Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 7962686, 18 pages
https://doi.org/10.1155/2022/7962686

Research Article

Joint Optimization of Jamming Link and Power Control in Communication Countermeasures: A Multiagent Deep Reinforcement Learning Approach

Ning Rao, Hua Xu, Yue Zhang, Dan Wang, Lei Jiang, and Xiang Peng

Information and Navigation College, Air Force Engineering University, Xi'an 710077, China

Correspondence should be addressed to Ning Rao; raoningmabma@163.com

Received 5 July 2022; Revised 28 November 2022; Accepted 14 December 2022; Published 29 December 2022

Academic Editor: Carles Gomez

Due to the nonconvexity feature of optimal controlling such as jamming link selection and jamming power allocation issues, obtaining the optimal resource allocation strategy in communication countermeasures scenarios is challenging. Thus, we propose a novel decentralized jamming resource allocation algorithm based on multiagent deep reinforcement learning (MADRL) to improve the efficiency of jamming resource allocation in battlefield communication countermeasures. We first model the communication jamming resource allocation problem as a fully cooperative multiagent task, considering the cooperative interrelationship of jamming equipment (JE). Then, to alleviate the nonstationarity feature and high decision dimensions in the multiagent system, we introduce a centralized training with decentralized execution framework (CTDE), which means all JEs are trained with global information and rely on their local observations only while making decisions. Each JE obtains a decentralized policy after the training process. Subsequently, we develop the multiagent soft actor-critic (MASAC) algorithm to enhance the exploration capability of agents and accelerate the learning of cooperative policies among agents by leveraging the maximum policy entropy criterion. Finally, the simulation results are presented to demonstrate that the proposed MASAC algorithm outperforms the existing centralized allocation benchmark algorithms.

1. Introduction

With the advent of the information age, the electromagnetic field has become the fifth battlefield domain following the land, sea, air, and space [1]. Advances in electronic warfare (EW) have determined the winners of war. Research, development, and adoption of these technical advances shape the tactics, operational maneuvers, and strategies employed by commanders at all levels in the battle space. Advanced research and development efforts contribute substantially to deterrence.

In recent years, EW has shifted towards cognitive electronic warfare (CEW). CEW is one of the critical advances that will determine the outcomes of future battles. The application of artificial intelligence (AI) to make EW systems cognitive is a promising attempt to allow systems to adapt and to learn during a mission. Given the rapid rate of innovation and advancements in AI, military systems must leverage these advances at the speed of relevance. In the observe-orient-decide-act (OODA) loop of EW, decision-making is the key link to ensure the effectiveness of electronic countermeasures [2]. Intelligent decision-making has become a significant research orientation in CEW [3–5]. Game theory, distributed optimization, and successive convex approximation methods have been adopted in the field of optimization of physical layer parameters [6–13]. Nevertheless, these approaches are limited to prior knowledge, which is challenging to attain on the battlefield. In a digital world of software-defined capabilities, EW systems must respond to previously unknown signals. In the densely connected battle space of today, the so-called internet of military things, feedback can be known or at least estimated continuously during a mission.

Reinforcement learning (RL), as a particular machine learning method, does not require prior information, which can be applied in the form of interacting with an unknown

environment repeatedly to optimize the strategy [14]. However, the effectiveness of RL diminishes as the size of the state action becomes large. With the rapid improvement of the computing capabilities of electronic devices, deep learning (DL) has achieved great success in the field of AI [15]. Deep reinforcement learning (DRL), which combines DL and RL, has demonstrated amazing autonomous decision-making capabilities in the fields of unmanned driving and video game playing with a large size of state-action space [16, 17]. Meanwhile, AI is also constantly driving the development of wireless networks towards intelligence [18]. Incorporating AI techniques into EW systems is the only way to manage the complexity of this problem domain and its rapid timescales.

Cognitive communication jamming in communication countermeasures has been a largely underexplored domain. However, few works have been done to solve communication jamming resource allocation to the best of our knowledge. Due to the sensitivity of the data sources, few public studies exist in this field. Research works on jamming resource allocation are urgent due to the future system countermeasures mode of CEW.

In this paper, we have established an efficient jamming resource allocation scheme to maximize the total blanket jamming rate and reduce the jamming resource consumed. The main contributions are presented as follows:

(i) We model the problem of jamming resource allocation in the confrontation with networked communication as a cooperative MADRL task. Specifically, the task is defined as a partially observable Markov's decision process (POMDP), and the global state, partial observation, action, and reward function are defined for all jamming equipment (JE)

(ii) Due to the nonstationarity in the multiagent system and large-scale action space, we adopt a centralized training with a decentralized execution framework to improve decision-making efficiency [19], i.e., using global information while training each JE, and after the training process, each JE can make decisions independently based on its local observation

(iii) We propose a multiagent soft actor-critic (MASAC) algorithm based on soft actor-critic (SAC) [20] to learn the joint optimal policy for all JEs, where the maximum policy entropy is applied to enhance the agent's exploration capability

(iv) We verify the feasibility of MASAC via numerical simulation and demonstrate that our algorithm can achieve higher performance compared with other existing works, including the centralized allocation algorithm based on DDPG in [21] and the multiagent deep deterministic policy gradient algorithm (MADDPG) in [19]

The rest of this paper is organized as follows. In Section 2, an overview of the related works is provided. The confrontation model is introduced in Section 3. Section 4 formulates the model as a cooperative multiagent task and presents the MASAC algorithm to solve the jamming resource allocation optimization problem. Section 5 gives the computational complexity analysis and convergence proof of the algorithm. The numerical results are provided in Section 6. Finally, Section 7 concludes this work.

2. Related Work

Intelligent decision-making is a significant part in the OODA loop. However, for mostly current studies, it is not practical enough due to the requirement of complete a priori information. At present, RL approach has preliminary applications in communication jamming decision-making without prior knowledge. In [22], the authors studied the optimal physical layer jamming patterns based on the multiarm bandit (MAB) framework. Furthermore, the authors in [23] used orthogonal matching pursuit based on MAB to optimize the jamming strategy, and the jamming patterns were enriched. The study in [24] designed an intelligent jamming method based on reinforcement learning to combat the DRL-based user and verified that the proposed RL-based jamming can effectively restrict the performance of DRL-based antijamming method in frequency band selection.

Those works mainly focus on the optimization of jamming signals' parameters. Optimization at this signal level is far from enough; optimization at higher levels such as function-oriented jamming resource allocation is also pivotal. Currently, applying DRL to solve the problem of resource allocation in large-dimensional space has become a research hotspot, due to its simple objective function and no requirement of accurate prior information. DRL can be divided into single-agent DRL and multiagent deep reinforcement learning (MADRL) methods.

In the single-agent RL methods, the agent concentrates the state and action information of all devices or users together to form an expanded state and action space and completes tasks such as user scheduling [25], channel management [26], and power allocation [27] through centralized controlling. However, all the aforementioned works [25–27] with centralized scheduling methods have inevitable problems such as large-scale decision space, unnecessary communication overhead, and poor system scalability [28]. Thus, this strategy is generally suitable for scenarios with small-scale decision dimensions.

In the MADRL-based methods, each device or user is regarded as an agent, finishing tasks through collaborative decision-making, so that the amount of input and output nodes of a neural network can be reduced since each agent makes decisions independently [29]. To further improve the decision-making efficiency in MADRL methods, the authors in [30] adopted a centralized proximal policy optimization algorithm in a fully cooperative multiagent task of handover control and power allocation, in which each device was trained with global information. The study in [31] investigated the distributed deep Q network algorithm in communication network resource allocation. Centralized training was carried out through the central node; then, the trained model parameters were distributed to each base station, and user satisfaction and system stability have been

improved in the case of large service demand. Considering the dueling double Q network structure, the authors in [32] proposed a more efficient deep Q network method in heterogeneous networks, in which each device relied on the global information obtained by information transmission to perform stochastic games. The research in [21] has assumed that the attributes of communication links in different regions are roughly identical, so that each agent could share a common policy network, and the total transmission rate of multiuser wireless cellular network is improved through centralized decision-making based on the deep deterministic policy gradient (DDPG) method [33].

These works on resource allocation mainly focus on the field of wireless communication networks. From these results, we can draw the feasibility of using DRL approach to solve the resource allocation problem.

Few works pay attention to the allocation of communication jamming resources, which is critical for improving operational effectiveness in the communication countermeasure. Therefore, we use MADRL algorithm to solve the problem of jamming resource allocation. Besides, we leverage the maximum policy entropy criterion to enhance the exploration capability of agents and accelerate the learning of cooperative policies among agents.

3. Confrontation Model

In the communication confrontation scenario, the jammer carries out a blanket jamming task against the targets, as shown in Figure 1. Several early-warning aircrafts offer communication services to fighter plane groups. When fighter planes find that the communication links are disturbed, to maintain the communication services, they would switch to other communication links provided by early-warning aircrafts in the region. The jammer conducts reconnaissance on the target spectrum, and the intelligent engine inside the C2 terminal completes the assignment of jamming tasks according to the reconnaissance information and sends them to all JEs. Unmanned aerial vehicle (UAV) is widely applied as jammers to defend against eavesdropping [14], and we use UAVs as aggressive JE in this paper. Assuming that the jammer possesses N JE, the set of JE is represented as N, and $N = \{1, 2, \cdots, N\}$. Each JE conducts interference in the barrage jamming mode. The set of communication links offered by early-warning aircrafts is denoted by M, and $M = \{1, 2, \cdots, M\}$, where M is the number of communication links. Moreover, the channel of each link is assumed to be orthogonal with equal bandwidth and mutually independent.

Assuming that the jammer grasps the center frequency of each link through communication reconnaissance and intelligence analysis, the relative importance factor of each link is obtained as

$$W = [\omega_1, \omega_2, \cdots, \omega_M]. \tag{1}$$

To degrade the performance of communication services, the jammer expects to reasonably designate the jamming task for each JE under the current jamming resource constraints and attempts to achieve the blanket jamming to all transmissions through the cooperation among all JEs.

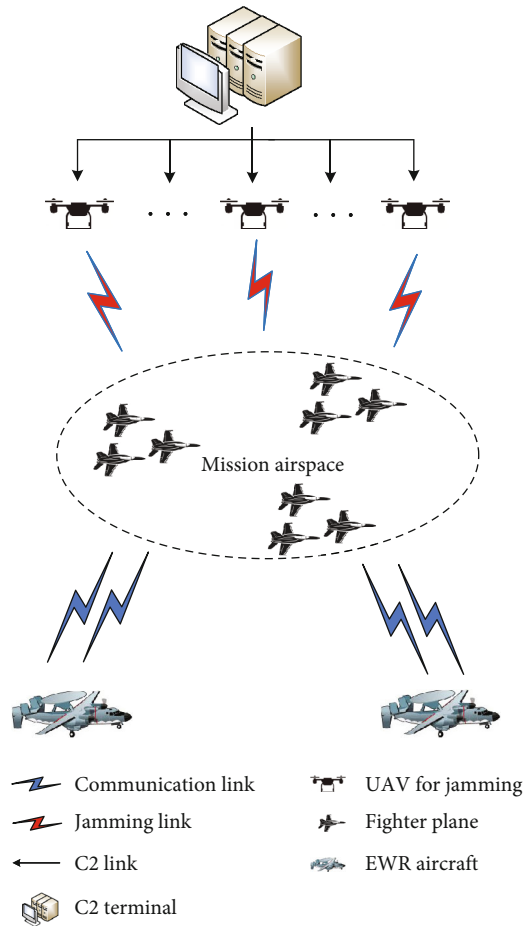


Figure 1: Communication confrontation scenario.

All JEs are faced with some constraints in jamming. Supposing each JE can simultaneously interfere with U links, the total jamming signal power allocated to different links is subject to a given maximum power, which is given by

$$\sum_{u=1}^{U} P_u \le P_{\max}, \tag{2}$$

where $P_{\max}$ is the maximum transmitting power of each JE, and P_u is the jamming-signal power of a link as seen at the JE.

Furthermore, let $P_i(t)$ denote the communication signal power transmitted by the early-warning aircraft of link i at time t, and $G_i(t)$ denotes the channel gain. $P_i^j(t)$ and $G_i^j(t)$ represent the jamming signal power distributed to the link i by the JE j and the corresponding jamming channel gain, respectively.

Note that a link may suffer interference from different JE; then, the jamming-signal-ratio (JSR) at the aircraft receiver in link i can be expressed as

$$\mathrm{JSR}_i(t) = \frac{\sum_{j=1}^{k} P_i^j(t) G_i^j(t) L_j + \sigma^2}{P_i(t) G_i(t) L_i}, k \le N, \tag{3}$$

where k represents the number of JEs that interfere with link i, σ^2 is the noise power. L_i and L_j denote the transmission loss of ground-to-air communication links and jamming links, respectively. In this paper, we adopt the precise interference mode, which refers to that the band of the jamming signal, and the band of the communication signal is exactly matched; the part of the jamming signal outside the filtering range of the communication receiver filter can be as small as negligible, so the out-of-spectrum loss of energy is not considered. For the convenience of analysis, the transmission loss is assumed to be the free space basic transmission loss, which can be given by

$$L = 20 \lg \left(\frac{4\pi r}{\lambda}\right) = 32.4 + 20 \lg (f/\text{MHz}) + 20 \lg (r/\text{km}), \tag{4}$$

where f denotes the center frequency, and r is the signal transmission distance.

Furthermore, the blanket jamming coefficient is introduced, denoted by K_i, to quantitatively describe the impact of jamming on communication receivers. When the JSR of all links exceed the jamming blanket coefficient K_i, the overall blanket jamming goal is achieved, which is based on the following criterion:

$$\text{JSR}_i(t) \geq K_i, \forall i \in M, \tag{5}$$

where K_i is the blanket jamming coefficient. Note that in the CEW, the accurate prior knowledge of the blanket jamming coefficient cannot be obtained in advance by the jammer, but the jammer can learn this knowledge by interacting with the environment.

Then, combined with the important factor of each link, under the unknown blanket jamming coefficient, the jamming link selection and jamming power allocation to achieve the overall blanket jamming can be formalized into an optimization, as

$$\max \sum_{i=1}^{M} \omega_i . \frac{\sum_{j=1}^{N} P_i^j x_i^j G_i^j L_j + \sigma^2}{P_i G_i L_i} \omega_i \in W. \tag{6}$$

Subject to

$$\begin{cases} C1: \dfrac{\sum_{j=1}^{N} P_i^j x_i^j G_i^j L_j + \sigma^2}{P_i G_i L_i} \geq K_i & \forall i \in M_s, \\ C2: \sum_{i=1}^{M} P_i^j \leq P_{\max} & \forall j \in N_s, \\ C3: \sum_{i=1}^{M} x_i^j \leq U & \forall j \in N_s, U \leq N, \\ C4: x_i^j \in \{0, 1\} & \forall i \in M_s, \forall j \in N_s, \end{cases} \tag{7}$$

The objective function (6) represents the goal of maximizing the overall JSR under the premise of giving priority to interfering with more important links. As for the constraint condition (7), $C1$ means that JSR of a certain link should exceed the jamming blanket coefficient. $C2$ and $C3$ mean that each JE can simultaneously interfere with U links, and the total maximum jamming power of each JE distributed to different links is limited to $P_{\max}$. And x_i^n in $C4$ is a binary indicator variable. When $x_i^n = 1$, the jammer allocates the JE N to interfere with link i. Correspondingly, the jamming signal power is P_i^n. Therefore, P_i^n and x_i^n are the optimization variables in this scenario.

4. Cooperative Jamming Resource Allocation Algorithm Based on MASAC

The variables to be optimized in the problem described by (6) and (7) are in mixed discrete and continuous space. In addition, this problem is a dynamic optimization, which requires the jammer to constantly interact with the external environment to obtain the feedback of the interference scheme. And the jammer lacks prior information, for instance, the characteristics of the transmitter-receiver pair are unknown to the jammer.

Traditional approaches to solving such NP-hard combinatorial optimization problems mainly include three categories: exact algorithms, approximate algorithms, and heuristic algorithms, all of which need complete model knowledge. However, in this optimization problem, the important information of the opponent, such as the jamming blanket coefficient K_i communication signal power and communication channel information, is not available for jammers, and these algorithms also cannot guarantee the quality of the solution in polynomial time.

In this paper, DRL approach is used to solve the problem. Different from traditional optimization approaches, DRL does not need prior information. It adopts trial and error to optimize the policy, which means controlling the agent to constantly interact with the environment and modifying the policy according to the feedback from the environment. The purpose is to maximize the expectation of cumulative rewards. This approach can deal with the problem proposed in this paper without accurate knowledge of the communication network.

4.1. Partially Observable Markov's Decision Process. The problem of cooperative resource allocation of multiple JE is modeled as a fully cooperative multiagent task [34]. The multiagent task can be defined as partially observable Markov's decision process (POMDP) denoted by E, and $E = <S, A, P, r, Z, O, N, \gamma>$, in which S represents the global environment state space, A is the action space, P denotes the state transition probability, r represents the reward function, Z is the local observation space, O is the observation function, N is the number of agents, and γ is the discount factor, respectively.

The POMDP can be described as follows:

At each time step t, each agent obtains its observation $z \in Z$ of the external environment according to the observation function $O(s)$: $S \longrightarrow Z$. Then, the agent j (in this paper, the JE) $j \in N = \{1, 2, \cdots, N\}$ chooses actions based on its

policy function $\pi^j(a_t^j|z_t^j)$: $Z \times A \longrightarrow [0,1]$, where z_t^j represents the local observation of agent j. The actions of all agents can constitute a joint action a_t. When the joint action a_t is applied to the environment under the state s_t based on all agents' observations, the environment will transit to the next state s_{t+1} according to the state transition function P and obtain the reward r_t. The entire interactive process repeats until the end of the task. In a fully cooperative task, all agents share a common reward function $r_t(s_t, a_t)$: $S \times A \longrightarrow \mathbb{R}$. The joint policy of all agents can be expressed as $\pi = \{\pi^1, \pi^2, \cdots, \pi^N\}$. The ultimate goal of the cooperative task is that all agents coordinate with each other to find an optimal joint policy π^*, satisfying the following condition

$$\pi^* = \text{argmax}_\pi \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[\sum_{t=0}^{+\infty} r_t(s_t, a_t)\right], \tag{8}$$

where $\mathbb{E}_{(s_t,a_t)\sim\rho_\pi}[\sum_{t=0}^{+\infty} r_t(s_t, a_t)]$ is the expected cumulative discount reward. The policy that maximizes the expected discount reward is the optimal policy.

According to the multiagent cooperative task studied in this paper, the elements of POMDP are defined as follows.

4.1.1. Action. The action of each JE includes the selection of jamming links and the corresponding jamming power allocation. For instance, the action of JE j can be denoted by $a_t^j = [p_t^1\ p_t^2 \cdots p_t^M]$, where $0 \le p_t^i(t) \le P_{\max}, 1 \le i \le M$. If the JE chooses to interfere with link i, then $p_t^i(t)$ is in the range of $(0, P_{\max}]$. Otherwise $p_t^i(t)$ equals to 0. Moreover, the constraints of jamming power and links selection can be formulated as

$$\sum_{i=1}^{M} p_t^i \le P_{\max}, \tag{9}$$

$$\sum_{i=1}^{M} \text{sign}\left(p_t^i\right) \le U, \tag{10}$$

where sign $(\cdot)$ denotes the indicator function and sign $(x > 0)$ equals to 1.

Then, the joint jamming action can be defined as

$$a_t = \left(a_t^1, a_t^2, \cdots, a_t^N\right). \tag{11}$$

4.1.2. Global State and Local Observation. Under the condition of battlefield confrontation, it is difficult for the jammer to directly obtain the jamming effect on the enemy, but it can indirectly obtain the jamming effect by intercepting data packets. Referring to the setting of [22], in this paper, the jammer receives feedback about its jamming scheme by observing the acknowledgement/negative acknowledgement (ACK/NACK) packets that are exchanged between early-warning aircrafts and fighter planes. The average number of NACKs gives an estimate of the packet error ratio (PER) which can be used to estimate the symbol error ratio (SER) as $1 - (1 - \text{PER})^{1/N_{\text{sym}}}$, where N_{sym} is the number of symbols in one packet.

In this paper, the local observation z_t^j of each JE contains its jamming scheme a_{t-1}^j at the last time step and the corresponding jamming effect SER^j, which can be expressed as

$$z_t^j = \left[a_{t-1}^j, \text{SER}^j\right], \tag{12}$$

where $\text{SER}^j = (\text{SER}_1^j, \text{SER}_2^j, \cdots, \text{SER}_i^j, \cdots, \text{SER}_M^j)$ and SER_i^j is the SER of link i under the jamming impact of JE j.

The reason the current state designed in this way is that the state needs to contain as many environmental features as possible to improve the generalization performance of the decision network. By comparing a lot of different state designs, we find that the state containing the jamming scheme at the last time step and the corresponding jamming effect can train the decision-making network in a more efficient way.

Then, the global state can be defined by the ensemble of observations of all JEs, which is expressed as

$$s_t = \left(z_t^1, z_t^2, \cdots, z_t^N\right) \in S, \tag{13}$$

where S is the global state space.

4.1.3. Reward Function. In DRL, the reward function can guide the optimization direction of the algorithm. In the confrontation model, the communication links with higher relative importance factors should be jammed preferentially. Then, on the premise of fulfilling the blanket jamming goal, the minimum jamming power should be used as far as possible to save resources and avoid the excessive radiated power affecting the performance of the jammers' communication services or exposing the position of the jammers.

Therefore, the reward function defined in this paper includes the reward of overall blanket jamming and the reward of resource utilization. The total reward function is

$$r(s_t, a_t) = r_T(s_t, a_t) + r_P(s_t, a_t). \tag{14}$$

$r_T(s_t, a_t)$ denotes the overall blanket jamming reward, which is defined as

$$r_T(s_t, a_t) = k_1 + k_2 \cdot \sum_{i=1}^{M} w_i \cdot \left[\text{sign}\left(\text{SER}_i(t) - K_j\right)\right], \tag{15}$$

where k_1 is a negative constant, k_2 is a positive proportionality constant, and the absolute value of k_2 is greater than that of k_1. k_1 is the penalty value given to the agent when the jamming scheme is not effective. $w_i \cdot [\text{sign}\ (\text{SER}_i(t) - K_j)]$ means that jamming reward can be obtained only after the link i is blanket jammed, and the reward is proportional to the important factor of link i.

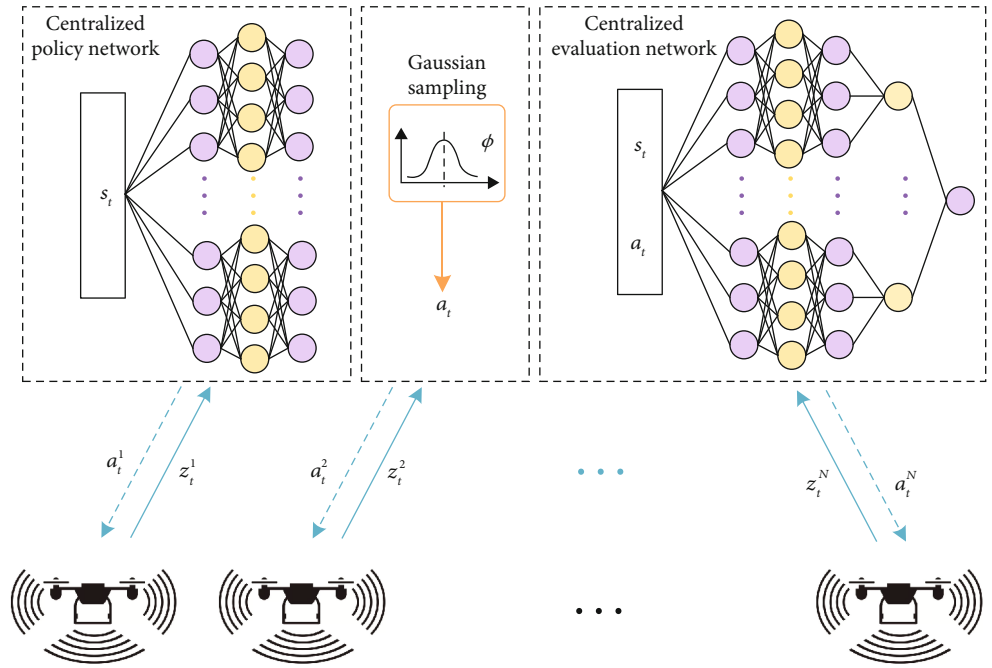


Figure 2: Centralized learning.

$r_P(s_t, a_t)$ represents the reward for resource utilization, which can be expressed as

$$r_P(s_t, a_t) = k_3 \cdot \sum_{i=1}^{M} w_i \cdot \left[\frac{\text{sign}\left(\text{SER}_i(t) - K_j\right)}{P_i(t)}\right], \tag{16}$$

where k_3 is a positive proportionality constant, and $P_i(t)$ is the total jamming power exerted by all JEs on link i. $w_i \cdot [\text{sign}\left(\text{SER}_i(t) - K_j\right)/P_i(t)]$ means jamming reward is inversely proportional to the total jamming power consumed. Remember that the SER and PER are functions of the jammer's action and thereby allow the JE to learn about its jamming scheme.

In the multiple JE cooperative task, all JEs share the common reward, which is used to guide the DRL algorithm to achieve a balance between the overall jamming effect and the optimal utilization of jamming resources.

4.2. Centralized Training and Decentralized Execution. In the multiagent cooperative task, the policy of each agent is associated with the behaviors and interrelationships of other agents. Mainstream learning methods for multiagent task can be divided into the following structures:

One is centralized learning, in which the actions and observations of all agents are ensembled to an expanded action space and observation space. Deep neural networks are used to map the joint observation actions of all agents to a centralized policy function and a centralized value function. Then traditional single-agent reinforcement learning methods are applied directly, as shown in Figure 2. Each JE uploads its observation to the central neural network, and the central policy network decides the jamming action for all JEs uniformly. In this learning mode, the joint observation and state space would expand largely with the increase of the number of agents. In this paper, we assume that the number of JEs and the number of communication links are denoted by N and M, respectively. It can be known from POMDP that the input dimension of the centralized policy network is $2M{\cdot}N$ and the output dimension is $M{\cdot}N$. As the number of JEs increases, the dimensions of the centralized policy network increase as well, and the cost of strategy exploration rises [29].

The second is the independent learning mode [35]. In this learning way, each agent maintains its own policy function and value function independently, and the input of each function only depends on the agent's observation and actions. Each agent makes independent decisions based on its policy network and the network of each JE is updated independently. The input dimension of the policy network is 2M and the output dimension is M, which indicates no correlation with the number of agents N. However, from the view of an individual agent, due to the constant change of policies in the learning process of other agents, it is usual to cause unstable environment state transition and it finds tough for training convergence.

The third is value function decomposition [36]. Based on independent learning, the value functions of each agent are composed, and the global value function is solved in a function approximation way. Then, the value function of each agent is updated from a global perspective with a team value function. This kind of learning method can solve the nonstationarity of the environment. It is only fit for the discrete action space, which is not suitable for our problem.

In this paper, we integrate the advantages of centralized learning and independent learning, adopting the structure of centralized training and decentralized execution (CTDE) [19], as shown in Figure 3.

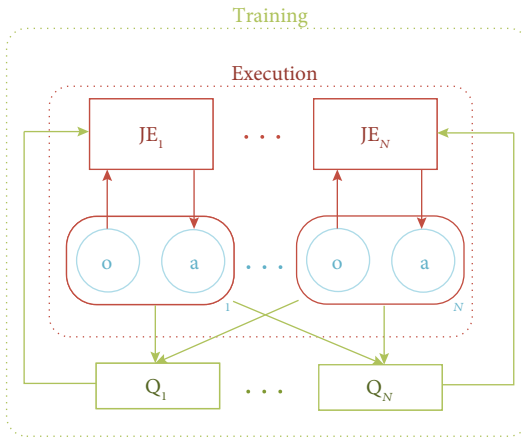


Figure 3: Centralized training and decentralized execution.

Centralized training means that each JE needs to provide the observations and actions of other JE (which can be regarded as global state information) to its evaluation network during training, which can enhance the coordination and cooperation of all JEs through centralized evaluation of joint actions. In addition, the way of centralized training relatively publicizes the policies of other JE, increasing the stability. Furthermore, decentralized execution means that each JE decides jamming actions based on its policy network and observation. Thus, no requirement exists for the central controller to centrally process the joint observation information of each JE.

In this way, the input nodes of the agent's policy network are merely 2M, and the output dimension is M, which reduces $M(N-1)$ compared with that of the centralized learning method. Practical experiences suggest that too large decision-making dimension is one of the important reasons for convergence failure, and the reduction of decision-making dimension can improve the feasibility of the method.

4.3. Cooperation Jamming Resource Allocation Method Based on Multiagent Soft Actor Critic. Under the framework of CTDE, we adopt the optimization route of maximizing the cumulative rewards and policy entropy in SAC [20] to improve the exploration efficiency of each agent in the unknown environment. We combine the policy entropy term with the cumulative rewards in (8), namely,

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[\sum_{t=0}^{+\infty} r_t(s_t,a_t) + \alpha H(\pi(\cdot|s_t))\right], \quad (17)$$

where $H(\pi(\cdot|s_t)) = -\log(\pi(a_{t+1}|s_{t+1}))$ and α is the entropy coefficient.

Policy entropy is the entropy of the action distribution. When the value of policy entropy is high, it means that the current policy has more randomness and has a stronger exploration ability in the unknown environment. Enough exploration can realize the complete learning of the environment model and avoid falling into the local optimum.

To balance the agent's exploration and exploitation, we set the optimization objective function of entropy coefficient α [20], and its value is updated through gradient descent by minimizing the following loss function:

$$J(\alpha) = \mathbb{E}_{a_t\sim\rho_\pi}[-\alpha \log \pi(a_t|s_t) - \alpha\mathscr{H}], \quad (18)$$

where $\mathscr{H}$ is the dimensions of the action space.

In the initial stage, a large value of α refers to the great randomness of the policy; thus, the algorithm has high exploration efficiency, which can avoid falling into the local optimum in the process of optimization. As the agent continues to learn by interacting with the environment, α is adaptively decreased. When α drops to 0, then no entropy exists in (17), and the optimization goal of the agent backs to the traditional maximization of cumulative rewards.

The iterative formula of the value function used in solving the optimal policy recursively can be expressed as follows:

$$Q_v(s_t,a_t) = r_t(s_t,a_t) + \gamma\mathbb{E}\left[Q_v(s_{t+1},a_{t+1}) - \alpha\log\left(\pi_\phi(a_{t+1}|s_{t+1})\right)\right], \quad (19)$$

where γ is the discount factor.

Considering the high decision dimension, this paper uses neural networks to fit the value function and policy function, and Kullback-Leibler's (KL) divergence constraint is used to update the strategy as

$$\pi_{\text{new}} = \arg\min D_{\text{KL}}\left(\pi_\phi(\cdot|s_t)\middle\|\frac{\exp\left((1/\alpha)Q_v^{\pi_{\text{old}}}(s_t,\cdot)\right)}{Z^{\pi_{\text{old}}}(s_t)}\right), \quad (20)$$

where $D_{\text{KL}}(\cdot)$ represents KL divergence constraint, $Q_v^{\pi_{\text{old}}}(s_i,\cdot)$ is the Q function of the original policy, and $Z^{\pi_{\text{old}}}(s_i)$ denotes the logarithmic partition function of the original policy.

The policy optimization based on the above ideas is carried out within each agent under the CTDE framework; then, the multiagent SAC (MASAC) algorithm is proposed. Figure 4 shows the schematic diagram of the MASAC algorithm.

In Figure 4, at each time step t, each JE makes a jamming decision based on its policy network simultaneously and independently and obtains the jamming action $a_t^i = \pi_i(z_t^i)$ under the current local observation. After the joint action $a_t = (a_t^1, a_t^2, \cdots, a_t^N)$ is executed, a shared reward is obtained, and the experience $(s_t, a_t, s_{t+1}, ,r_t)$ gained from interaction with the environment is stored in the common replay buffer (CRB). Moreover, s_t represents the global state and $s_t = (z_t^1, z_t^2, \cdots, z_t^i \cdots, z_t^N)$.When the experiences in the CRB are accumulated to a certain level, a minibatch of samples are randomly selected from the CRB for each JE to train its policy network and evaluation network.

In the process of training, a target network that is identical to the evaluation network is introduced [37] to improve the stability of network training; the sum of the output discounted Q value of the target network, and the shared

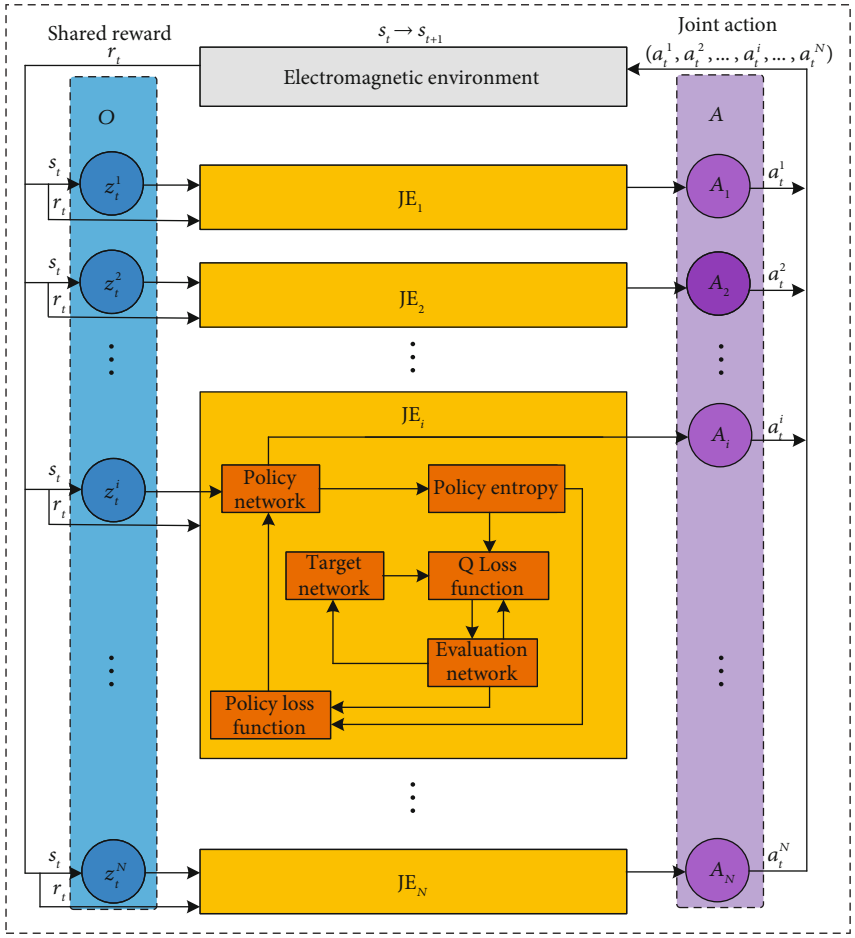


Figure 4: MASAC algorithm diagram.

reward is used as the label of the evaluation network training.

In addition, the twin network structure [38] is adopted in the evaluation network to reduce the overestimation of the action's Q value by the evaluation network. That is, two deep neural networks with the same structure are deployed inside the evaluation network, and the smaller result of their output is taken to carry out Q value iteration and gradient descent during each iteration. In this way, rewrite (19) as

$$Q_{v,\theta_{i,j}}(s_t, a_t) = r_t(s_t, a_t) + \gamma \mathbb{E}\left[\min_{j=1,2} Q_{v,\theta_{i,j}}(s_{t+1}, a_{t+1}) - \alpha log\left(\pi_\phi(a_{t+1}|s_{t+1})\right)\right]. \tag{21}$$

Then, we use the output of the twin target network to calculate the target value of the joint jamming action for agent i, which is defined as

$$Q_{v,\theta_{i,j},\text{tar}}\left(s_t, a'\right) = r\left(a', s_t\right) + \gamma\left[\min_{j=1,2} Q_{v,\theta_{i,j}}\left(s_t, a'\right) - a\ \log \pi_\phi\left(a'|s_{t+1}\right)\right], \tag{22}$$

where a' is the joint jamming action selected under the next state s_{t+1}. Specifically, it can be expressed as $a' = (a_{t+1}^1, a_{t+1}^2, \cdots, a_{t+1}^i, \cdots, a_{t+1}^N)$, and $a_{t+1}^i \sim \pi_{i=1,2,\cdots,N}(\cdot|s_{t+1}, \phi_i)$.

Moreover, the twin evaluation network of JE i is updated by minimizing the loss function $J_{Q_v}(\theta_{i,j})$, which is given as

$$J_{Q_v}(\theta_{i,j}) = \mathbb{E}_{(s_k,a_k)\sim B}\left[\frac{1}{2}\left(Q_{v,\theta_{i,j}}(s_k, a_k) - Q_{v,\theta_{i,j},\text{tar}}\left(s_k, a'\right)\right)^2\right] \quad \text{for } j = 1, 2. \tag{23}$$

Next, based on (24) and (25), the policy network of agent i can be optimized with stochastic gradients.

$$\nabla_{\phi_i} \frac{1}{|B|} \sum_{s_k \in B}\left(\min_{j=1,2} Q_{v,\theta_{i,j}}\left(s_k, a_{\phi_i}(s_k)\right) - a\ \log \pi_{\phi i}\left(a_{\phi_i}(s_k)|s_k\right)\right)^2, \tag{24}$$

$$\phi_i \longleftarrow \phi_i - \nabla_{\phi_i} J_\pi(\phi_i), \tag{25}$$

where $a_{\phi_i}(s_k) \sim \pi_{i=1,2,\cdots,N}(\cdot|s_k, \phi_i)$.

- Initialize the CRB.
- Initialize policy network $\pi_i(z_i, \phi_i)$, twin evaluation network $Q_{i,1}(s, a_t^1, a_t^2, \cdots, a_t^N, \theta_{i,1})$ and $Q_{i,2}(s, a_t^1, a_t^2, \cdots, a_t^N, \theta_{i,2})$ for each JE i with weights ϕ_i 、 $\theta_{i,1}$ 、 $\theta_{i,2}$, respectively.
- Initialize twin target network $\bar{Q}_{i,1}(s, a_t^1, a_t^2, \cdots, a_t^N, \bar{\theta}_{i,1})$ and $\bar{Q}_{i,2}(s, a_t^1, a_t^2, \cdots, a_t^N, \bar{\theta}_{i,2})$ for each JE with weights $\bar{\theta}_{i,1}$ 、 $\bar{\theta}_{i,2}$, respectively.
- Training episode =1.
- **While** Training episode ≤ E **do**
- Initialize the environment state $S(t) = (0, 0, \ldots, 0)$.
- **for** time step t =1, 2, …, $T_{\max}$
- Each JE i selects the jamming action $a_t^i \sim \pi_i(a_t^i | z_t^i, \phi_i)$ according to the current observation z_t^i.
- Obtain and carry out the joint jamming action a_t, $a_t = (a_t^1, a_t^2, \cdots, a_t^N)$, then each JE i obtains the shared reward r_t and achieves the next observations z_{t+1}^i.
- The experience from all JEs is stored in CRB:
- **If** the capacity of CRB is larger than β, then the training process begins:
- Stochastically Sampling mini-batch of experiences $(s_k, a_k^1, a_k^2, \cdots, a_k^i, \cdots, a_k^N, s_{k+1}, , r_k)$ from CRB.
- **for** each JE i =1, 2, …, N
- Update the weight $\theta_{i,1}$ and$\theta_{i,2}$ of twin evaluation network with (23)
- Update the weight ϕ_i of the policy network by (24) and (25)
- Soft update the weight $\bar{\theta}_{i,1}$ and $\bar{\theta}_{i,2}$ of twin target network through (26)
- **end for**
- **end If**
- **end for**
- **end while**

Algorithm 1: Decentralized allocation of communication jamming resource based on MASAC.

Finally, the twin target network parameters are updated in a soft updating manner as

$$\theta_{i,j} \longleftarrow \tau \cdot \theta_{i,j} + (1 - \tau) \cdot \theta_{i,j} \text{ for } j = 1, 2. \tag{26}$$

The decentralized allocation of jamming resources based on MASAC is summarized in Algorithm 1. At the beginning of our algorithm, the weights of the policy network, twin evaluation network, and twin target network in each JE are initialized randomly. Note that the training procedure consists of E episodes, each of which comprises of $T_{\max}$ time steps. At the beginning of each training episode, initialize the environment state $S(t)$ firstly. Then, at time step t, each JE i chooses jamming action a_t^i simultaneously and independently based on its current observation according to the policy network $\pi_i(a_t^i | z_t^i, \phi_i)$. After the joint action a_t is executed, each JE obtains a shared reward and the next observation. We store the interaction experiences in the CRB. When the experiences are accumulated to a certain level and randomly sample a minibatch of samples to train the deep neural network for each JE, it is worth noting that the experiences in the CRB contain global information.

After the process of training, each JE obtains a decentralized jamming policy; then, each JE only relies on local observation to decide the jamming action. The schematic diagram of the decentralized policy is shown in Figure 5.

5. Computational Complexity Analysis and Convergence Proof

In this section, we first compare the computational complexity of our method with the centralized learning method in [21]. Then, we present the proof of the convergence of our method.

5.1. Computational Complexity Analysis. The computational complexity of the method in this paper is mainly determined by the network structure of the evaluation network and policy network. Assume that the evaluation network is a fully connected network with H_e hidden layers, and the h-th $(1 < h < H_e)$ hidden layer contains n_h^e neurons. The input layer is determined by the dimensions of the joint state and the joint action, which is 3 MN. The number of neurons in the output layer is 1. Therefore, the total number of neurons in the evaluation network is $3\text{MN}n_1^e + \sum_{h=2}^{H_e} n_{h-1}^e n_h^e + n_{H_e}^e$. Similarly, suppose that the hidden layer of the policy network is a fully connected network with H_p layer, and the h-th hidden layer $(1 < h < H_p)$ contains n_h^p neurons. The input layer is determined by the local state dimension, which is 2 M. And the number of neurons in the output layer is M. Therefore, the total number of neurons in the policy network is $2Mn_1^p + \sum_{h=2}^{H_p} n_{h-1}^p n_{H_p}^p + n_{H_p}^p M$. If the computation complexity of training a neuron weight is W, then the computational complexity of the method in this paper can be expressed as $O(W[3\text{MN}n_1^e + \sum_{h=2}^{H_e} n_{h-1}^e n_h^e + n_{H_e}^e + 2Mn_1^p + \sum_{h=2}^{H_p} n_{h-1}^p n_{H_p}^p + n_{H_p}^p M])$. The computational complexity of the algorithm is positively correlated with the number of JEs, and the number of communication links, i.e., N and M.

The complexity of the evaluation network in the centralized learning algorithm is the same as that of our algorithm. The difference lies in that the policy network's input is based on the observation of all JEs and outputs the schemes of all

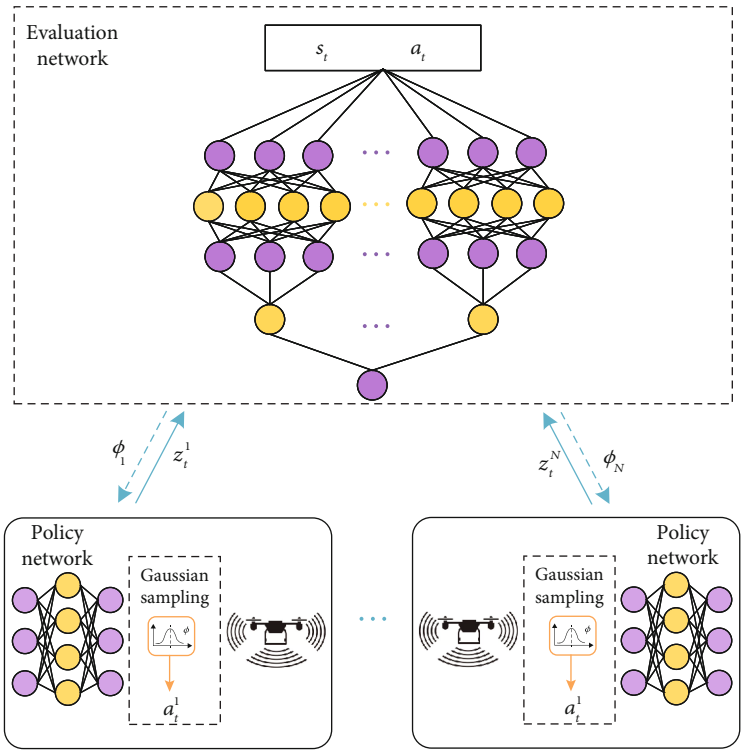


Figure 5: The training process for decentralized policy of MASAC.

JEs. Thus, the number of input neurons is 2 MN, and the number of neurons in the output layer is MN.

In the same way, the computational complexity of the centralized learning algorithm can be obtained as $O(W[3\mathrm{MN}n_1^e + \sum_{h=2}^{H_e} n_{h-1}^e n_h^e + n_{H_e}^e + 2\mathrm{MN}n_1^p + \sum_{h=2}^{H_p} n_{h-1}^p n_{H_p}^p + n_{H_p}^p \mathrm{MN}])$. By comparison, the complexity of the centralized learning algorithm is $O(W[2M(N-1)n_1^p + n_{H_p}^p M(N-1)])$, higher than that of the algorithm we proposed. In addition, the decision-making dimension of centralized learning is $2M(N-1)$, which is also higher than that of our decentralized algorithm.

5.2. Proof of Algorithm Convergence. For the convergence analysis of the algorithm in this paper, we derive the following Theorem.

Theorem 1. *In the joint policy set Π, when the dimension of action space is finite, i.e., $|A| < \infty$, there is a joint policy $\pi \in \Pi$, which can converge to the best joint strategy π^*, and $Q^{\pi^*}(s_t, a_t) \geq Q^{\pi}(s_t, a_t), \forall \pi \in \Pi$ can be guaranteed.*

Proof. Divide the policy iteration optimization process into two parts, i.e., policy evaluation and policy improvement.

In policy evaluation, the reward combined with policy entropy can be defined as

$$r_\pi(s_t, a_t) \triangleq r(s_t, a_t) + \mathbb{E}_{s_{t+1}\sim p}[\mathscr{H}(\pi(\cdot|s_{t+1}))]. \tag{27}$$

Then, to look more concise, we rewrite (19) as

$$\begin{aligned} Q_v(s_t, a_t) = r_\pi(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}\sim p, a_{t+1}\sim \rho_\pi} \\ \cdot [Q_v(s_{t+1}, a_{t+1}) + \mathscr{H}(\pi(\cdot|s_{t+1}))]. \end{aligned} \tag{28}$$

According to the Bellman iteration equation, we have

$$\begin{aligned} Q_v^{\pi_{\mathrm{old}}}(s_t, a_t) &= r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}\sim p}[V^{\pi_{\mathrm{old}}}(s_{t+1})], \\ &\leq r(s_t, a_t) + \lambda \mathbb{E}_{s_{t+1}\sim p}\left[\gamma \mathbb{E}_{a_{t+1}\sim \pi_{\mathrm{new}}}\left[-\log \frac{Q_v^{\pi_{\mathrm{old}}}(s_{t+1}, a_{t+1})}{\pi_{\mathrm{new}}(a_{t+1}|s_{t+1})}\right]\right], \\ &\vdots \\ &\leq Q_v^{\pi_{\mathrm{new}}}(s_t, a_t). \end{aligned} \tag{29}$$

We denote the policy in the i-th iteration as π_i, it is seen that the sequence of value functions $\{Q_v^{\pi^1}, Q_v^{\pi^2}, \cdots, Q_v^{\pi^i}, \cdots\}$ is monotonically increasing. Since the reward and the entropy of the policy are bounded, the sequence can converge to an optimal value function.

In the policy improvement, based on (20), we can further obtain

$$\begin{aligned} \pi_{\mathrm{new}}(\cdot|s_t) &= \arg \min_{\pi' \in \Pi} D_{\mathrm{KL}}\left(\pi'(\cdot|s_t) \middle\| \exp\left(\begin{matrix} Q_v^{\pi_{\mathrm{old}}}(s_t, \cdot) \\ -\log Z^{\pi_{\mathrm{old}}}(s_t) \end{matrix}\right)\right) \\ &= \arg \min_{\pi' \in \Pi} J_{\pi_{\mathrm{old}}}\left(\pi'(\cdot|s_t)\right). \end{aligned} \tag{30}$$

For all policies $\pi \in \Pi, \pi \neq \pi^*$, it is apparent to find that $J_{\pi_{\text{old}}}(\pi_{\text{new}}(\cdot|s_t)) \leq J_{\pi_{\text{old}}}(\pi_{\text{old}}(\cdot|s))$, using the iterative proof of the policy evaluation. Then, we can obtain $Q_v^{\pi^*}(s_t, a_t) \geq Q_v^{\pi}(s_t, a_t)$ for all state-action pairs (s_t, a_t). Clearly, Q_v^{π} of other policies in Π are smaller than the policy obtained after convergence. Thus, π^* is the optimal policy in Π. □

6. Experimental Simulation Results and Analysis

In this section, the performance of the proposed MASAC approach is numerically evaluated. The simulation is executed on a server with Intel Xeon Platinum 8260 CPU and NVIDIA Tesla V100 GPU with a memory size of 32GB. The simulation of the confrontation scenario and algorithm programs is constructed in Python 3.7, and the deep neural networks are built and trained by using PyTorch with version 1.4.0.

6.1. Simulation Parameter Setting. In the simulation scenario, the C2 terminal of the jammer is possessed of three JEs, and each JE can simultaneously interfere with two links. There are several early-warning aircrafts in the mission area, providing five communication links for them. The intelligence information of each link obtained after communication reconnaissance intelligence analysis is summarized in Table 1. The types of FF and FH mean fixed frequency communication mode and frequency hopping communication mode, respectively. To achieve the overall blanket jamming over the mission area, the minimum distance between early-warning aircrafts and the receiver in the mission airspace is taken as the communication signal propagation distance, and the maximum distance between the JEs and the receiver in the mission airspace is taken as the jamming signal propagation distance.

The parameters of the confrontation scenario are shown in Table 2.

In the MASAC method, both the policy and evaluation networks are fully connected networks with three hidden layers (256, 128, and 128 neurons). We use the Adam optimizer [39], and the activation function is rectified linear unit (ReLU); the output layer of the policy network is tanh. More detailed parameters of the MASAC approach are presented in Table 3.

6.2. Analysis of Experimental Results. In this subsection, the performance of the decentralized MASAC method proposed in this paper is compared with centralized allocation based on DDPG (abbreviated to CA-DDPG below) in [21] and MADDPG in [19]. Firstly, we compare the overall blanket jamming efficiency of the three methods against all communication links, and their learning ability of collaborative policy are investigated.

Figure 6 shows the learning curves of overall blanket jamming on all communication links among different approaches when the blanket jamming coefficient is 2. It is clearly seen from the learning curves that the learning rates of decentralized MASAC and MADDPG algorithms are both relatively faster, and their overall blanket jamming success rate is greatly improved after 300-500 trainings, among which MASAC has the maximum overall success rate up to more than 0.85. Since MADDPG only seeks to maximize the cumulative rewards without maximizing the entropy of the policy simultaneously, the exploration is slightly insufficient,

Table 1: Intelligence information of each communication link.

Link	Type	Center frequency of FF (MHz)	The length of FH frequency set	Importance factor
1	FF	230.25	—	0.2
2	FF	275.5	—	0.3
3	FH	—	160	0.6
4	FF	366	—	0.5
5	FH	—	200	0.8

Table 2: Simulation parameters of the confrontation scenario.

Parameters	Value
The number of JEs (N)	3
The number of communication links (M)	5
Maximum number of simultaneous jamming (U)	2
Total jamming channel bandwidth (B_j)	2 MHz
The bandwidth of FF link (B_d)	50 kHz
Frequency interval of FH link (f_i)	25 kHz · n(n = 1,2,3,4)
Maximum jamming signal power ($P_{\max}$)	70 dBm
Communication transmitting power (Pc)	55 dBm
Noise power (σ^2)	-85 dBm
Gain of communication link (G_c)	8 dB
Gain of jamming link (G_j)	10 dB
Minimum distance of communication links (R_c)	110 km
Maximum distance of jamming links (R_j)	300 km
Blanket jamming coefficient (K_j)	2

Table 3: Main hyper parameters of MASAC.

Parameters	Value
k_1	-0.25
k_2	5
k_3	1
Training episodes (E)	5000
Total steps of each episode ($T_{\max}$)	500
Soft updating rate (τ)	0.01
Capacity of CRB	2^{17}
Minibatch size (B)	256
Initial value of entropy coefficient (α)	1
Discount factor (γ)	0.98
Learning rate of policy network	0.001
Learning rate of evaluation network	0.003
Training threshold of CRB (β)	1024

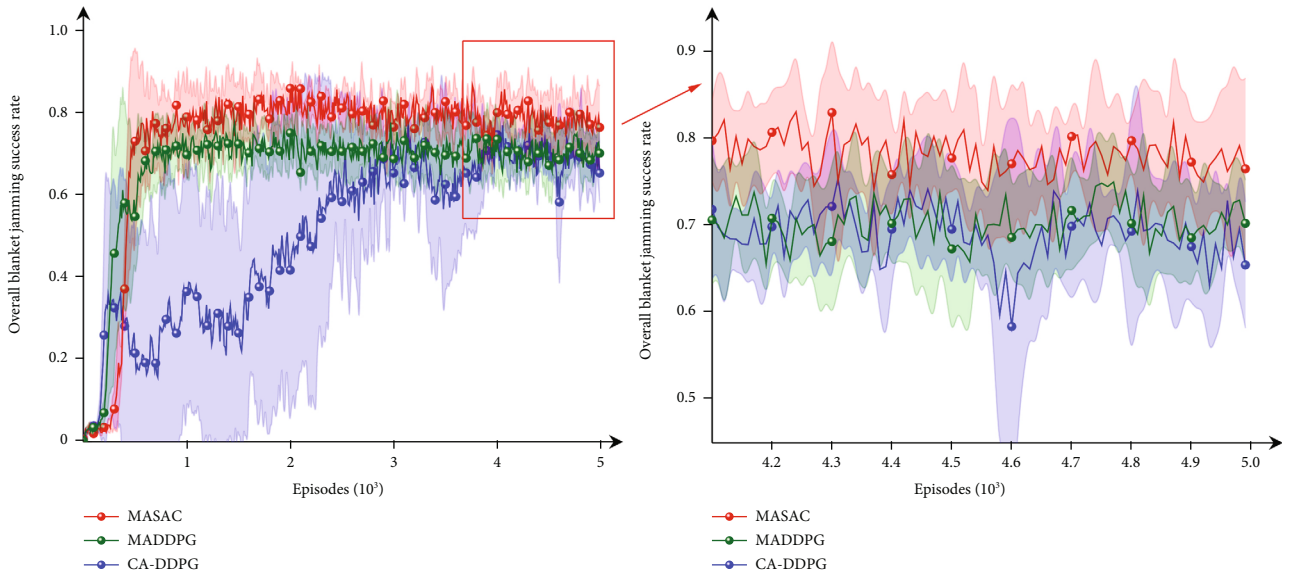


Figure 6: Comparison of overall blanket jamming rate among different approaches under the condition of the blanket jamming coefficient is 2.

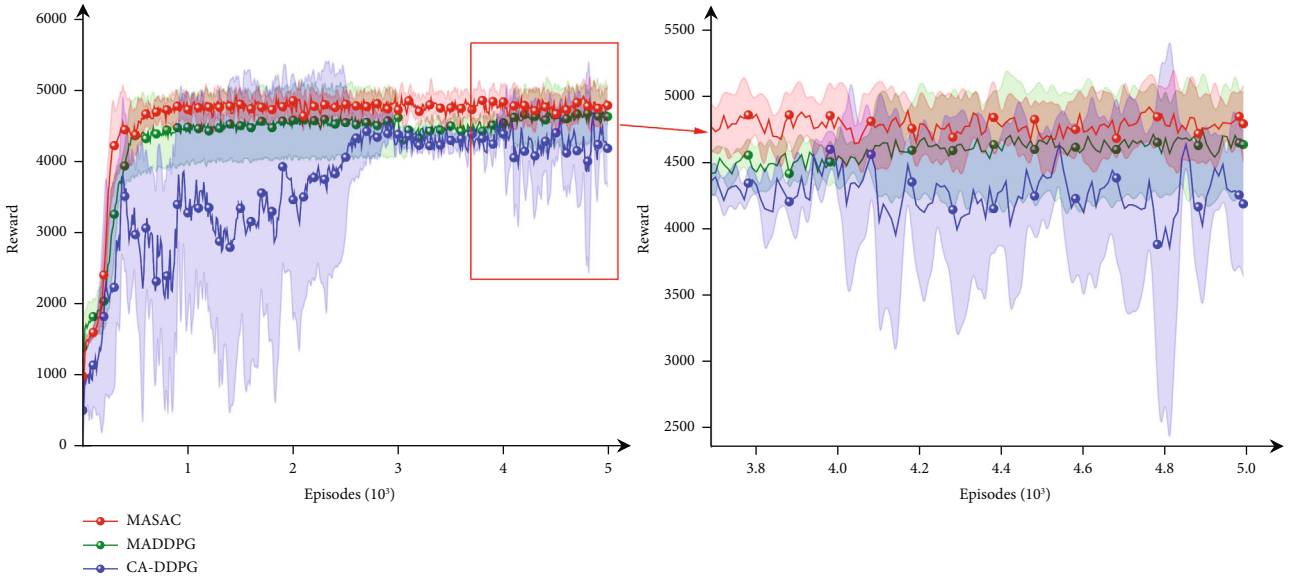
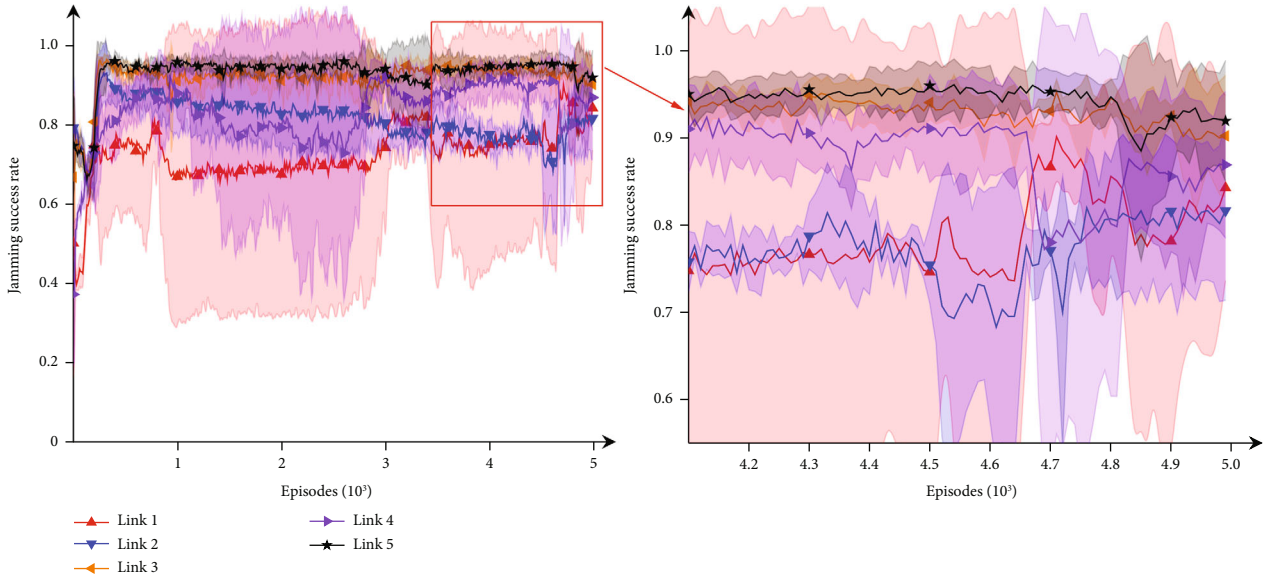


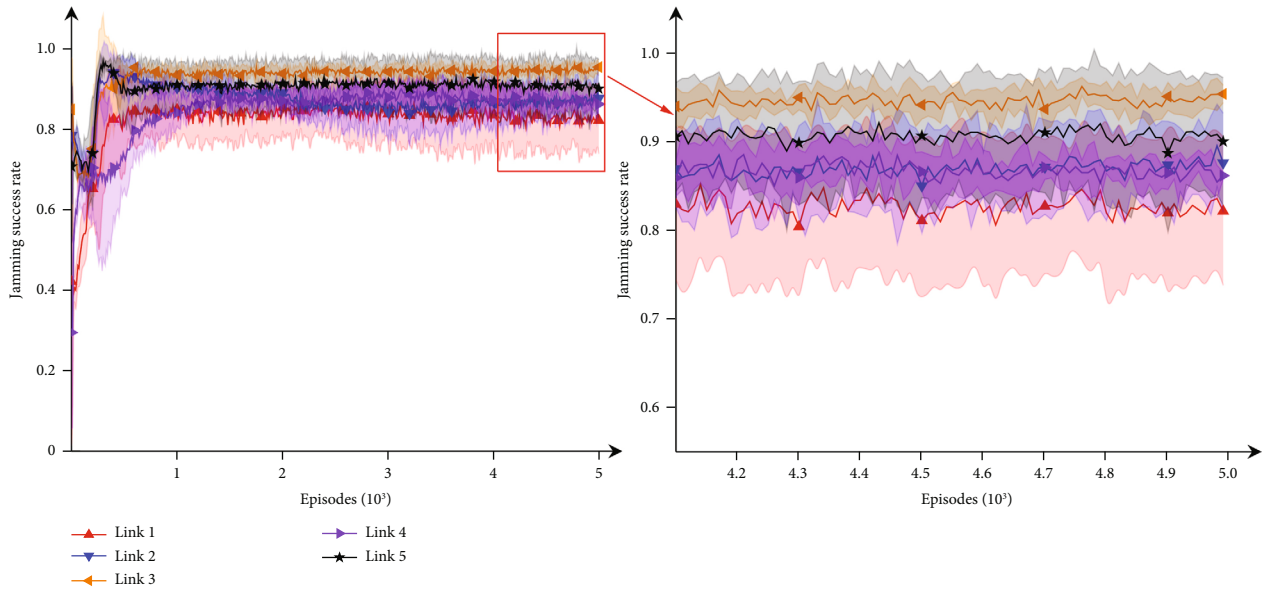
Figure 7: Comparison of cumulative rewards among different approaches under the condition of the blanket jamming coefficient is 2.

and it is easy to fall into the local optimum. Thus, the maximum success rate of overall blanket jamming is comparable to that of MASAC. Besides, at the initial stage of CA-DDPG, the joint jamming action space of each JE is larger, the exploration time is longer, and the learning process is more volatile. In addition, the deep deterministic strategy adopted by CA-DDPG itself has poor exploration ability and is not efficient enough to explore unknown jamming actions. There is an overestimation of jamming actions' value which makes JEs overly optimistic about the jamming actions taken, and it is not conducive to the learning of the optimal policy. Finally, the blanket success rate of CA-DDPG is around 0.7 merely. It is implied that combining the action space of each JE to make centralized decisions will increase the complexity of the decision-making, which is inferior to the decentralized algorithm both in terms of the convergence speed and the jamming effect after convergence. Moreover, the performance of MASAC on cooperative tasks is better than MADDPG, which also verifies that the overall performance of the decentralized algorithm can be improved to a certain extent after the maximum policy entropy criterion is applied.

Figure 7 compares the cumulative rewards of different approaches. Similar to the result in Figure 6, MASAC outperforms other approaches. In addition, it is noticed that the shaded part in Figures 6 and 7 represent the fluctuation range according to the results of 500 repeated experiments. And they both indicate that the overall oscillation amplitude

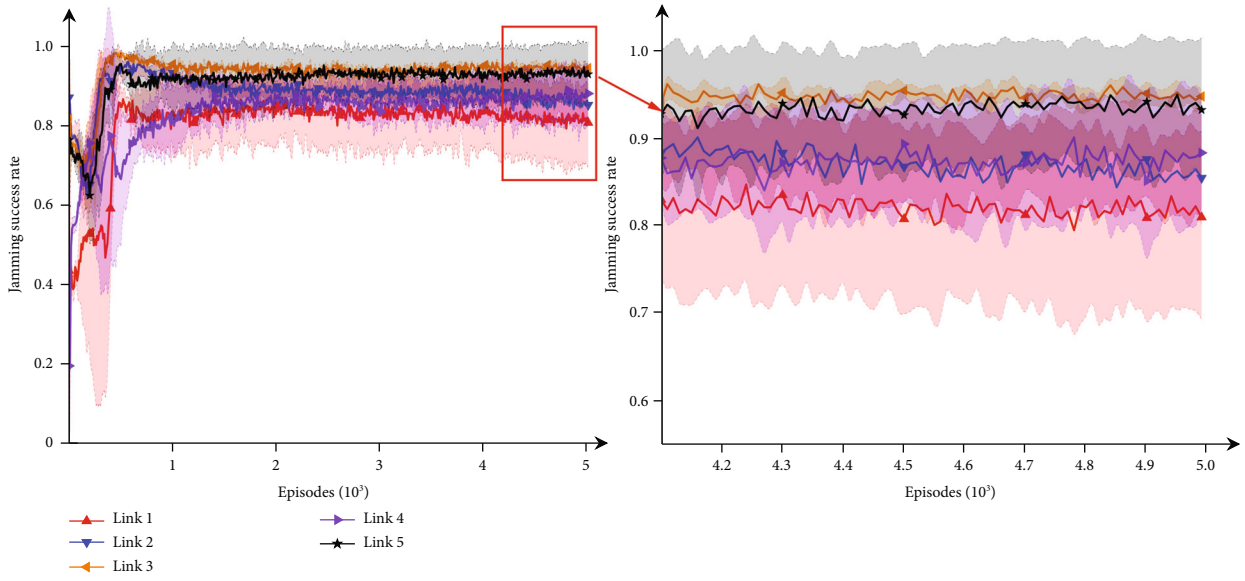


(a) CA-DDPG



(b) MADDPG

Figure 8: Continued.



(c) MASAC

Figure 8: Blanket jamming success rate of each link of different approaches under the condition of the blanket jamming coefficient is 2.

of CA-DDPG is relatively large, whereas the learning process of MASAC and MADDPG is relatively stable. The reason is that CA-DDPG is centralized controlling, which explores and optimizes the policy in a higher dimensional expanded action space. The high-dimensional space increases the difficulty of decision-making, whereas MASAC and MADDPG are both decentralized methods, of which decision-making dimensions depend on the action space of a single JE. Thus, the dimension is relatively smaller, leading to higher learning efficiency.

Next, the blanket jamming performance of different algorithms against each link is presented in Figure 8.

From the jamming curves of the three approaches in Figure 8, it is observed that all approaches can give priority to jam on link 5 and link 3 which have higher importance factors. However, for link 1 and link 2 with low importance factor, the average jamming success rate of CA-DDPG is at a low level. In comparison, MASAC and MADDPG can better coordinate the jamming resource allocation of each JE, and the average jamming success rate of each link is improved compared with CA-DDPG. This indicates that the decentralized algorithm is more conducive to the learning of collaborative strategies. In addition, the learning process of the decentralized algorithm is more stable according to the shaded parts.

In Table 4, we compare the maximum success jamming rate of each link among different approaches. The maximum jamming success rate refers to the maximum value that can be achieved in 5000 training episodes. CA-DDPG's jamming success rate for Link 1 with the minimum importance factor can arrive to 0.91, which is higher than MASAC and MADDPG. Yet, CA-DDPG can only reach 0.94 towards Link 5 with the maximum importance factor, whereas MASAC and MADDPG can both reach above 0.95, and MASAC even reaches close to 0.97. The data in the last row of the Table 4 represents the maximum success rate of

Table 4: The comparison of the maximum success jamming rate among different approaches of each link.

Type	Method		
	MASAC	MADDPG	CA-DDPG
Link 1	0.8616	0.8692	0.9102
Link 2	0.9448	0.9632	0.9308
Link 3	0.9616	0.9866	0.9652
Link 4	0.8988	0.9106	0.9276
Link 5	0.9676	0.9574	0.9468
Simultaneously jamming all links	0.8694	0.7824	0.7760

jamming all links simultaneously, corresponding to the maximum values of each curve in Figure 6. These comparisons further verify that MASAC can better interfere with links with higher relative importance factors preferentially under the premise of blanket jamming.

Then, we compare the total jamming power allocated to all links of different approaches and investigate the performance of different approaches on minimizing resource utilization while achieving overall blanket jamming, as shown in Figure 9.

When the sum of maximum power of three JEs is set as 75.2 dBm, all approaches can reduce the resource utilization partly. The final jamming power allocated to each link by MASAC is around 74.8 dBm, which saves a certain amount of jamming power compared with full-power jamming and also saves more resource than CA-DDPG and MADDPG algorithms. In a battlefield environment, reducing the jammer's own radiation power under the premise of blanket jamming can alleviate the impact on own communication performance and avoid exposing the own position or being

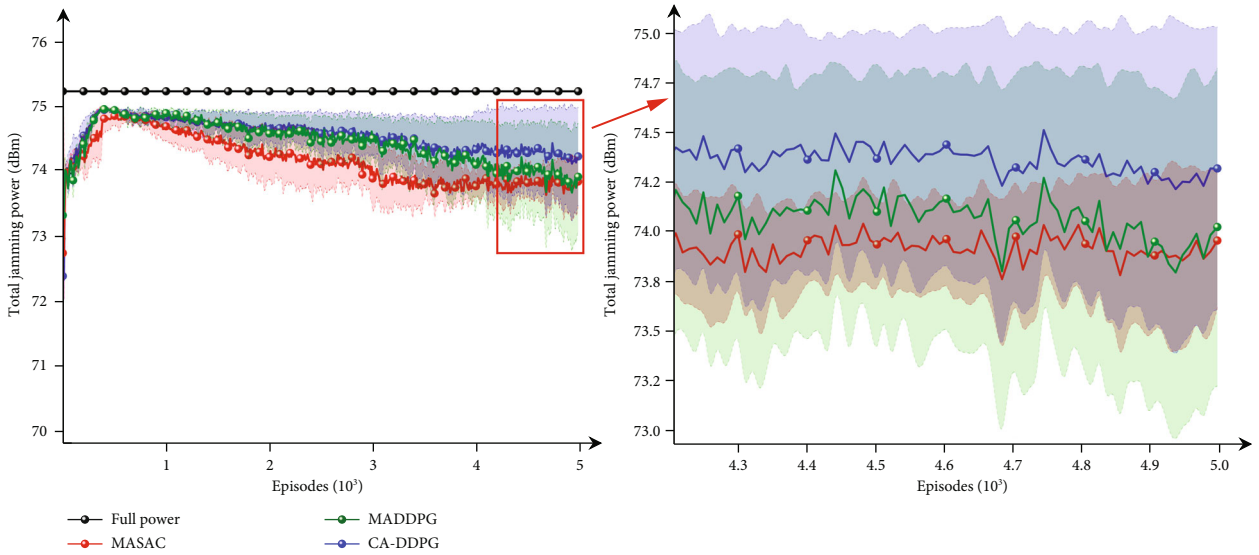


Figure 9: Comparison of total jamming power allocated to all links.

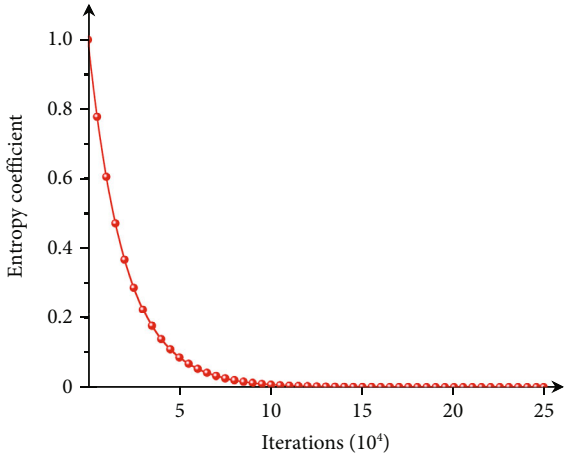


Figure 10: The changing curve of policy entropy coefficient.

detected due to excessive power, which can reduce the risk of antiradiation strikes as well.

Figure 10 is the curve of policy entropy coefficient α during the training process. The initial value of the entropy coefficient is 1, and the entropy coefficient is adaptively decreasing with the optimization of the policy during the training process. The small entropy coefficient infers the small weight of policy entropy in the objective function. When the number of iterations is close to 12000, the entropy coefficient drops to 0 gradually, indicating that the MASAC algorithm has reached a relatively sufficient stage in the exploration of the environment. It is no longer necessary to consider the impact of policy entropy and instead makes full use of the learned environment knowledge.

Finally, the maximum overall blanket jamming rates that the three approaches can achieve under different blanket jamming coefficient conditions are evaluated, as shown in Figure 11.

As the blanket coefficient increases, more resources are required for blanket jamming for the same communication links. Under the condition of limited resources, it is necessary to coordinate the allocation of jamming resources of JEs more finely. In Figure 11, the overall blanket jamming success rates of the three algorithms all show a downward trend. When the blanket jamming coefficient is 2, the overall success rate of MASAC is 12.5% higher than that of CA-DDPG. When the blanket jamming coefficient is 4, the overall blanket success rate of MASAC is relatively 16.8% higher, indicating that the blanket jamming success rate gap between MASAC and CA-DDPG with $K_j = 4$ is bigger than that with $K_j = 2$. We can find that under the condition of a larger blanket jamming coefficient, the centralized algorithm is less effective in allocating the jamming resources of each JE. The reason lies in that the form of centralized decision-making by a single agent creates a higher dimensional jamming action space, and it is intractable to coordinate the jamming links selection and jamming power allocation for

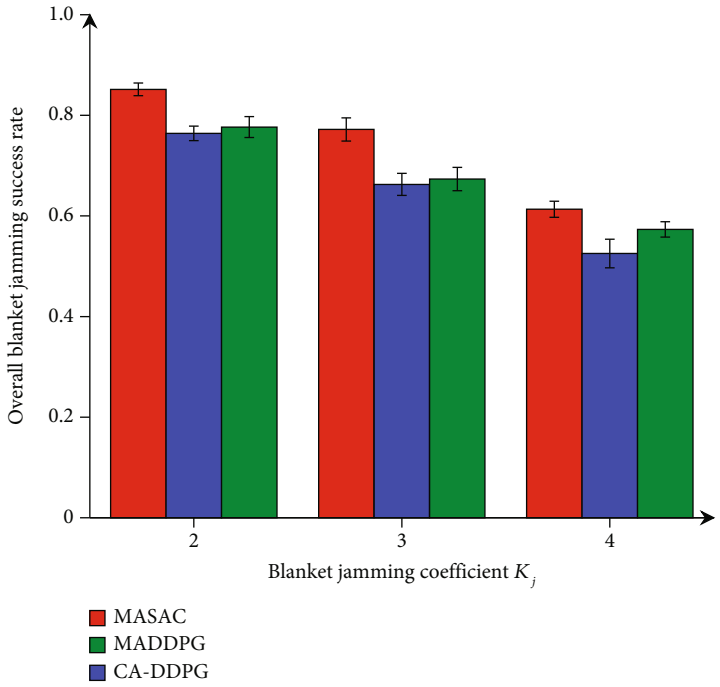


Figure 11: The maximum overall success rate of blanket jamming under different blanket jamming coefficients.

Table 5: The comparison of the maximum blanket jamming rate under different confrontation scenarios and the blanket jamming coefficient is 2.

Method	Scenarios		
	3 JEs vs. 5 links	6 JEs vs. 9 links	9 JEs vs. 14 links
MASAC	0.8694	0.8375	0.7213
MADDPG	0.7824	0.7261	0.6754
CA-DDPG	0.7760	0.5367	0.3862

all JEs. The decentralized strategies can reduce the decision-making dimension which merely depends on single JE through multiagent cooperation. The method of training each JE's policy network through global information can relatively schedule the jamming resources of each JE more efficiently, which implies that the cooperative resource allocation capability of MASAC is better than that of CA-DDPG in larger action space.

Table 5 compares the maximum blanket jamming success rate achieved by each algorithm under different confrontation scenarios and the blanket jamming coefficient is 2.

As can be seen from Table 5, when the number of JEs and the number of communication links increases, the performance of the three algorithms decreases to varying degrees. The performance of CA-DDPG algorithm decreases the most, indicating that when the number of jamming devices and communication links increases, the centralized algorithm is difficult to cope with that situation. However, the overall jamming success rate of MASAC and MADDPG can still maintain around 0.7, and MASAC is 4.6 percentage points higher than MADDPG algorithm. At the same time, it also shows that the algorithm proposed in this paper still has relatively better scalability in the multiagent scenario.

7. Conclusions

In this paper, we have proposed a multiagent reinforcement learning method to the field of jamming resource allocation in CEW novelly, based on MASAC and combining the framework of centralized training and decentralized execution with a soft actor-critic method. The algorithm can learn the optimal control policy of jamming link selection and jamming power allocation by constantly interacting with the environment without accurate prior information. In the learning process, we use the optimization idea of maximizing policy entropy to improve the exploration ability. And the algorithm can learn a more scalable decentralized policy with lower computational complexity. The simulation results demonstrate that the performance of the proposed algorithm exceeds the centralized allocation method and the MADDPG-based method, with a faster learning rate and stronger stability when the decision space is large.

In the future, we will try to introduce value decomposition networks and use a shared policy network to improve the adaptability of the algorithm in large-scale multiagent systems and further improve the practicality of the algorithm.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

The paper was funded by the National Natural Science Foundation of China (Grant No. 61906156).

References

[1] D. Baker, "Advances in communications electronic warfare," in *2006 Canadian Conference on Electrical and Computer Engineering*, pp. 52–55, Ottawa, ON, Canada, 2006.

[2] F. A. Butt and M. Jalil, "An overview of electronic warfare in radar systems," in *2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE)*, pp. 213–217, Konya, Turkey, 2013.

[3] A. Orduyilmaz, M. Ispir, M. Serin, and M. Efe, "Ultra wideband spectrum sensing for cognitive electronic warfare applications," in *2019 IEEE Radar Conference (RadarConf)*, pp. 1–6, Boston, MA, USA, 2019.

[4] B. Zhang and W. Zhu, "Research on decision-making system of cognitive jamming against multifunctional radar," in *2019 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, pp. 1–6, Dalian, China, 2019.

[5] S. You, M. Diao, and L. Gao, "Deep reinforcement learning for target searching in cognitive electronic warfare," *IEEE Access*, vol. 7, pp. 37432–37447, 2020.

[6] K. Gomadam, V. Cadambe, and S. A. Jafar, "Approaching the capacity of wireless networks through distributed interference alignment," in *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, pp. 1–6, New Orleans, LA, USA, 2008.

[7] Y. E. Sagduyu, R. A. Berry, and A. Ephremides, "Jamming games in wireless networks with incomplete information," *IEEE Communications Magazine*, vol. 49, pp. 112–118, 2011.

[8] S. Bayram, N. D. Vanli, B. Dulek, and I. Sezer, "Optimum power allocation for average power constrained jammers in the presence of non-Gaussian noise," *IEEE Communications Letters*, vol. 16, pp. 1153–1156, 2012.

[9] C. Xu, M. Sheng, X. Wang, and C. X. Wang, "Distributed subchannel allocation for interference mitigation in OFDMA femtocells: a utility-based learning approach," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 6, pp. 2463–2475, 2014.

[10] S. Amuru and R. M. Buehrer, "Optimal jamming strategies in digital communication-impact of modulation," in *2014 IEEE Global Communications Conference*, pp. 1619–1624, Austin, TX, USA, 2014.

[11] 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 *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8158–8162, Florence, Italy, 2014.

[12] S. Amuru and R. M. Buehrer, "Optimal jamming against digital modulation," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 10, pp. 2212–2224, 2015.

[13] C. Zhong, J. Yao, and J. Xu, "Secure UAV communication with cooperative jamming and trajectory control," *IEEE Commun. Lett.*, vol. 23, no. 2, pp. 286–289, 2018.

[14] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, p. 1054, 1998.

[15] J. Schmidhuber, "Deep learning in neural networks: an overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[16] D. Sliver, A. Huang, C. J. Maddison et al., "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 787, pp. 484–489, 2016.

[17] O. Vinyals, I. Babuschkin, W. M. Czarnecki et al., "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature.*, vol. 575, pp. 350–354, 2019.

[18] N. C. Luong, D. T. Hoang, S. Gong et al., "Applications of deep reinforcement learning in communications and networking: a survey," *IEEE Communications Surveys & Tutorials*, vol. 21, pp. 3133–3174, 2019.

[19] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multiagent actor-critic for mixed cooperative-competitive environments," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6379–6390, Long Beach, 2017.

[20] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning*, pp. 1861–1870, Stockholm, 2018.

[21] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2239–2250, 2019.

[22] S. Amuru, C. Tekin, M. Schaar, and R. M. Buehrer, "Jamming bandits–a novel learning method for optimal jamming," *IEEE Transactions on Wireless Communications*, vol. 15, pp. 2792–2808, 2016.

[23] S. S. Zhuansun, J. Yang, and H. Liu, "An algorithm for jamming strategy using OMP and MAB," *EURASIP Journal on Wireless Communications and Networking*, vol. 1, Article ID 85, 2019.

[24] Y. Y. Li, X. M. Wang, D. X. Liu et al., "On the performance of deep reinforcement learning-based anti-jamming method confronting intelligent jammer," *Applied Sciences*, vol. 9, pp. 1361–1375, 2019.

[25] D. Zhao, Q. Hao, B. Song, B. Han, and M. Guizani, "A graph convolutional network-based deep reinforcement learning approach for resource allocation in a cognitive radio network," *Sensors*, vol. 20, pp. 5216–5239, 2020.

[26] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 310–323, 2019.

[27] H. Zhang, N. Yang, W. Huangfu, K. Long, and V. C. M. Leung, "Power control based on deep reinforcement learning for spectrum sharing," *IEEE Transactions on Wireless Communications*, vol. 19, no. 6, pp. 4209–4219, 2020.

[28] K. Zhang, Z. Yang, and T. Basar, "Multi-agent reinforcement learning: a selective overview of theories and algorithms," 2019, https://arxiv.org/abs/1911.10635.

[29] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: a review of challenges, solutions, and applications," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3826–3839, 2021.

[30] D. Guo, L. Tang, X. Zhang, and Y. C. Liang, "Joint optimization of handover control and power allocation based on multi-agent deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13124–13128, 2020.

[31] N. Zhao, Y. C. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, "Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5141–5152, 2019.

[32] M. Fan, P. Chen, L. Wu, and J. Cheng, "Power allocation in multi-user cellular networks: deep reinforcement learning approaches," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6255–6267, 2021.

[33] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, and Y. Tassa, "Continuous control with deep reinforcement learning," *Computer Science*, vol. 8, pp. 1–14, 2015.

[34] J. N. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, pp. 2974–2983, 2018.

[35] A. Tampuu, T. Matiisen, D. Kodelja et al., "Multiagent cooperation and competition with deep reinforcement learning," *PLoS One*, vol. 12, article e0172395, 2017.

[36] P. Sunehag, G. Lever, A. Gruslys et al., "Value-decomposition networks for cooperative multi-agent learning," 2017, https://arxiv.org/abs/1706.05296.

[37] V. Mnih, K. Kavuhcuoglu, D. Silver et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–540, 2015.

[38] S. Fujimoto, H. Hoof, and M. Meger, "Addressing function approximation error in actor-critic methods," in *Proceedings of the 35th International Conference on Machine Learning*, pp. 1587–1596, Stockholm, 2018.

[39] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," 2014, https://arxiv.org/abs/1412.6980.