a higher PER for the communication party with reduced jamming power consumption.

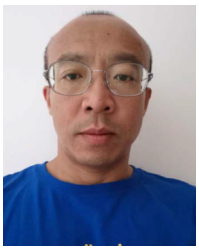
ACKNOWLEDGMENT

(Yongcheng Li and Weijian Miao are co-first authors.)

REFERENCES

[1] H. Sun, A. Nallanathan, C.-X. Wang, and Y. Chen, "Wideband spectrum sensing for cognitive radio networks: A survey," *IEEE Wireless Commun.*, vol. 20, no. 2, pp. 74–81, Apr. 2013.
[2] L. Zhao, B. Wang, and W. Hou, "Game decision modeling of communication electronic jamming pattern selection," *J. Detection Control*, vol. 43, no. 4, pp. 71–80, 2021.
[3] L. Pei, H. Liu, and K. Liu, "A jamming scheme decision method based on artificial bee colony algorithm," *Fire Control Command Control*, pp. 1–6, 2024.
[4] S. Bayram, N. D. Vanli, B. Dulek, I. Sezer, and S. Gezici, "Optimum power allocation for average power constrained jammers in the presence of non-Gaussian noise," *IEEE Commun. Lett.*, vol. 16, no. 8, pp. 1153–1156, Aug. 2012.
[5] S. Amuru and R. M. Buehrer, "Optimal jamming strategies in digital communications—Impact of modulation," in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 1619–1624.
[6] S. Amuru and R. M. Buehrer, "Optimal jamming against digital modulation," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 10, pp. 2212–2224, Oct. 2015.
[7] Y. E. Sagduyu, R. A. Berry, and A. Ephremides, "Jamming games in wireless networks with incomplete information," *IEEE Commun. Mag.*, vol. 49, no. 8, pp. 112–118, Aug. 2011.
[8] L. Xiao, J. Liu, Q. Li, N. B. Mandayam, and H. V. Poor, "User-centric view of jamming games in cognitive radio networks," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 12, pp. 2578–2590, Dec. 2015.
[9] K. Dabcevic, A. Betancourt, L. Marcenaro, and C. S. Regazzoni, "A fictitious play-based game-theoretical approach to alleviating jamming attacks for cognitive radios," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 8158–8162.
[10] Y. Gao, Y. Xiao, M. Wu, M. Xiao, and J. Shao, "Game theory-based anti-jamming strategies for frequency hopping wireless communications," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5314–5326, Aug. 2018.
[11] K. Firouzbakht, G. Noubir, and M. Salehi, "Linearly constrained bimatrix games in wireless communications," *IEEE Trans. Commun.*, vol. 64, no. 1, pp. 429–440, Jan. 2016.
[12] A. Garnaev, W. Trappe, and A. Petropulu, "Combating jamming in wireless networks: A Bayesian game with Jammer's channel uncertainty," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 2447–2451.
[13] N. Qi, W. Wang, F. Zhou, L. Jia, Q. Wu, S. Jin, and M. Xiao, "Two birds with one stone: Simultaneous jamming and eavesdropping with the Bayesian-stackelberg game," *IEEE Trans. Commun.*, vol. 69, no. 12, pp. 8013–8027, Dec. 2021.
[14] H.-S. Im and S.-H. Lee, "Anti-jamming games in multi-band wireless ad hoc networks," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 872–887, Jun. 2023, doi: 10.1109/TIFS.2022.3227422.
[15] L. Jia, F. Yao, Y. Sun, Y. Niu, and Y. Zhu, "Bayesian Stackelberg game for antijamming transmission with incomplete information," *IEEE Commun. Lett.*, vol. 20, no. 10, pp. 1991–1994, Oct. 2016.
[16] Y. Xu, Y. Xu, G. Ren, J. Chen, C. Yao, L. Jia, D. Liu, and X. Wang, "Play it by ear: Context-aware distributed coordinated anti-jamming channel access," *IEEE Trans. Inf. Forensics Security*, early access, Nov. 15, 2021, doi: 10.1109/TIFS.2021.3128249.
[17] Y. Sun, Y. Zhu, K. An, G. Zheng, S. Chatzinotas, K.-K. Wong, and P. Liu, "Robust design for RIS-assisted anti-jamming communications with imperfect angular information: A game-theoretic perspective," *IEEE Trans. Veh. Technol.*, vol. 71, no. 7, pp. 7967–7972, Jul. 2022.
[18] Q. Zhu, H. Li, Z. Han, and T. Basar, "A stochastic game model for jamming in multi-channel cognitive radio systems," in *Proc. IEEE Int. Conf. Commun.*, May 2010, pp. 1–6.
[19] N. Wu, X. Zhou, and M. Sun, "Multi-channel jamming attacks against cooperative defense: A two-level Stackelberg game approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.
[20] A. Garnaev, A. Petropulu, W. Trappe, and H. V. Poor, "A multi-jammer game with latency as the user's communication utility," *IEEE Commun. Lett.*, vol. 24, no. 9, pp. 1899–1903, Sep. 2020.
[21] L.-W. Feng, S.-T. Liu, and H.-Z. Xu, "Multifunctional radar cognitive jamming decision based on dueling double deep Q-network," *IEEE Access*, vol. 10, pp. 112150–112157, 2022.
[22] Q. Xing, X. Jia, and W. Zhu, "Intelligent radar countermeasure based on Q-learning," *Syst. Eng. Electron.*, vol. 40, no. 5, p. 1031, 2018.
[23] Z. Liu, D. Cheng, N. Li, L. Min, and Z. Guo, "Two-dimensional precise controllable smart jamming against SAR via phase errors modulation of transmitted signal," *IEEE Geosci. Remote Sens. Lett.*, vol. 21, pp. 1–5, 2024.
[24] N. Rao, H. Xu, and B. Song, "An intelligent jamming decision algorithm based on action elimination dueling double deep Q network," *J. Air Force Eng. Univ. Natural Sci. Ed.*, vol. 22, no. 4, pp. 92–98, 2021.
[25] N. Rao, H. Xu, and B. Song, "Q-learning intelligent jamming decision algorithm based on efficient upper confidence bound variance," *J. Harbin Inst. Technol.*, vol. 54, no. 5, pp. 162–170, 2022.
[26] S. Luo and X. Liu, "UAV intelligent approach jamming wireless communication system," in *Proc. 3rd Int. Conf. Neural Netw., Inf. Commun. Eng. (NNICE)*, Feb. 2023, pp. 427–432.
[27] Y. Shi, Y. E. Sagduyu, T. Erpek, K. Davaslioglu, Z. Lu, and J. H. Li, "Adversarial deep learning for cognitive radio security: Jamming attack and defense strategies," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2018, pp. 1–6.
[28] S. Amuru, C. Tekin, M. v. der Schaar, and R. M. Buehrer, "Jamming bandits—A novel learning method for optimal jamming," *IEEE Trans. Wireless Commun.*, vol. 15, no. 4, pp. 2792–2808, Apr. 2016.
[29] G. Kim and H. Lim, "Reinforcement learning based beamforming jammer for unknown wireless networks," *IEEE Access*, vol. 8, pp. 210127–210139, 2020.
[30] D. Fudenberg and J. Tirole, *Game Theory*. Cambridge, MA, USA: MIT Press, 1991.
[31] A. Dixit and S. Skeath, *Games of Strategy*. New York, NY, USA: W. W. Norton & Company, 1999.
[32] J. Proakis and M. Salehi, *Digital Communications*, 5th ed., New York, NY, USA: McGraw-Hill, 2008.
[33] Y. Li, R. Zhao, Y. Wang, G. Pan, and C. Li, "Artificial noise aided precoding with imperfect CSI in full-duplex relaying secure communications," *IEEE Access*, vol. 6, pp. 44107–44119, 2018.
[34] Q. Yu and X. Wang, "Evolutionary algorithm for solving Nash equilibrium based on particle swarm optimization," *J. Wuhan Univ.*, no. 1, pp. 25–29, Feb. 2006.
[35] H. Duan, P. Li, and Y. Yu, "A predator-prey particle swarm optimization approach to multiple UCAV air combat modeled by dynamic game theory," *IEEE/CAA J. Autom. Sinica*, vol. 2, no. 1, pp. 11–18, Jan. 2015.
[36] Y. Zhao, Y. Song, and L. Kang, "Solving Nash equilibrium for non-cooperative game based on the particle swarm optimization integrating multiply strategies," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2019, pp. 4506–4511.

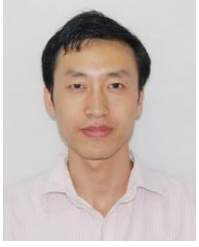
[37] C. Zou, K. An, Z. Lin, Y. He, X. Zhong, G. Zheng, and N. Al-Dhahir, "Multi-layer RIS-assisted anti-jamming communications: A hierarchical game learning approach," *IEEE Commun. Lett.*, vol. 27, no. 11, pp. 2998–3002, Nov. 2023.
[38] P. D. Thanh, H. T. H. Giang, and I.-P. Hong, "Anti-jamming RIS communications using DQN-based algorithm," *IEEE Access*, vol. 10, pp. 28422–28433, 2022.
[39] Z. Yin, J. Li, Z. Wang, Y. Qian, Y. Lin, F. Shu, and W. Chen, "UAV communication against intelligent jamming: A Stackelberg game approach with federated reinforcement learning," *IEEE Trans. Green Commun. Netw.*, early access, Mar. 8, 2024, doi: 10.1109/TGCN.2024.3373886.
[40] W. Li, Y. Xu, J. Chen, H. Yuan, H. Han, Y. Xu, and Z. Feng, "Know thy enemy: An opponent modeling-based anti-intelligent jamming strategy beyond equilibrium solutions," *IEEE Wireless Commun. Lett.*, vol. 12, no. 2, pp. 217–221, Feb. 2023.
[41] V. Mnih, K. Kavukcuoglu, and D. Silver, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[42] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
[43] Y. Li, A. H. Aghvami, and D. Dong, "Path planning for cellular-connected UAV: A DRL solution with quantum-inspired experience replay," *IEEE Trans. Wireless Commun.*, vol. 21, no. 10, pp. 7897–7912, Oct. 2022.
[44] X. Tao and A. S. Hafid, "DeepSensing: A novel mobile crowdsensing framework with double deep Q-network and prioritized experience replay," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11547–11558, Dec. 2020.
[45] N. Gao, Z. Qin, X. Jing, Q. Ni, and S. Jin, "Anti-intelligent UAV jamming strategy via deep Q-networks," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 569–581, Jan. 2020.
[46] Y. Chen, Y. Li, D. Xu, and L. Xiao, "DQN-based power control for IoT transmission against jamming," in *Proc. IEEE 87th Veh. Technol. Conf. (VTC Spring)*, Jun. 2018, pp. 1–5.



YONGCHENG LI received the master's degree from the University of Science and Technology of China, in 2012. Then, he joined the State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System. His research interests include cognitive radio networks and comprehensive effect mechanisms of electromagnetic environment.

WEIJIAN MIAO received the B.E. degree in information engineering from Xi'an Jiaotong University, Xi'an, China, in 2021, where he is currently pursuing the master's degree in information and communications engineering. His current research interests include intelligent jamming techniques and game theory in wireless communication systems.

ZHENZHEN GAO received the B.S. degree in communication engineering from Lanzhou University, Lanzhou, China, in 2005, and the Ph.D. degree from Xi'an Jiaotong University, Xi'an, China, in 2011. From August 2009 to September 2011, she was a Visiting Student with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD, USA. Since 2012, she has been with the School of Information and Communication Engineering, Xi'an Jiaotong University, where she is currently an Associate Professor. Her current research interests include physical-layer security, index modulation, and advanced techniques in 5/6G wireless communication networks.



GANGMING LV received the Ph.D. degree in communication and information systems from Xi'an Jiaotong University, China, in 2010. He is currently with the Department of Information and Communications Engineering, Xi'an Jiaotong University. His research interests include mobile wireless communications and networking, with a focus on radio resource allocation, QoS provisioning, satellite communications, and wireless sensor networks.