remote sensing

MDPI

Article

Cooperative Jamming Resource Allocation with Joint Multi-Domain Information Using Evolutionary Reinforcement Learning

Qi Xin [1], Zengxian Xin [2] and Tao Chen [1,*]

[1] College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China; xinqi@hrbeu.edu.cn
[2] Shanghai Radio Equipment Research Institute, Shanghai 201109, China; zengxian_xin@126.com
* Correspondence: chentao@hrbeu.edu.cn

Abstract: Addressing the formidable challenges posed by multiple jammers jamming multiple radars, which arise from spatial discretization, many degrees of freedom, numerous model input parameters, and the complexity of constraints, along with a multi-peaked objective function, this paper proposes a cooperative jamming resource allocation method, based on evolutionary reinforcement learning, that uses joint multi-domain information. Firstly, an adversarial scenario model is established, characterizing the interaction between multiple jammers and radars based on a multi-beam jammer model and a radar detection model. Subsequently, considering real-world scenarios, this paper analyzes the constraints and objective function involved in cooperative jamming resource allocation by multiple jammers. Finally, accounting for the impact of spatial, frequency, and energy domain information on jamming resource allocation, matrices representing spatial condition constraints, jamming beam allocation, and jamming power allocation are formulated to characterize the cooperative jamming resource allocation problem. Based on this foundation, the joint allocation of the jamming beam and jamming power is optimized under the constraints of jamming resources. Through simulation experiments, it was determined that, compared to the dung beetle optimizer (DBO) algorithm and the particle swarm optimization (PSO) algorithm, the proposed evolutionary reinforcement learning algorithm based on DBO and Q-Learning (DBO-QL) offers 3.03% and 6.25% improvements in terms of jamming benefit and 26.33% and 50.26% improvements in terms of optimization success rate, respectively. In terms of algorithm response time, the proposed hybrid DBO-QL algorithm has a response time of 0.11 s, which is 97.35% and 96.57% lower than the response times of the DBO and PSO algorithms, respectively. The results show that the method proposed in this paper has good convergence, stability, and timeliness.

Keywords: electronic countermeasures; jamming resource allocation; cooperative jamming; reinforcement learning; dung beetle optimizer algorithm

Citation: Xin, Q.; Xin, Z.; Chen, T. Cooperative Jamming Resource Allocation with Joint Multi-Domain Information Using Evolutionary Reinforcement Learning. *Remote Sens.* **2024**, *16*, 1955. https://doi.org/10.3390/rs16111955

Academic Editor: Piotr Samczynski

Received: 12 March 2024
Revised: 18 May 2024
Accepted: 23 May 2024
Published: 29 May 2024

1. Introduction

With the advancements in electronic warfare and technologies such as artificial intelligence (AI), radar and jamming systems endowed with cognitive capabilities have made significant progress [1–6]. These emerging technologies have transformed the traditional paradigms of electronic warfare engagement, shifting away from the conventional "one-to-one", "one-to-many", or "many-to-one" configurations to the more complex "many-to-many" engagement [7–9]. Consequently, conventional electronic warfare methods have proven incapable of adapting to the evolving dynamics of modern battlefields. The concept of cognitive electronic warfare has thus been introduced and is gradually emerging as a new developmental trend, poised to play a crucial role in future warfare scenarios [10–12]. In the context of "many-to-many" scenarios, the jamming entities now contend with multiple radars or networked radar systems as opposed to a single radar. Taking a networked radar

system as an example, it possesses the ability to gather, integrate, and leverage the resources and information advantages of various radars, rendering the traditional approach of employing a single jamming system less effective. Consequently, the cooperative jamming technique, involving multiple jammers and characterized by its broad jamming range, low hardware requirements, high system flexibility, and coordinated control of jamming beams, has become a focal point of current research [13–15].

As the complexity of real battlefield environments continues to increase, the efficient allocation of jamming resources, enabling jammers to maximize their jamming effectiveness with limited resources, has become a central challenge in the field of electronic warfare and related domains. Jamming resource allocation represents one manifestation of resource allocation in the domain of electronic warfare, and various resource allocation methods across different domains offer opportunities for mutual cross-fertilization. In recent years, research on communication resource allocation in the context of cognitive communication has yielded rich results, encompassing aspects such as time [16], space [17], spectra [18–20], and energy [20–22]. These achievements provide valuable references for the investigation of radar-jamming resource allocation.

Numerous scholars have conducted research on the problem of cooperative jamming resource allocation for multiple jammers. Zou [23], Wu [24], Jiang [25], and others have made improvements to the PSO algorithm to enhance its optimization probability and stability when dealing with cooperative jamming resource allocation. Lu et al. [26] employed detection and targeting probability to characterize the jamming effectiveness of a networked radar system in search and tracking modes, constructed a dual-factor jamming effectiveness assessment function, established a non-convex optimization model for cooperative jamming in a networked radar system, and utilized the dynamic adaptive discrete cuckoo search algorithm (DADCS), featuring improved path update strategies and the introduction of a global learning mechanism, to solve the model. This approach achieved the cooperative jamming resource allocation for aircraft formation in a networked radar system. Xing et al. [27] evaluated jamming suppression effects using a constant false-alarm rate detection model for networked radar, established a cooperative control model for multiple jammers involving jamming beams and transmission power levels, and utilized an enhanced improved artificial bee colony (IABC) algorithm to derive optimized allocation solutions for different operational scenarios. Yao et al. [28] developed a jamming resource allocation model considering factors such as jamming beams and jamming power, and solved the model using an improved genetic algorithm (GA), allowing for jamming resource allocation in networked radar systems with any number of nodes under conditions of limited jamming resources. Xing et al. [29] quantified target threat levels through fuzzy comprehensive evaluation, constructed a jamming resource allocation model based on the jamming efficiency matrix, and proposed an improved firefly algorithm (FA) to solve the model, ensuring both the rationality of task assignments and the stability of the algorithm. While these research achievements provide insights into addressing the problem of cooperative jamming resource allocation for multiple jammers, most of the established models are relatively simplistic. Additionally, the three dimensions of the spatial, frequency, and energy domains have not been comprehensively considered together. Moreover, as the scale of jamming resource allocation expands, existing algorithms are becoming increasingly prone to problems such as dimension explosion in solving, decreased convergence speed, reduced optimization probability, and longer algorithm response times. Consequently, obtaining an optimal solution becomes challenging and may not meet practical application requirements.

In recent years, reinforcement learning (RL) technology has been developed and deepened, with its ability to learn strategies that maximize rewards through the interaction of the agent with the environment [30] providing an effective solution for perceptual decision-making problems in complex systems. Nowadays, RL is widely applied in various prominent fields such as autonomous driving, game AI, and robot control. Since 2014, the Google DeepMind team has been applying RL technology to Atari games, and this

trained game AI can surpass the highest levels achieved by human players [31]. RL has also been applied in natural language processing, significantly enhancing its capabilities in semantic association, logical reasoning, and sentence generation [32]. Therefore, applying RL technology to the problem of jamming resource allocation is a direction worth exploring. Li et al. [33] modeled the breach process using a Markov decision process, and employed proximal policy optimization algorithms to learn the optimal sequential actions composed of jamming power allocation matrices and waypoints. The reward function defined depends on the distance to the target and the success or failure of the breach. The simulation experiments showed that this model could jointly learn the optimal path and jamming power allocation strategy, effectively completing the breach. Yue et al. [34] decoupled the coordination decision-making problem of a heterogeneous drone swarm for a suppression of enemy air defense mission divided into two sub-problems and solved them using a hierarchical multi-agent reinforcement learning approach. However, the current application of reinforcement learning in cooperative-jamming resource allocation for multiple jammers still faces challenges such as modeling difficulties and complex reward function design, resulting in poor adaptability to complex adversarial scenarios.

The development of evolutionary reinforcement learning (ERL) [35], which combines traditional evolutionary algorithms with reinforcement learning algorithms in order to enhance the speed and effectiveness of convergence, has provided a new perspective for solving the resource allocation problem. Xue et al. [36] proposed a hybrid algorithm combining deep Q networks (DQN) and GA for a resource-scheduling problem involving parallelism and subtask dependency. This algorithm generates the initial population of GA using DQN, improving its own convergence speed and optimization effect. Asghari et al. [37] used a coral reef algorithm to allocate resources to tasks and then used reinforcement learning to avoid falling into local optimums, by using this hybrid algorithm to improve resource allocation efficiency. Zhang et al. [38] addressed the problem of cluster-coordinated jamming decisions, improved the Q-learning algorithm by the ant colony algorithm, and a radar-jamming decision model was constructed, wherein each jammer in the cluster was mapped as an intelligent ant seeking the optimal path. Multiple jammers interacted to exchange information. This method enhanced the convergence speed and stability of the algorithm while reducing the hardware and power resource requirements for jammers.

Based on the above research findings, this paper further investigates the problem of cooperative jamming resource allocation based on multi-domain information. Combined with the idea of evolutionary reinforcement learning, this paper proposes a two-layer model for cooperative jamming resource allocation that integrally considers information in three dimensions: spatial, frequency, and energy domains. The jamming beam allocation matrix and jamming power allocation matrix were optimized using the outer DBO algorithm and the inner Q-learning algorithm, respectively. This method transforms the complex two-dimensional decision problem of the joint optimization of jamming beam allocation and jamming power allocation into two one-dimensional decision problems. The smaller dimensions of the solution space effectively prevent the algorithm from becoming trapped in a locally optimal solution and, at the same time, reduce the response time of the algorithm and ensure its stability.

In summary, the main contributions of this paper can be outlined as follows:

1. A model containing a spatial condition constraint matrix, a jamming beam allocation matrix, and a jamming power allocation matrix was constructed to address the cooperative-jamming resource allocation problem. The model integrates information from the spatial, frequency, and energy domains to formulate constraints and an objective function, making its results more aligned with the complex real-world environment.
2. In order to better solve the constructed model, an evolutionary reinforcement learning method called the hybrid DBO-QL algorithm was proposed. This method adopts a hierarchical selection and joint optimization strategy for jamming beam and power allocation, greatly reducing the response time of the algorithm while providing good convergence and stability.

The rest of this paper is organized as follows. Section 2 introduces the adversarial scenario model and formulates the constraints and objective function for cooperative jamming resource allocation among multiple jammers. Section 3 provides a detailed explanation of the DBO algorithm, Q-learning algorithm, and the proposed hybrid DBO-QL algorithm. Section 4 presents the simulation experiments and results' analysis for the three algorithms. Finally, the conclusions drawn from this study are presented in Section 5.

2. System Model

Section 2 describes the modelling of an adversarial scenario for cooperative jamming resource allocation and the design of the constraints and an objective function based on the model.

2.1. Adversarial Scenario Model

This paper considers a "many-to-many" adversarial scenario model, as illustrated in Figure 1. In the spatial adversarial scenario, multiple multi-beam jamming aircraft (jammers) collaborate to perform coordinated jamming tasks against multiple ground radars. The multi-beam jamming systems of each aircraft can simultaneously generate multiple jamming beams to jam multiple radars in different directions. The resources, such as the pointing direction, quantity, and transmission power of the jamming beams, can be flexibly controlled. The radar side detects targets based on the received signal-to-jamming ratio (SJR). Due to the limited jamming power that each jamming aircraft can provide, it is necessary to reasonably allocate the attitude of each jamming aircraft and the transmission power of different beams based on real-time battlefield situation information to improve the utilization of jamming resources. This allocation method is designed to achieve the highest possible jamming benefit using limited jamming resources.

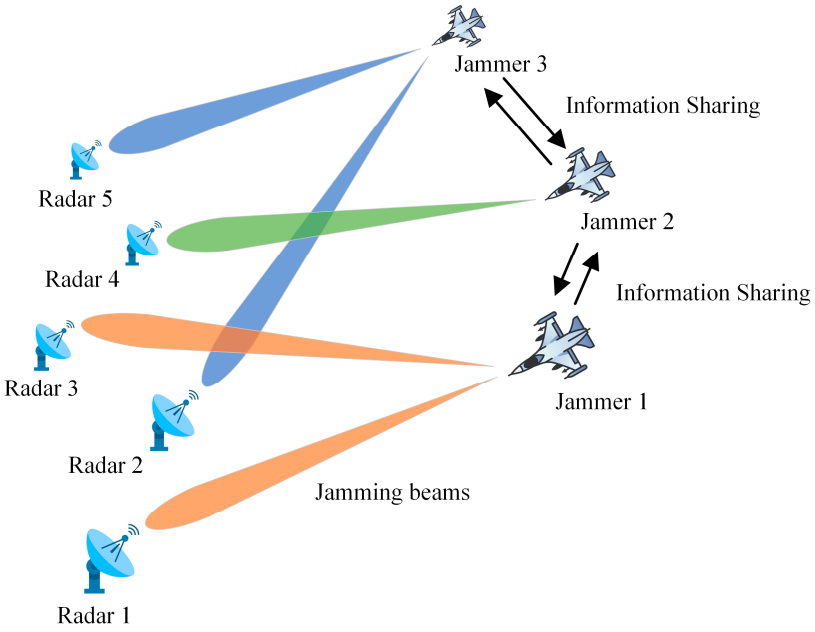


Figure 1. Schematic diagram of the adversarial scenario.

2.2. Constraints and Objective Function for Cooperative Jamming Resource Allocation Using Multiple Jammers

From a mathematical perspective, the problem of cooperative jamming resource allocation using multiple jammers can be formulated as a multi-choice problem under multi-dimensional constraints. The objective is to minimize the detection performance of enemy radars under the constraint of limited jamming resources. In light of this, the present study integrates information from the spatial, frequency, and energy domains to

consider and formulate the constraints and objective function for the cooperative-jamming resource allocation problem.

Assume there are M jammers and N radars in the adversarial scenario. To maximize jamming benefit, it is necessary to optimize the two jamming resources of the cooperative jamming system, namely, the jamming beam directions from jammers to radars and the transmission power of different jamming beams. For this purpose, a binary variable matrix K was defined to characterize the allocation relationship of jamming beam directions from jammers to radars, as shown in Equation (1).

$$K = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1N} \\ k_{21} & k_{22} & \cdots & k_{2N} \\ \vdots & \vdots & k_{mn} & \vdots \\ k_{M1} & k_{M2} & \cdots & k_{MN} \end{bmatrix} \tag{1}$$

Here, k_{mn} is a binary variable that can only assume values of '0' or '1'. $k_{mn} = 1$ denotes that jammer m allocates jamming beams to radar n, while $k_{mn} = 0$ indicates that jammer m does not allocate jamming beams to radar n.

In terms of the allocation of transmission power for different jamming beams, a jamming power allocation matrix P was defined to quantify the distribution of power resources in the cooperative jamming system. The corresponding formulation is provided in Equation (2).

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1N} \\ p_{21} & p_{22} & \cdots & p_{2N} \\ \vdots & \vdots & p_{mn} & \vdots \\ p_{M1} & p_{M2} & \cdots & p_{MN} \end{bmatrix} \tag{2}$$

Here, p_{mn} represents the transmission power of the jamming beam assigned by jammer m to radar n.

Therefore, the objective of the cooperative-jamming resource allocation problem for multiple jammers is to solve for the optimal jamming beam allocation matrix K and jamming power allocation matrix P under multiple constraints. In this regard, the aim is to achieve optimal jamming performance in situations where system-jamming resources are limited. Furthermore, considering the constraints in cooperative-jamming resource allocation for multiple jammers, this paper designed an objective function that accurately quantifies the jamming benefit obtained from cooperative-jamming resource allocation. By solving the optimization problem, the best configuration for the system's jamming resource variables is sought to maximize the jamming benefit characterized by the objective function.

2.2.1. Constraints

Setting the constraints for jamming resource allocation adequately and reasonably is beneficial for quickly finding the optimal jamming resource allocation strategy based on the actual battlefield situation, thereby enhancing the efficiency of jammer utilization. In this regard, this paper considers constraints for the system model based on the following five aspects:

1. Spatial condition constraint. When allocating jamming resources, considering the practical situation, not all jammers can be assigned to jam a specific target radar. An essential prerequisite for such an assignment is that the jammer must be within the beam coverage area of that radar. Therefore, this paper defined a binary variable matrix Q to characterize the spatial relationship between jammers and radars, as shown in Equation (3).

$$Q = \begin{bmatrix} q_{11} & q_{12} & \cdots & q_{1N} \\ q_{21} & q_{22} & \cdots & q_{2N} \\ \vdots & \vdots & q_{mn} & \vdots \\ q_{M1} & q_{M2} & \cdots & q_{MN} \end{bmatrix}, \tag{3}$$

where $q_{mn} = 1$ indicates that jammer m is within the beam coverage area of radar n and can be assigned to jam that radar, while $q_{mn} = 0$ indicates that jammer m is not within the beam coverage area of radar n and cannot be assigned to jam that radar.

2. Jamming beam allocation quantity constraint. Constrained by the payload capacity of the jammer itself, it is assumed that each jammer can simultaneously allocate a maximum of l jamming beams to jam multiple radars, as follows:

$$\sum_{n=1}^{N} k_{mn} \leq l, m = 1, 2, \cdots, M \tag{4}$$

3. Jamming resource utilization constraint. A single jamming beam emitted by a multi-beam jammer can effectively jam one radar. In practice, the number of radars to be jammed may be several times the number of jammers. To enhance the efficiency of jamming resources and avoid wastage, it is stipulated that each radar can be assigned a maximum of one jamming beam, as follows:

$$\sum_{m=1}^{M} k_{mn} \leq 1, n = 1, 2, \cdots, N \tag{5}$$

4. Jamming power allocation constraint. Each targeted radar must be allocated jamming power. Assume that the SJR at the radar end is greater than 14 dB indicates that the jamming provided by the jammers is completely ineffective for the radar in question. An SJR less than 3 dB indicates that the radar has been effectively jammed, and further jamming power allocation beyond this range would result in a waste of jamming resources, as follows:

$$\begin{cases} p_{mn\_14\text{dB}} \leq p_{mn} \leq p_{mn\_3\text{dB}}, \textit{if } k_{mn} = 1 \\ p_{mn} = 0, \textit{else} \end{cases} \tag{6}$$

5. Total jamming power constraint. Constrained by the payload capacity of the jammer itself, the sum of the jamming power allocated to all jamming beams of a multi-beam jammer should not exceed the maximum jamming power it can provide, as follows:

$$\sum_{n=1}^{N} p_{mn} \leq p_{m\_\max}, m = 1, 2, \cdots, M \tag{7}$$

2.2.2. Objective Function

The purpose of multi-jammer cooperative-jamming resource allocation is to minimize the detection probability of a radar system by optimizing the allocation of the jamming beam and power resources of the multi-jammer cooperative-jamming system while satisfying the working frequency matching between the jammers and the radars. Therefore, the objective function designed in this paper consists of the following four evaluation factors:

1. Spectral alignment.

The spectral alignment factor for jammer m with radar n, denoted as r_{mn} ($0 \leq r_{mn} \leq 1$), is used to describe the alignment of jamming frequency with radar operating frequency. A spectral targeting benefit factor, J_f, was defined to assess the overall spectral targeting degree of the current jamming resource allocation scheme and its impact on jamming

effectiveness. A larger J_f indicates a higher overall spectral targeting degree in the current jamming resource allocation scheme, leading to better jamming effectiveness.

Assuming the jamming frequency range for jammer m is $[f_{m1}, f_{m2}]$ and the operating frequency range for radar n is $[f_{n1}, f_{n2}]$, the spectral overlap between the jammer's jamming frequency and the radar's operating frequency in the frequency domain is illustrated in Figure 2.

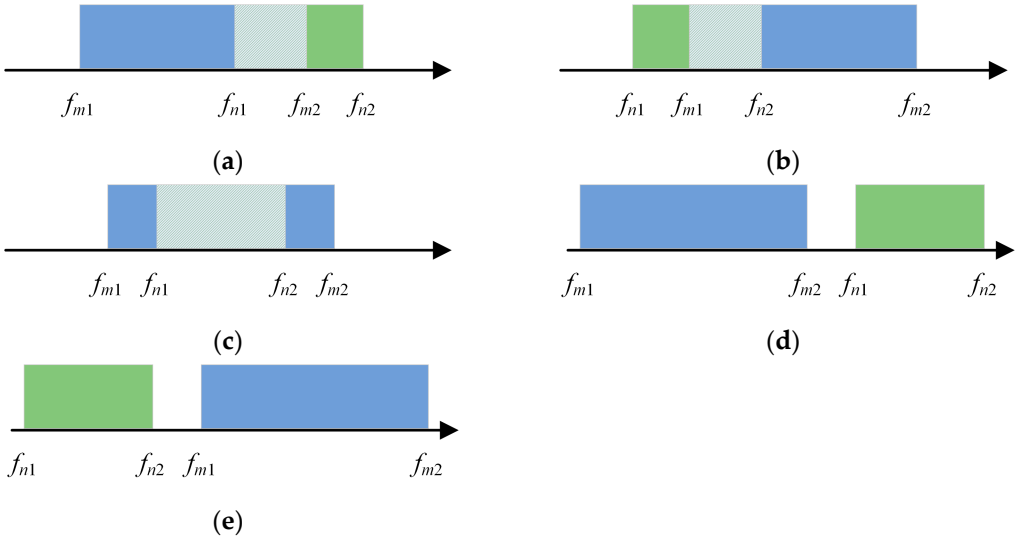


Figure 2. Spectral overlap between the jammer and radar in the frequency domain: (**a**) $f_{m1} < f_{n1}$, $f_{n1} < f_{m2} < f_{n2}$; (**b**) $f_{n1} < f_{m1} < f_{n2}$, $f_{m2} > f_{n2}$; (**c**) $f_{m1} < f_{n1}$, $f_{m2} > f_{n2}$; (**d**) $f_{m2} < f_{n1}$; (**e**) $f_{m1} > f_{n2}$.

From Figure 2, it can be observed that when there are three scenarios (a), (b), and (c) depicting the spectral overlap between the jammer's jamming frequency and the radar's operating frequency, the overlapping region in the figure can be represented using Equation (8).

$$\Delta f_{mn} = \min(f_{m2}, f_{n2}) - \max(f_{m1}, f_{n1}) \tag{8}$$

When the jammer's jamming frequency coincides with the radar's operating frequency in scenarios (d) and (e), it can be observed that there is no overlapping region in the figure. In this case, the spectral overlap degree should be 0. However, if Equation (8) is used for calculation, Δf_{mn} will be a negative number, which is inaccurate. Therefore, this paper introduced the function $\mathrm{sgn}(\cdot)$, whose value is defined as follows:

$$\mathrm{sgn}(x) = \begin{cases} 1, if\ x > 0 \\ 0, else \end{cases} \tag{9}$$

Therefore, a formula satisfying all the situations depicted in Figure 2 for the spectral overlap degree r_{mn} can be defined using Equation (10).

$$r_{mn} = \frac{\Delta f_{mn}}{B_n} \cdot \mathrm{sgn}(\Delta f_{mn}), \tag{10}$$

where B_n represents the operating bandwidth of the radar. Furthermore, the frequency-domain targeting benefit factor J_f is expressed by Equation (11).

$$J_f = \sum_{m=1}^{M} \sum_{n=1}^{N} \alpha_1 k_{mn} r_{mn}, \tag{11}$$

where α_1 is a normalization factor, and r_{mn} represents the frequency-domain overlap between jammer m and radar n.

2. Signal-to-jamming ratio at the radar receiver.

The detection probability on the radar side is closely associated with the SJR at its receiving end. The objective of cooperative-jamming resource allocation among multiple jammers is to appropriately determine the jamming power for jammers, thereby reducing the SJR at the radar end. Consequently, the SJR at the radar end serves as an evaluative metric for the jamming effectiveness at a given jamming power.

In the system model established in this paper, the jammers jam the radars to conceal themselves from detection by said radars. Therefore, the distance between the radar and the target is assumed to be equal to the distance between the radar and the jammer. Consequently, based on the radar equation, the SJR obtained by the radar can be expressed as follows:

$$SJR_{radar} = \frac{\sigma P_r G_r}{4\pi {R_d}^2 P_j G_j \mu}, \tag{12}$$

where σ represents the target's radar cross section area; P_{r} and P_{j} denote the transmission power of the radar signal and jamming signal, respectively; R_d is the distance between the radar and the jammer; G_r and G_{j} are the power gains of the radar and jammer antennas, respectively; and μ is the polarization matching loss coefficient between the jamming signal and the radar signal.

Similarly, the SJR obtained by the jammer can be derived as follows:

$$SJR_{jammer} = \frac{P_r G_r G_j \lambda^2 \mu}{(4\pi)^2 {R_d}^2 P_j}, \tag{13}$$

where λ represents the wavelength of the transmitted signal.

The ratio between the SJR obtained by the radar and that obtained by the jammer can be obtained using the following formula:

$$\frac{SJR_{radar}}{SJR_{jammer}} = \frac{\sigma P_r G_r}{4\pi {R_d}^2 P_j G_j \mu} \cdot \frac{(4\pi)^2 {R_d}^2 P_j}{P_r G_r G_j \lambda^2 \mu} = \frac{4\pi\sigma}{G_j^2 \lambda^2 \mu^2} \tag{14}$$

Hence, the jammer can estimate the SJR at the radar receiver based on the SJR obtained by its own platforms. To assess the jamming effect of jammer m transmitting a jamming signal with power p_{mn} on radar n, the variable η_{mn} was defined to measure the effectiveness of the jamming, as expressed in Equation (15).

$$\eta_{mn} = \log_2\left(\frac{\varepsilon}{SJR_{radar}} + 1\right), \tag{15}$$

where ε is a scaling factor, and since this paper conventionally sets the minimum SJR to 3 dB, during the simulation experiments, ε was set to 3.

A radar receiver SJR benefit-factor, denoted as J_p, was defined to evaluate the jamming effectiveness of the current jamming resource allocation scheme, as follows:

$$J_p = \sum_{m=1}^{M} \sum_{n=1}^{N} \alpha_2 k_{mn} \eta_{mn}, \tag{16}$$

where α_2 is a normalization factor and η_{mn} represents the jamming effectiveness of jammer m with respect to radar n.

3. Distance between jammers and radars.

According to Equation (12), under the condition of maintaining a constant SJR at the radar end, the jamming power required to jam a radar is inversely proportional to the square of the distance between the radar and the jammer. In other words, the greater the distance between the jammer and the radar being jammed, the less jamming power

required to jam the radar. This enables greater jamming effectiveness with reduced jamming power consumption.

Assuming the position of jammer m is denoted as (x_m, y_m, z_m) and the position of radar n is denoted as (x_n, y_n, z_n), the distance d_{mn} between them can be expressed as follows:

$$d_{mn} = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2 + (z_m - z_n)^2} \tag{17}$$

An overall distance benefit factor J_d was defined to evaluate the jamming effectiveness of the current jamming resource allocation scheme, as follows:

$$J_d = \sum_{m=1}^{M} \sum_{n=1}^{N} \alpha_3 k_{mn} d_{mn}, \tag{18}$$

where α_3 is a normalization factor and d_{mn} represents the distance between jammer m and radar n.

4. Number of jammed radars.

In practical scenarios, the number of radars that can be jammed may be constrained by factors such as the quantity of jammers, payload limitations, and constraints in the spatial, frequency, and energy domains. Consequently, not all detected radars can be jammed by the jammers. The objective of cooperative-jamming resource allocation involving multiple jammers is to maximize the jamming coverage over the detected radars. To assess the jamming effectiveness of the current resource allocation scheme, a benefit factor for the number of jammed radars was defined as follows:

$$J_n = \sum_{m=1}^{M} \sum_{n=1}^{N} \alpha_4 k_{mn}, \tag{19}$$

where α_4 is a normalization factor.

Employing a linear weighting method to balance the weights of the four evaluation factors mentioned above, these factors can be integrated into the following unified objective function:

$$\max J = \max[J_f, J_p, J_d, J_n] = \max(\beta_1 J_f + \beta_2 J_p + \beta_3 J_d + \beta_4 J_n), \tag{20}$$

where β_1, β_2, β_3, and β_4 are weighting factors and $\beta_1 + \beta_2 + \beta_3 + \beta_4 = 1$.

J represents the jamming benefit, and a larger value indicates more effective jamming of the entire cooperative jamming system with respect to the radars. The magnitude of J can be used to gauge the rationality of jamming resource allocation, thereby enhancing the utilization efficiency of jamming resources.

3. Evolutionary Reinforcement Learning Method Based on the DBO and Q-Learning Algorithms

An evolutionary reinforcement learning method called the hybrid DBO-QL algorithm was developed for the cooperative-jamming resource allocation model involving the complex constraints constructed in Section 2. The principle of the proposed algorithm is elaborated in detail below.

3.1. DBO Algorithm

The DBO algorithm, introduced by Xue et al. [39] in 2022, is a novel metaheuristic swarm intelligence optimization algorithm and a type of evolutionary algorithm. It draws inspiration from the rolling, dancing, foraging, stealing, and reproductive behaviors of dung beetles. This algorithm simultaneously considers global and local exploitation, endowing it with fast convergence and high accuracy. It can effectively address complex optimization problems.

In the DBO algorithm, each dung beetle's position corresponds to a solution. There are five behaviors exhibited by the dung beetles when foraging in this algorithm: rolling a ball, utilizing celestial cues such as the sun for navigation, thus allowing the ball to be rolled in a straight line; dancing, which allows a dung beetle to reposition itself; reproduction, a natural behavior in which dung beetles roll fecal balls to a secure location, where they hide and use the balls as a breeding ground; foraging, in which some adult dung beetles emerge from the ground to search for food; and stealing, in which certain dung beetles, known as thieves, steal fecal balls from other beetles.

Therefore, the dung beetle population in the algorithm is divided into four categories, namely, the ball-rolling dung beetle, the brood ball, the small dung beetle, and the thief, as shown in Figure 3. The population is divided into the different roles in a ratio of 6:6:7:11. In other words, according to the population size in Figure 3, out of 30 individuals, six dung beetles are assigned to engage in ball-rolling behavior. These ball-rolling dung beetles adjust their running direction based on various natural environmental influences, initially searching for a safe location for foraging. Another six dung beetles are designated as beetles engaging in reproductive behavior, and the reproduction balls will be placed in a known safe area. Seven dung beetles are defined as small dung beetles, which forage in the optimal foraging area. The remaining eleven dung beetles are classified as thieves, and thief dung beetles search for food based on the positions of other dung beetles and the optimal foraging area.

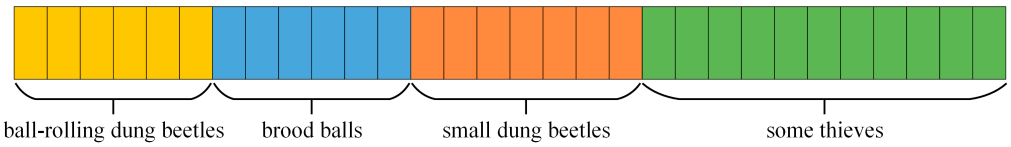


Figure 3. The partitioning rules for the population.

3.1.1. Dung Beetle Ball Rolling

The conditions for ball rolling carried out by dung beetles can be categorized into two scenarios: obstacle-free conditions and conditions involving obstacles.

1. The scenario without obstacles

When there are no obstacles along the path of the dung beetle's progression, the ball-rolling dung beetle employs the sun as a navigation reference to ensure its dung-ball rolls along a straight trajectory. Natural factors can affect the beetle's path during ball rolling. The position-updating mechanism for the dung beetle during the ball-rolling process is represented by Equation (21) as follows:

$$x_i^{t+1} = x_i^t + \alpha_{DBO} \cdot k_{DBO} \cdot x_i^{t-1} + b \cdot \left| x_i^t - x_{worst}^t \right|, \tag{21}$$

where t denotes the current iteration number; x_i^t represents the position of the ith dung beetle in the population in the tth iteration; $k_{DBO} \in (0, 0.2]$ denotes the deflection coefficient, a constant value; $b \in (0, 1)$ represents a constant value; α_{DBO} is a natural coefficient indicating whether there is a deviation from the original direction, assigned a value of 1 or -1 based on a probability method, with 1 indicating no deviation and -1 indicating a deviation from the original direction; x_{worst}^t represents the worst position in the current population; and $\left| x_i^t - x_{worst}^t \right|$ is utilized to simulate changes in light intensity.

2. The scenario with obstacles

When dung beetles encounter obstacles that hinder their progression, they need to adjust their direction through a dancing mechanism. The formula defining the position update during a ball-rolling dung beetle's dance is as follows:

$$x_i^{t+1} = x_i^t + \tan(\theta) \cdot \left| x_i^t - x_i^{t-1} \right|, \tag{22}$$

where $\theta \in [0, \pi]$ represents the deflection angle. When θ equals 0, $\frac{\pi}{2}$, or π, $\tan(\theta)$ is either 0 or meaningless. Therefore, it is stipulated that when θ equals 0, $\frac{\pi}{2}$, or π, the dung beetle's position will not be updated.

3.1.2. Dung Beetle Reproduction

In dung beetle reproduction, a boundary selection strategy is used to simulate the oviposition area for female dung beetles. The definition of the oviposition area is expressed in the following Equation (23):

$$\begin{cases} Lb^* = \max\left\{x_{gbest}^t \cdot (1-R), Lb\right\} \\ Ub^* = \min\left\{x_{gbest}^t \cdot (1+R), Ub\right\} \end{cases}, \tag{23}$$

where $R = 1 - t/T$, with T representing the maximum iteration count; Lb and Ub are the lower and upper bounds of the optimization problem, respectively; and x_{gbest}^t denotes the global optimal position of the current population.

The DBO algorithm defines the lower bound Lb^* and upper bound Ub^* of the oviposition area with each iteration. In other words, the region where dung beetles lay eggs changes dynamically as the iteration count progresses. Once the female dung beetle determines the oviposition area, it proceeds to lay eggs within that region. Each female dung beetle generates only one dung ball in each iteration. Since the oviposition area dynamically adjusts with the iteration count, the position of the dung ball is also dynamically adjusted during the iteration process, which is defined as follows:

$$B_i^{t+1} = x_{gbest}^t + b_1 \cdot (B_i^t - Lb^*) + b_2 \cdot (B_i^t - Ub^*), \tag{24}$$

where B_i^{t+1} represents the position of the ith dung ball in the tth iteration; b_1 and b_2 denote two independent random vectors with a size of $1 \times D$; and D represents the dimensionality of the optimization problem.

3.1.3. Dung Beetle Foraging

Guide dung beetle larvae to search for food and simulate their foraging behavior by establishing an optimal foraging area, which is defined in Equation (25).

$$\begin{cases} Lb^p = \max\left\{x_{pbest}^t \cdot (1-R), Lb\right\} \\ Ub^p = \min\left\{x_{pbest}^t \cdot (1+R), Ub\right\} \end{cases} \tag{25}$$

Here, R remains consistent with the previous definition and x_{pbest}^t represents the local optimal position of the current population.

The DBO algorithm defines the lower bound Lb^p and upper bound Ub^p for the foraging area of the small dung beetles. The position update process for the small dung beetles is expressed in Equation (26), as follows:

$$x_i^{t+1} = x_i^t + C_1 \cdot (x_i^t - Lb^p) + C_2 \cdot (x_i^t - Ub^p), \tag{26}$$

where C_1 is a random number following a normal distribution, $C_1 \sim N(0,1)$ and C_2 is a random vector with a size of $1 \times D$ belonging to $(0,1)$.

3.1.4. Dung Beetle Stealing

Within the dung beetle population, there will be some that steal dung balls from other dung beetles. The position update process for the thieving dung beetles is expressed in Equation (27), as follows:

$$x_i^{t+1} = x_{pbest}^t + S_{DBO} \cdot g \cdot \left(\left|x_i^t - x_{gbest}^t\right| + \left|x_i^t - x_{pbest}^t\right|\right), \tag{27}$$

where g represents a random vector of size $1 \times D$ following a normal distribution and S_{DBO} denotes a constant value.

A flowchart of the DBO algorithm is shown in Figure 4, which primarily consists of the following six steps:

1. Initialize the dung beetle populations and set the parameters of the DBO algorithm;
2. Calculate the fitness values for all dung beetle positions based on the objective function;
3. Update the positions of all dung beetle populations according to the set rule;
4. Check whether each updated dung beetle has exceeded the boundaries;
5. Update the current optimal solution and its fitness value;
6. Repeat the above steps, and after the iteration count t reaches the maximum iteration count, output the global optimal value and its corresponding solution.

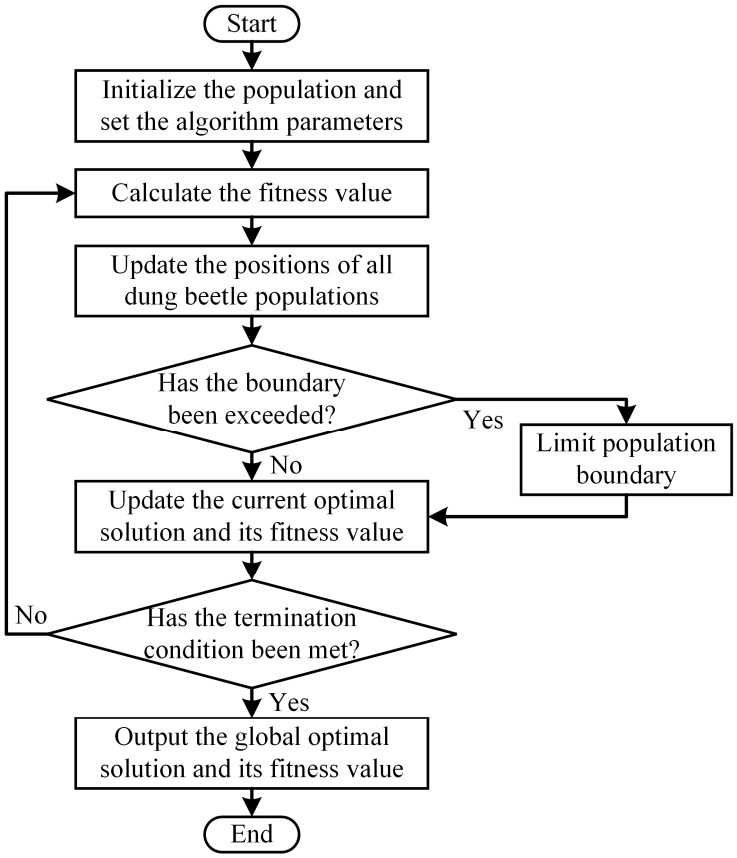


Figure 4. DBO algorithm flow chart.

3.2. Q-Learning Algorithm

As a value-function-based algorithm, Q-learning is a typical temporal difference (TD) algorithm [40] used within the realm of reinforcement learning. This algorithm engages with the environment through a trial-and-error approach, adjusting the next action based on environmental feedback. Through multiple interactions, Q-learning can effectively solve optimal decision-making problems under model-free conditions.

The Q-learning algorithm updates the value function based on the immediate reward obtained from the next state and the estimated value of the value function. Thus, at time $k+1$, the update of the value function is expressed as follows:

$$Q_{k+1}^{\pi}(s_k, a_k) = Q_k^{\pi}(s_k, a_k) + \frac{1}{k+1}[r_{k+1} - Q_k^{\pi}(s_k, a_k)] \tag{28}$$

Hence, the update function can be expressed as follows:

$$Q(s_k, a_k) \leftarrow Q(s_k, a_k) + \alpha[r_k + \gamma \max_* Q(s_{k+1}, a_{k+1}) - Q(s_k, a_k)], \tag{29}$$

where $Q(s_k, a_k)$ is the action-value function at the current time step, while $Q(s_{k+1}, a_{k+1})$ is the action-value function at the next time step; α is the learning rate; γ is the discount factor, $\alpha, \gamma \in (0, 1)$; and r is the reward value.

When the agent selects an action, to avoid falling into local optima, the choice of strategy π typically involves a trade-off between "exploration and exploitation". The greedy algorithm ($\varepsilon - Greedy$) is a commonly used method for this purpose, where ε represents the exploration factor, $\varepsilon \in (0, 1)$. In this algorithm, exploitation is performed with a probability of $1 - \varepsilon$, and exploration is performed with a probability of ε. The specific expression is as follows:

$$\pi^*(s_k) = \begin{cases} rand(a_k), p = \varepsilon \\ \pi(s_k), p = 1 - \varepsilon \end{cases} \tag{30}$$

The optimization process of the Q-learning algorithm is an exploration–exploitation process. After reaching the maximum iteration count, a convergent state–action two-dimensional table, known as the Q-table, is obtained.

3.3. The Proposed Hybrid DBO-QL Algorithm

Since the "many-to-many" adversarial scenario model proposed in Section 2 has numerous input parameters, complex constraints, and a multi-peaked objective function, commonly used metaheuristic swarm intelligence algorithms for solving optimization problems often exhibit slow convergence and unsatisfactory convergence results. On the other hand, it is difficult to apply reinforcement learning algorithms with good convergence results and high efficiency in situations with multiple inputs and complex constraints. To address this, this paper proposes a combination of the DBO algorithm, which is an evolutionary algorithm, and the classical Q-Learning algorithm from reinforcement learning. Through this evolutionary reinforcement learning method, this paper aims to tackle the jamming resource allocation problem in a scenario where multiple jammers are jamming multiple radars.

Assuming the adversarial process between the jammers and radars involves a total of U adversarial rounds at a certain time, the process of cooperative jamming resource allocation with joint multi-domain information proceeds as follows.

In the uth adversarial round, the friendly side acquires information about jammers and radars through situational awareness and electromagnetic spectrum sensing in the external environment. Using radar–signal–deinterleaving technology, it determines the quantity, positions, and parameters of the radar radiation sources. Subsequently, based on the received information about the parameters of jammers and radar radiation sources, and with specific constraints in mind, the outer layer of the DBO algorithm is employed to assess which radars can be jammed by the jamming beams of the same jammer. This process generates the jamming beam allocation matrix K. Furthermore, the inner layer of the Q-learning algorithm, utilizing the jamming beam allocation matrix K as a foundation, allocates the transmission power for each jamming beam. This generates the jamming power allocation matrix P, completing the cooperative jamming resource allocation for multiple jammers in the adversarial round. The process then proceeds to the $(u + 1)$th adversarial round. The method for cooperative jamming resource allocation with joint multi-domain information is illustrated in Figure 5.

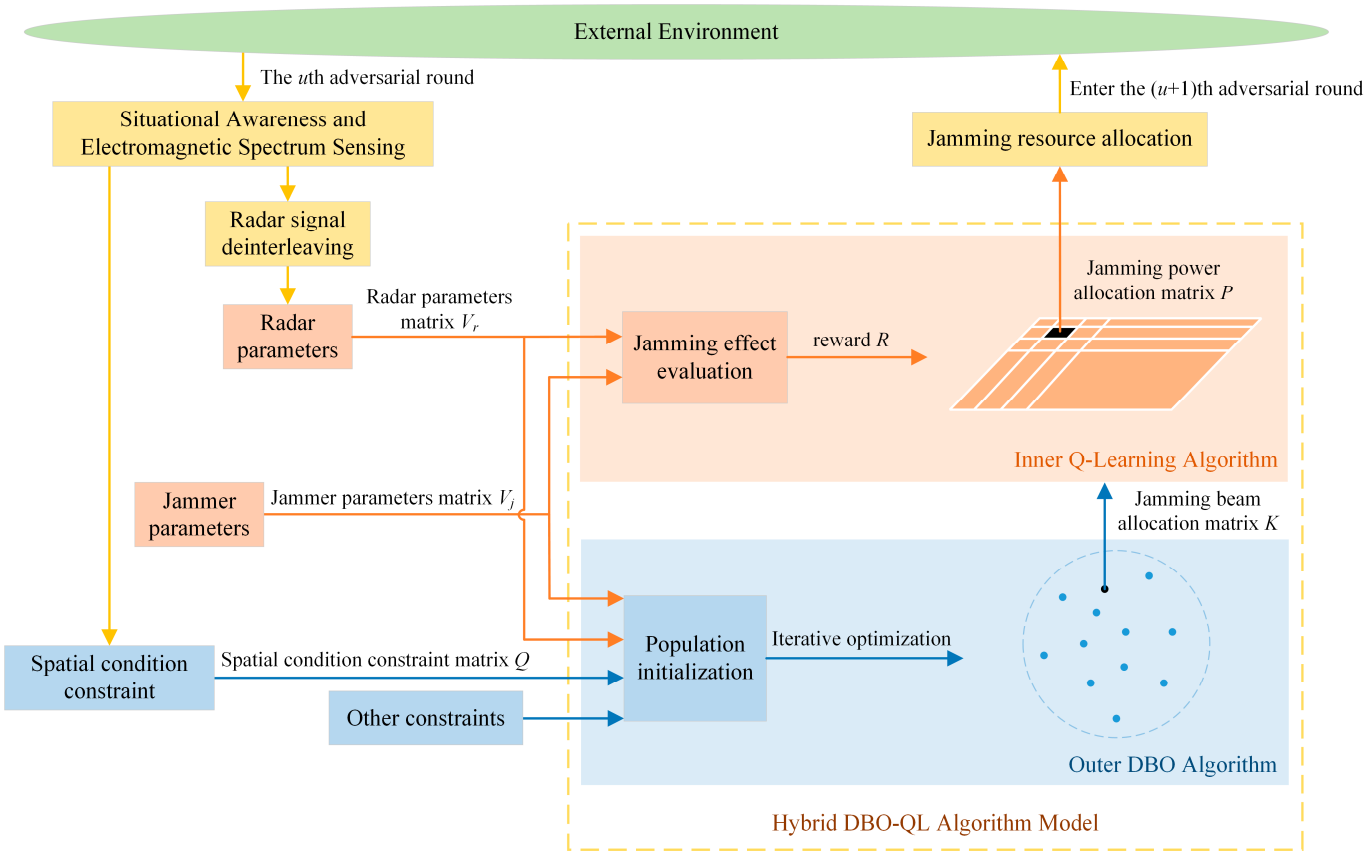


Figure 5. Cooperative jamming resource allocation with joint multi-domain information.

The objective of the outer-layer DBO algorithm is to find the optimal jamming beam allocation matrix K while satisfying constraints (3), (4), and (5). The objective function of the outer-layer DBO algorithm should incorporate Equations (11), (18), and (19). Therefore, the objective function of the outer-layer DBO algorithm is defined as follows:

$$J_O = \gamma_1 J_f + \gamma_2 J_d + \gamma_3 J_n, \tag{31}$$

where γ_1, γ_2, and γ_3 are all weighting factors and $\gamma_1 + \gamma_2 + \gamma_3 = 1$. These factors can be configured based on the specific emphasis of the criteria during utilization.

The inner-layer Q-learning algorithm's objective is to make decisions about the optimal jamming power allocation matrix P under the guidance of the optimal jamming beam allocation matrix K. This process is aimed at achieving cooperative jamming resource allocation with multiple jammers. Therefore, the design approach for the inner-layer Q-learning algorithm is as follows:

(1) The state set S consists of matrices representing jamming power allocations. The initial state is randomly selected from the set of jamming power allocation matrices. When the agent takes action in the current state, the transition to the next state transpires.

(2) The action set A consists of jamming power values. Adhering to the constraints outlined in Equation (6), the jamming power range $[p_{mn\_14\mathrm{dB}}, p_{mn\_3\mathrm{dB}}]$ is reasonably partitioned into h discrete values. The initial action is randomly selected from these h discrete values, and subsequent actions are chosen according to the policy π from Q-learning.

(3) The value of the reward, denoted by R^*, is determined based on the effectiveness of jamming η as defined in Equation (15). The specific definition is provided below:

$$R^*_{mn} = \begin{cases} \eta_{mn}, if \sum_{n=1}^{N} p_{mn} \le p_{m\_\max} \\ 0, else \end{cases} \tag{32}$$

Thus, the jamming effect of the jamming power allocation matrix P, determined by the decision of the inner-layer Q-learning algorithm, can be assessed using the radar receiver SJR benefit factor J_p defined by Equation (16), as follows:

$$J_I = J_p = \sum_{m=1}^{M} \sum_{n=1}^{N} \alpha_2 k_{mn} \eta_{mn} \tag{33}$$

Therefore, for the proposed hybrid DBO-QL algorithm in this paper, the jamming benefit, as defined by Equation (20), can be expressed as follows:

$$\max J = \max[J_O, J_I] = \max[\lambda_1 J_O + \lambda_2 J_I] = \max[\lambda_1(\gamma_1 J_f + \gamma_2 J_d + \gamma_3 J_n) + \lambda_2 J_p], \tag{34}$$

where λ_1 and λ_2 both represent weighting factors and $\lambda_1 + \lambda_2 = 1$.

The pseudo-code of the hybrid DBO-QL algorithm proposed in this paper is Algorithm 1.

Algorithm 1 Pseudo-Code of the hybrid DBO-QL algorithm

Input: Radar parameter matrix V_r; jamming parameter matrix V_j; spatial condition constraint matrix Q; weighting factors γ_1, γ_2, γ_3, λ_1, andλ_2; and the parameters for the DBO algorithm include the maximum number of iterations T_{DBO} and the population size N_{DBO}, while the Q-Learning algorithm involves the maximum number of iterations T_{QL}, initial learning rate α_0, initial exploration factor ε_0, and discount factor γ.
Output: Jamming beam allocation matrix K, jamming power allocation matrix P.

```
Initialize the population and parameters for the DBO algorithm.
while(t < T_DBO) do
  for i = 1 : N_DBO do
    if i == ball-rolling dung beetle then
      δ = rand(1);
      if δ < 0.9 then
        η_DBO = rand(1);
        if η_DBO > λ_DBO(set a probability value λ_DBO) then
          α_DBO = 1;
        else
          α_DBO = −1;
        end if
        Update the ball-rolling dung beetle's position using Equation (21);
      else
        Update the ball-rolling dung beetle's position using Equation (22);
      end if
    end if
    if i == brood ball then
      R = 1 − t/T_DBO
      for i = 1 : N_brood (the brood ball number) do
        Update the brood ball's position using Equation (24);
        for j = 1:D do
          if B_ij^t > Ub_j^* then
            B_ij^t = Ub_j^*
          end if
          if B_ij^t < Lb_j^* then
            B_ij^t = Lb_j^*
          end if
        end for
      end for
    end if
```

```
    if i == small dung beetle then
      Update the small dung beetle's position using Equation (26);
    end if
    if i == thief then
      Update the thief's position using Equation (27);
    end if
  end for
  if the newly generated position is better than before then
    Update it;
  end if
  t = t + 1;
end while
return K and its fitness value J_O
Initialize the Q(s, a) matrix and the parameters for the Q-learning algorithm.
for k = 1 : T_QL do
  Obtain the initial state s;
  Use the ε–greedy policy to select an action a in the current state s based on Q;
  Obtain the feedback r, s' from the environment;
  Update the Q(s, a) by using Equation (29);
  Update the next state s by using s';
end for
return P and its fitness value J_I
```

4. Simulation Experiments and Results Analysis

To highlight the effectiveness, superiority, and timeliness of the proposed hybrid DBO-QL algorithm with respect to the joint optimization problem of jamming beam and jamming power allocation, this paper conducted simulation experiments concerning a complex scenario involving "multiple jammers against multiple radars". Within the same simulation environment, the hybrid DBO-QL algorithm was compared with the DBO algorithm and the PSO algorithm in terms of jamming benefit, algorithm response time, and optimization success rate. To ensure accuracy and fairness in the comparison, all the operational parameters of the jammers and radars were maintained consistently.

4.1. Simulation Parameter Configuration

Assuming that at a certain moment, there are a total of $M = 4$ jammers conducting cooperative jamming tasks against $N = 10$ radars in the adversarial space, the radar parameter information and jammer parameter information sets during the simulation are presented in Tables 1 and 2, respectively.

Table 1. Radar parameter information.

Radar ID	Position/km	Center Frequency/GHz	Bandwidth/MHz	Pulse Repetition Frequency/Hz	Pulse Width/us	Power/kW
No.1	(83.2, 48.2, 0.1)	8.4	84.0	2616.0	5.0	154.3
No.2	(92.1, 1.2, 0.1)	8.8	80.0	1706.0	16.0	148.1
No.3	(75.9, 68.4, 0.0)	9.2	42.0	2337.0	8.0	162.0
No.4	(83.7, 24.9, 0.0)	6.7	100.0	844.0	45.0	159.0
No.5	(96.3, 14.2, 0.1)	6.6	90.0	1412.0	22.0	141.2
No.6	(79.2, 32.8, 0.1)	7.3	33.0	1920.0	15.0	165.5
No.7	(92.3, 98.5, 0.0)	8.2	62.0	855.0	43.0	162.4
No.8	(89.6, 61.8, 0.1)	9.4	50.0	2252.0	10.0	144.7
No.9	(73.8, 92.2, 0.1)	9.5	20.0	1262.0	24.0	142.6
No.10	(77.4, 84.9, 0.0)	6.4	92.0	1180.0	26.0	165.0

Table 2. Jammer parameter information.

Jammer ID	Position/km	Azimuth Angle/°	Elevation Angle/°	Jamming Frequency Range/GHz	Power/kW
No.1	(28.2, 31.1, 1.9)	0.5	45	(6.3, 8.2)	0.2
No.2	(3.6, 26.6, 2.2)	0	47.5	(6.6, 8.5)	0.2
No.3	(24.0, 77.5, 2.4)	0	42.5	(7.5, 9.4)	0.2
No.4	(10.0, 52.3, 1.8)	−0.5	45	(7.9, 9.7)	0.2

The spatial relationship between the jammers and radars is illustrated in Figure 6.

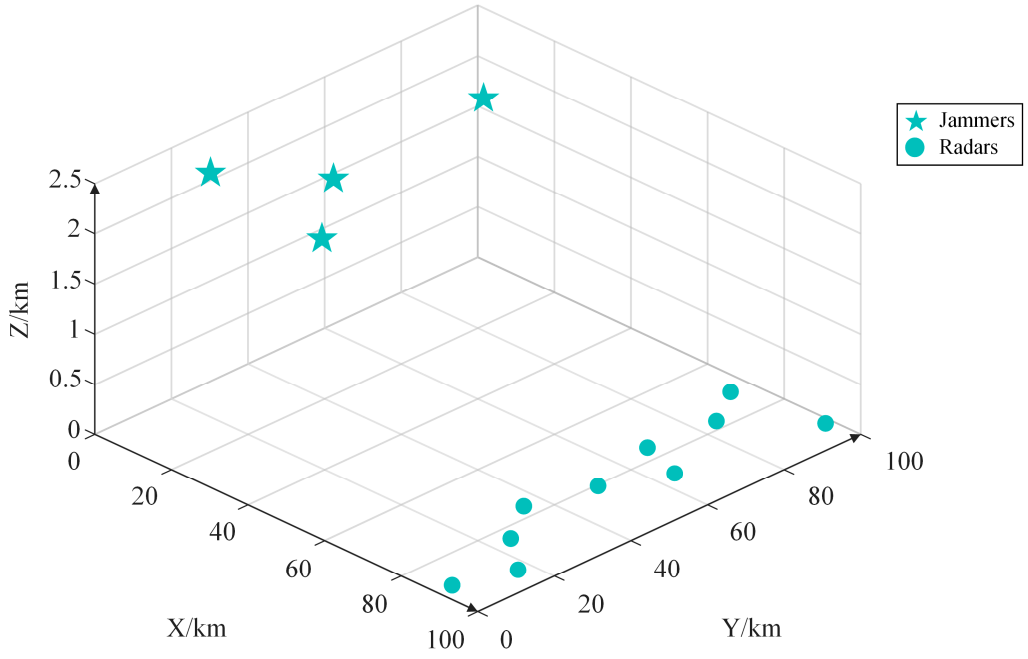


Figure 6. Visualization of the adversarial scenario.

The spatial condition constraint matrix at this moment is presented in Table 3.

Table 3. Spatial condition constraint matrix.

	Radar 1	Radar 2	Radar 3	Radar 4	Radar 5	Radar 6	Radar 7	Radar 8	Radar 9	Radar 10
Jammer 1	1	0	0	1	0	1	1	0	0	1
Jammer 2	1	0	0	1	1	0	1	0	0	1
Jammer 3	0	1	1	1	0	1	0	1	1	0
Jammer 4	0	0	1	0	0	0	0	1	0	1

Other parameter settings for the adversarial scenario are provided in Table 4.

Table 4. Other parameter settings for the adversarial scenario.

Parameters	Values
Radar Antenna Gain G_r/dB	30
Jammer Antenna Gain G_j/dB	5
Polarization Matching Loss Coefficient between Jamming and Radar Signals μ	0.5
Target's Radar Cross Section Area σ/m^2	1

The specific parameter settings for each algorithm are as follows:

For the hybrid DBO-QL algorithm, weighting factors γ_1, γ_2, γ_3, λ_1, and λ_2 were set to 0.35, 0.25, 0.4, 0.65, and 0.35, respectively. For the outer-layer DBO algorithm, the maximum iteration count corresponded to $T_{DBO} = 200$, and the population size corresponded to $N_{DBO} = 30$; for the inner-layer Q-learning algorithm, the maximum iteration count corresponded to $T_{QL} = 20,000$, and the discount factor corresponded to $\gamma = 0.8$. An adaptive strategy was employed for the learning rate α and exploration factor ε in this paper. In the initial stages of the algorithm, where exploration is emphasized, relatively larger learning rates and exploration factors are used to enable the algorithm to explore the solution space rapidly. As the training progresses, the learning rate and exploration factor gradually decrease, provoking the transition of the algorithm into the exploitation phase, where smaller values of these parameters facilitate convergence to the optimal result. The definitions of the learning rate α and exploration factor ε in this paper are provided in Equations (35) and (36), and their function plots are shown in Figure 7. The initial learning rate α_0 was set to 0.7, and the initial exploration factor ε_0 was set to 0.9.

$$\alpha = \alpha_0 \cdot e^{-\frac{\rho t}{T_{QL}}} \tag{35}$$

where α_0 is the initial learning rate and ρ is the decay factor. The decay factor ρ is related to the maximum iteration count T_{QL}, and in the simulation experiments in this paper, ρ was set to 9.

$$\varepsilon = \begin{cases} \varepsilon_0 \cdot e^{-\frac{\rho t}{T_{QL}}}, if \varepsilon > 0.1 \\ 0.1, else \end{cases} \tag{36}$$

where ε_0 is the initial exploration factor and the definition and value of ρ are the same as those in Equation (35).

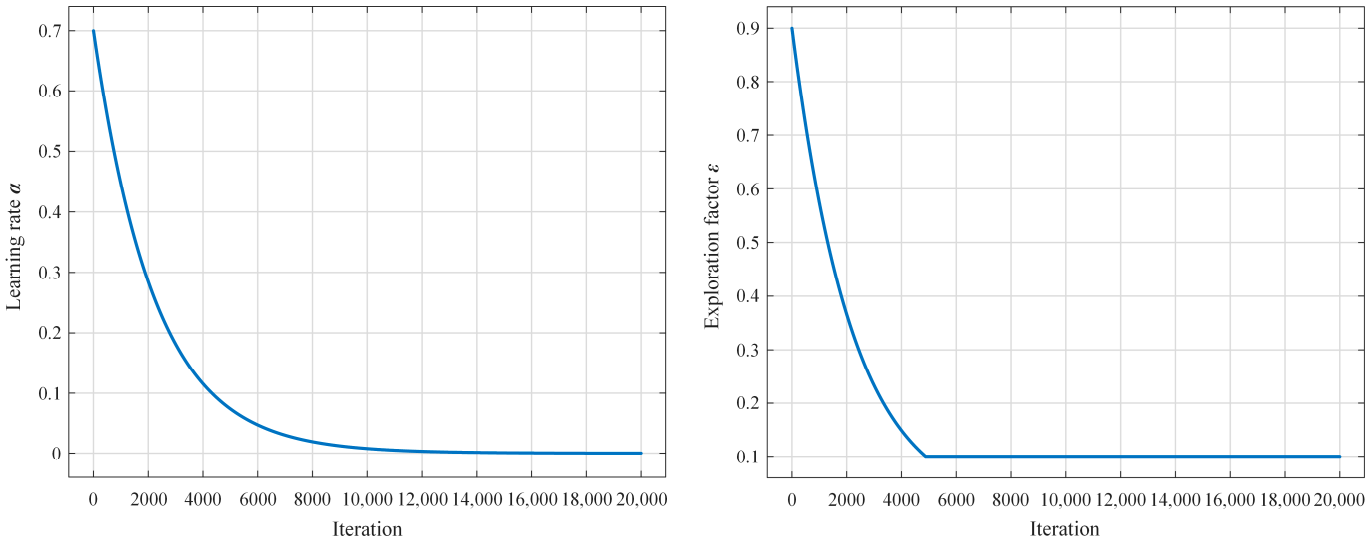


Figure 7. Function plots of learning rate α and exploration factor ε: (**a**) function plot of learning rate α; (**b**) function plot of exploration factor ε.

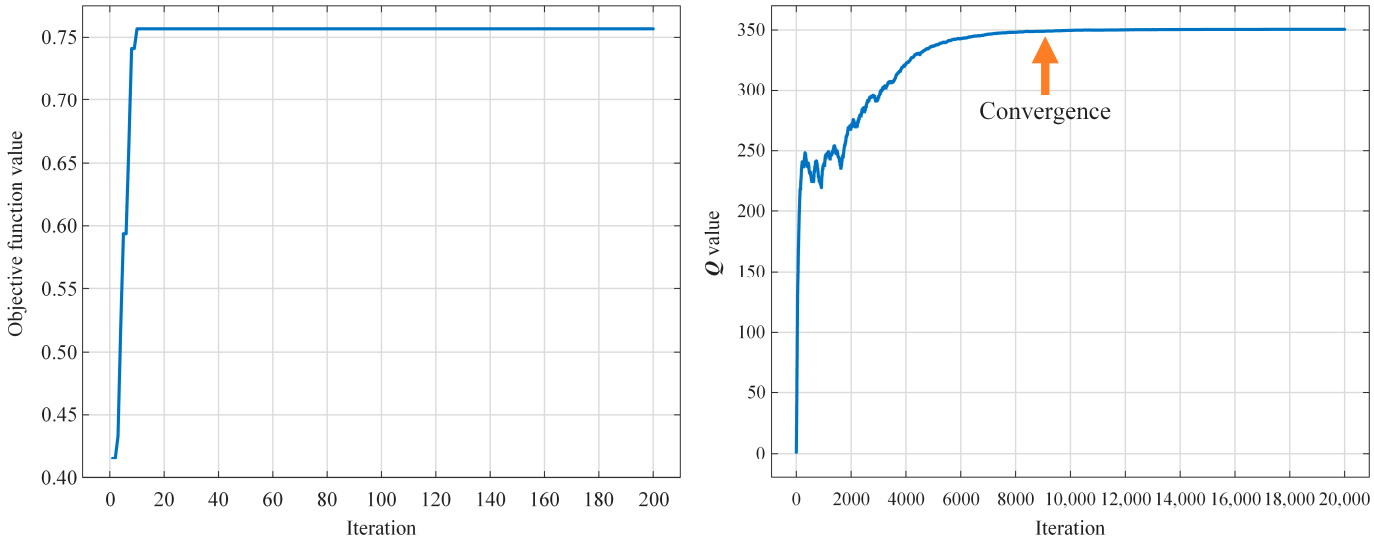
The parameter settings for the DBO algorithm are as follows: weighting factors β_1, β_2, β_3, and β_4 were set to 0.25, 0.35, 0.15, and 0.25, respectively. The maximum iteration count corresponded to $T'_{DBO} = 500$, and population size corresponded to $N'_{DBO} = 180$.

The parameter settings for the PSO algorithm are as follows: Weighting factors β_1, β_2, β_3, and β_4 were set the same as they were in the DBO algorithm. The maximum iteration count corresponded to $T_{PSO} = 500$, and population size corresponded to $N_{PSO} = 200$. The learning factor has a minimum value of $c_{1\min} = c_{2\min} = 0.5$ and a maximum value of

$c_{1max} = c_{2max} = 2.5$. The inertia factor has a minimum value of $\omega_{min} = 0.4$ and a maximum value of $\omega_{max} = 0.9$. Particle velocity has a minimum value of $v_{min} = -0.1$ and a maximum value of $v_{max} = 0.1$.

4.2. Experimental Results and Analysis

The convergence results for the hybrid DBO-QL algorithm are illustrated in Figure 8. It can be observed that the outer-layer DBO algorithm reached convergence around the 10th iteration, while the inner-layer Q-learning algorithm achieved convergence around the 9000th iteration.



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

Figure 8. Convergence curve of the hybrid DBO-QL algorithm: (**a**) the convergence curve of the outer-layer DBO algorithm; (**b**) the convergence curve of the inner-layer Q-learning algorithm.

When the algorithm converged, the jamming beam allocation matrix K and the jamming power allocation matrix P are presented in Tables 5 and 6, respectively.

Table 5. Jamming beam allocation matrix.

	Radar 1	**Radar 2**	**Radar 3**	**Radar 4**	**Radar 5**	**Radar 6**	**Radar 7**	**Radar 8**	**Radar 9**	**Radar 10**
Jammer 1	0	0	0	0	0	1	0	0	0	1
Jammer 2	1	0	0	1	1	0	1	0	0	0
Jammer 3	0	1	0	0	0	0	0	0	0	0
Jammer 4	0	0	1	0	0	0	0	1	0	0

Table 6. Jamming power allocation matrix (Unit: kW).

	Radar 1	**Radar 2**	**Radar 3**	**Radar 4**	**Radar 5**	**Radar 6**	**Radar 7**	**Radar 8**	**Radar 9**	**Radar 10**
Jammer 1	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00	0.00	0.10
Jammer 2	0.03	0.00	0.00	0.05	0.05	0.00	0.07	0.00	0.00	0.00
Jammer 3	0.00	0.11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Jammer 4	0.00	0.00	0.10	0.00	0.00	0.00	0.00	0.10	0.00	0.00

The jamming beam allocation matrix and jamming power allocation matrix shown in Tables 5 and 6 represent the optimal cooperative jamming resource allocation scheme output by the algorithm. According to this scheme, out of the 10 radars in the adversarial scenario, 9 were allocated jamming beams for jamming. Taking the first jammer as an

example, it was assigned to jam the sixth and tenth radars, with both jamming beams having a transmission power of 0.10 kW. Moreover, from Table 6, it can be observed that only the ninth radar was not assigned a jamming beam for jamming. This is due to the spatial condition constraint matrix (Table 3), wherein only the third jammer can be assigned to jam the ninth radar. However, referring to the radar parameter information and jammer parameter information in Tables 1 and 2, it is evident that the operating frequency range of the ninth radar does not overlap with the jamming frequency range of the third jammer. Their frequency domain intersection is 0, indicating that the third jammer cannot effectively jam the ninth radar. Hence, no allocation is made, reflecting a scenario that may occur in practical situations.

To provide a more intuitive display of the results of cooperative jamming resource allocation with multiple jammers, the outputs of the outer-layer DBO algorithm and the inner-layer Q-learning algorithm are depicted separately in Figures 9 and 10.

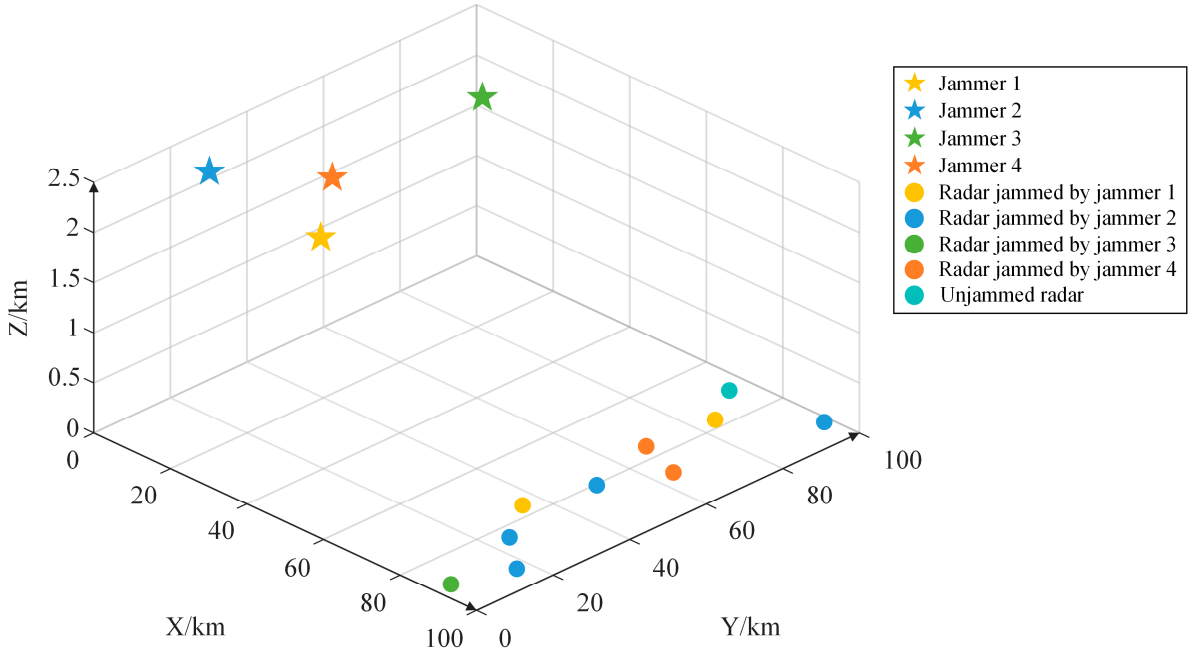


Figure 9. Visualization of DBO algorithm output.

In order to evaluate the stability and adaptability of the algorithm with respect to solving the optimization problem with complex constraints, the optimization success rate $R_{success}$ is defined in Equation (37) as follows:

$$R_{success} = \frac{N_{convergence}}{N_{total}} \times 100\%, \tag{37}$$

where $N_{convergence}$ is the number of times the algorithm converges to the optimum and N_{total} is the total number of experiments.

Under the same simulation scenario, model parameters, and hardware conditions, 200 Monte Carlo experiments were conducted on the hybrid DBO-QL algorithm, DBO algorithm, and PSO algorithm to obtain the optimal jamming benefit, average algorithm response time, and the optimization success rate, as shown in Table 7. The algorithm response time is calculated as follows: For swarm intelligence optimization algorithms, it constitutes the time elapsed from the start of the algorithm to reaching the convergence state. For the Q-learning algorithm, it consists of the time that has elapsed since the algorithm reached the convergence state, ranging from the next state input to the corresponding action output by the agent.

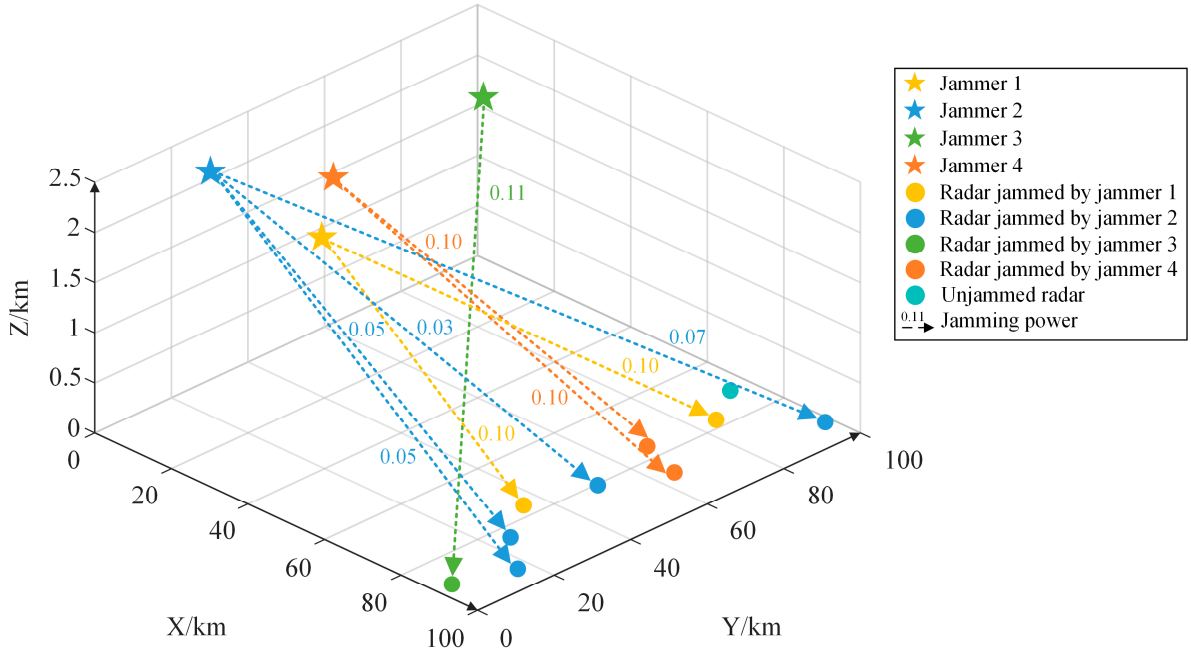


Figure 10. Visualization of Q-learning algorithm output.

Table 7. The evaluation results of different algorithms.

Evaluation Indicator	Method		
	DBO-QL	DBO	PSO
Jamming Benefit	0.68	0.66	0.64
Algorithm Response Time/s	0.11	4.15	3.21
Optimization Success Rate/%	98.93	78.31	65.84

The simulation results validated the effectiveness, superiority, and timeliness of the proposed hybrid DBO-QL algorithm. Compared with the DBO and PSO algorithms, the proposed hybrid DBO-QL algorithm improved the jamming benefit by 3.03% and 6.25%, reduced the algorithm response time by 97.35% and 96.57%, and improved the optimization success rate by 26.33% and 50.26%, respectively. This is because, for the DBO algorithm and PSO algorithm, the increase in constraints entails an increase in algorithm optimization difficulty, so when facing problems with complex constraints, these algorithms often need to expand the number of populations to increase their own optimization success rates, but the expansion of the number of populations will bring about the prolongation of the algorithm response time; therefore, it is necessary to balance the relationship between the algorithm response time and the optimization success rate when using these algorithms. A benefit is provided by the hybrid DBO-QL algorithm's unique model, where the outer-layer DBO algorithm has a discrete solution space with only two values (0 and 1) for each dimension. This significantly simplified the optimization difficulty, resulting in improved convergence and stability and reduced time consumption. For the inner-layer Q-learning algorithm, once the agent is trained, it can rapidly select the optimal action for a given input state at different times based on the trained policy. Unlike the other two swarm intelligence optimization algorithms, the Q-learning algorithm does not require repetitive computations for each iteration, leading to a significant reduction in response time.

Table 7 also reveals that, compared to the PSO algorithm, the DBO algorithm provides a greater jamming benefit but at the cost of a longer response time. This is due to the fact that the DBO algorithm incorporates optimization strategies inspired by the rolling, dancing, foraging, stealing, and reproductive behaviors of dung beetles, which enhance its convergence effectiveness but prolong the algorithm response time.

5. Conclusions

This paper addresses the problem of cooperative jamming resource allocation with multiple jammers in a "many-to-many" scenario, proposing a method based on evolutionary reinforcement learning. This method comprehensively considers information from spatial, frequency, and energy domains to construct constraints and an objective function. It characterizes the jamming resource allocation scheme through the jamming beam allocation matrix and jamming power allocation matrix and optimizes these matrices using the outer-layer DBO algorithm and the inner-layer Q-learning algorithm, respectively. This approach achieves cooperative jamming resource allocation among multiple jammers by leveraging radar parameter information and jammer parameter information. In the simulation experiments, commonly used swarm intelligence optimization algorithms, specifically the DBO algorithm and the PSO algorithm, were selected for comparison in terms of jamming benefit, algorithm response time, and optimization success rate. The results demonstrate that the proposed algorithm outperforms the other two swarm intelligence optimization algorithms, obtaining higher jamming benefits and optimization success rates and showing significant advantages in algorithm response time.

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

Funding: This research was funded by the Shanghai Aerospace Science and Technology Innovation Fund under grant number SAST2022-063.

Data Availability Statement: The data can be obtained by contacting the corresponding author.

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

References

1. Gurbuz, S.Z.; Griffiths, H.D.; Charlish, A.; Rangaswamy, M.; Greco, M.S.; Bell, K. An overview of cognitive radar: Past, present, and future. *IEEE Aerosp. Electron. Syst. Mag.* **2019**, *34*, 6–18. [CrossRef]
2. Haykin, S. New generation of radar systems enabled with cognition. In Proceedings of the 2010 IEEE Radar Conference, Arlington, VA, USA, 10–14 May 2010; p. 1.
3. Haykin, S. Cognitive radar: A way of the future. *IEEE Signal Process. Mag.* **2006**, *23*, 30–40. [CrossRef]
4. Darpa, A. Behavioral learning for adaptive electronic warfare. In *Darpa-BAA-10-79*; Defense Advanced Research Projects Agency: Arlington, TX, USA, 2010.
5. Haystead, J. DARPA seeks proposals for adaptive radar countermeasures. *J. Electron. Def.* **2012**, *2012*, 16–18.
6. du Plessis, W.P.; Osner, N.R. Cognitive electronic warfare (EW) systems as a training aid. In Proceedings of the Electronic Warfare International: Conference India (EWCI), Bangalore, India, 13–16 February 2018; pp. 1–7.
7. Wang, X.; Fei, Z.; Huang, J.; Zhang, J.A.; Yuan, J. Joint resource allocation and power control for radar interference mitigation in multi-UAV networks. *Sci. China Inf. Sci.* **2021**, *64*, 182307. [CrossRef]
8. Ren, Y.; Li, B.; Wang, H.; Xu, X. A novel cognitive jamming architecture for heterogeneous cognitive electronic warfare networks. In *Information Science and Applications: ICISA 2019*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 97–104.
9. Xiang, C.-W.; Jiang, Q.-S.; Qu, Z. Modeling and algorithm of dynamic resource assignment for ESJ electronic warfare aircraft. *Command Control Simul.* **2017**, *39*, 85–89.
10. Haigh, K.; Andrusenko, J. *Cognitive Electronic Warfare: An Artificial Intelligence Approach*; Artech House: London, UK, 2021.
11. Zhang, C.; Wang, L.; Jiang, R.; Hu, J.; Xu, S. Radar Jamming Decision-Making in Cognitive Electronic Warfare: A Review. *IEEE Sens. J.* **2023**, *23*, 11383–11403. [CrossRef]
12. Zhou, H. An introduction of cognitive electronic warfare system. In Communications, Signal Processing, and Systems: Proceedings of the 2018 CSPS Volume III: Systems 7th, Dalian, China, 14–16 July 2020; Springer: Singapore, 2020; pp. 1202–1210.
13. Qingwen, Q.; Wenfeng, D.; Meiqing, L.; Yang, Y. Cooperative jamming resource allocation of UAV swarm based on multi-objective DPSO. In Proceedings of the 2018 Chinese Control And Decision Conference (CCDC), Shenyang, China, 9–11 June 2018; pp. 5319–5325.
14. Gao, Y.; Li, D.-S. Electronic countermeasures jamming resource optimal distribution. In Information Technology and Intelligent Transportation Systems: Volume 2, Proceedings of the 2015 International Conference on Information Technology and Intelligent Transportation Systems ITITS 2015, Xi'an, China, 12–13 December 2015; Springer: Cham, Switzerland, 2017; pp. 113–121.

15. Liu, X.; Li, D. Analysis of cooperative jamming against pulse compression radar based on CFAR. *EURASIP J. Adv. Signal Process.* **2018**, *2018*, 69. [CrossRef]
16. Xiong, X.; Zheng, K.; Lei, L.; Hou, L. Resource allocation based on deep reinforcement learning in IoT edge computing. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 1133–1146. [CrossRef]
17. Shi, W.; Li, J.; Wu, H.; Zhou, C.; Cheng, N.; Shen, X. Drone-cell trajectory planning and resource allocation for highly mobile networks: A hierarchical DRL approach. *IEEE Internet Things J.* **2020**, *8*, 9800–9813. [CrossRef]
18. Zhao, B.; Liu, J.; Wei, Z.; You, I. A deep reinforcement learning based approach for energy-efficient channel allocation in satellite Internet of Things. *IEEE Access* **2020**, *8*, 62197–62206. [CrossRef]
19. Lei, W.; Ye, Y.; Xiao, M. Deep reinforcement learning-based spectrum allocation in integrated access and backhaul networks. *IEEE Trans. Cogn. Commun. Netw.* **2020**, *6*, 970–979. [CrossRef]
20. He, C.; Hu, Y.; Chen, Y.; Zeng, B. Joint power allocation and channel assignment for NOMA with deep reinforcement learning. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 2200–2210. [CrossRef]
21. Alwarafy, A.; Ciftler, B.S.; Abdallah, M.; Hamdi, M. DeepRAT: A DRL-based framework for multi-RAT assignment and power allocation in HetNets. In Proceedings of the 2021 IEEE International Conference on Communications Workshops (ICC Workshops), Montreal, QC, Canada, 14–23 June 2021; pp. 1–6.
22. Meng, F.; Chen, P.; Wu, L.; Cheng, J. Power allocation in multi-user cellular networks: Deep reinforcement learning approaches. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 6255–6267. [CrossRef]
23. Zou, W.Q.; Niu, C.Y.; Liu, W.; Wang, Y.Y.; Zhan, J.Q. Combination search strategy-based improved particle swarm optimisation for resource allocation of multiple jammers for jamming netted radar system. *IET Signal Process.* **2023**, *17*, e12198. [CrossRef]
24. Wu, Z.; Hu, S.; Luo, Y.; Li, X. Optimal distributed cooperative jamming resource allocation for multi-missile threat scenario. *IET Radar Sonar Navig.* **2022**, *16*, 113–128. [CrossRef]
25. Jiang, H.; Zhang, Y.; Xu, H. Optimal allocation of cooperative jamming resource based on hybrid quantum-behaved particle swarm optimisation and genetic algorithm. *IET Radar Sonar Navig.* **2017**, *11*, 185–192. [CrossRef]
26. Lu, D.-j.; Wang, X.; Wu, X.-t.; Chen, Y. Adaptive allocation strategy for cooperatively jamming netted radar system based on improved cuckoo search algorithm. *Def. Technol.* **2023**, *24*, 285–297. [CrossRef]
27. Xing, H.; Xing, Q.; Wang, K. A Joint Allocation Method of Multi-Jammer Cooperative Jamming Resources Based on Suppression Effectiveness. *Mathematics* **2023**, *11*, 826. [CrossRef]
28. Yao, Z.; Tang, C.; Wang, C.; Shi, Q.; Yuan, N. Cooperative jamming resource allocation model and algorithm for netted radar. *Electron. Lett.* **2022**, *58*, 834–836. [CrossRef]
29. Xing, H.-x.; Wu, H.; Chen, Y.; Wang, K. A cooperative interference resource allocation method based on improved firefly algorithm. *Def. Technol.* **2021**, *17*, 1352–1360. [CrossRef]
30. Gronauer, S.; Diepold, K. Multi-agent deep reinforcement learning: A survey. *Artif. Intell. Rev.* **2022**, *55*, 895–943. [CrossRef]
31. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
32. Das, A.; Kottur, S.; Moura, J.M.; Lee, S.; Batra, D. Learning cooperative visual dialog agents with deep reinforcement learning. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2951–2960.
33. Li, S.; Liu, G.; Zhang, K.; Qian, Z.; Ding, S. DRL-Based Joint Path Planning and Jamming Power Allocation Optimization for Suppressing Netted Radar System. *IEEE Signal Process. Lett.* **2023**, *30*, 548–552. [CrossRef]
34. Yue, L.; Yang, R.; Zuo, J.; Zhang, Y.; Li, Q.; Zhang, Y. Unmanned Aerial Vehicle Swarm Cooperative Decision-Making for SEAD Mission: A Hierarchical Multiagent Reinforcement Learning Approach. *IEEE Access* **2022**, *10*, 92177–92191. [CrossRef]
35. Lü, S.; Han, S.; Zhou, W.; Zhang, J. Recruitment-imitation mechanism for evolutionary reinforcement learning. *Inf. Sci.* **2021**, *553*, 172–188. [CrossRef]
36. Xue, F.; Hai, Q.; Dong, T.; Cui, Z.; Gong, Y. A deep reinforcement learning based hybrid algorithm for efficient resource scheduling in edge computing environment. *Inf. Sci.* **2022**, *608*, 362–374. [CrossRef]
37. Asghari, A.; Sohrabi, M.K. Combined use of coral reefs optimization and reinforcement learning for improving resource utilization and load balancing in cloud environments. *Computing* **2021**, *103*, 1545–1567. [CrossRef]
38. Zhang, C.; Song, Y.; Jiang, R.; Hu, J.; Xu, S. A Cognitive Electronic Jamming Decision-Making Method Based on Q-Learning and Ant Colony Fusion Algorithm. *Remote Sens.* **2023**, *15*, 3108. [CrossRef]
39. Xue, J.; Shen, B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. *J. Supercomput.* **2023**, *79*, 7305–7336. [CrossRef]
40. Clifton, J.; Laber, E. Q-learning: Theory and applications. *Annu. Rev. Stat. Its Appl.* **2020**, *7*, 279–301. [CrossRef]