

Article

VAM-Enhanced Deep Reinforcement Learning for Cooperative Jamming Task Allocation

Yulian Song, Xiaoshuai Li *, Yang Pan, Hongwei Liu and Junan Yang *

College of Electronic Engineering, National University of Defense Technology, Hefei 230027, China; syl@nudt.edu.cn (Y.S.); panyang22a@nudt.edu.cn (Y.P.); liuhongwei22@nudt.edu.cn (H.L.)

* Correspondence: xiaoshuai.li@nudt.edu.cn (X.L.); yangjunan@ustc.edu (J.Y.)

Abstract

This paper addresses the cooperative jamming task allocation problem for multiple jammers against multiple communication targets in dynamic electronic warfare environments. Traditional algorithms struggle with adaptability and slow decision-making. To overcome these limitations, we propose a deep reinforcement learning (DRL) method enhanced by an improved Vogel's approximation method (VAM) pre-training strategy, where VAM incorporates situational matrices for initial allocation. The proposed approach aims to maximize the total jamming situational value by intelligently assigning optimal target combinations to each multi-beam jammer. Specifically, the model evaluates the situational value of each target by integrating factors including the distance, target firepower, and threat levels, while adhering to system constraints of both jamming and target platforms. To meet the real-time decision-making requirements in dynamic adversarial environments, we integrate VAM with the proximal policy optimization (PPO) algorithm, leveraging human knowledge to accelerate the training process of DRL. Simulation results demonstrate that the proposed algorithm improves both the training efficiency and decision-making timeliness of the jamming allocation model, achieving cumulative reward increases of 38.45% and 13.86% over the respective baselines, while ensuring target coverage and effectively avoiding redundant or excessive jamming.

Keywords: collaborative jamming; target allocation; deep reinforcement learning; Vogel's approximation method

1. Introduction

Facing increasingly complex battlefield environments and diverse operational tasks, the limited flight capabilities and resource constraints of a single jammer severely restrict its functionality, making it increasingly difficult to execute multi-dimensional, large-scale missions. Moreover, it may even fail to operate due to ground-based threats or air defense countermeasures [1]. Collaborative operations using multiple jammers like unmanned aerial vehicles (UAVs) can effectively address this challenge and significantly enhance the overall combat system's effectiveness. This trend toward swarm-based cooperative warfare is an inevitable evolution in modern combat [2].

Multi-UAV cooperative target allocation refers to the assignment of mission tasks to UAVs within a formation based on reconnaissance data and mission requirements, aiming to maximize overall combat effectiveness or minimize operational costs [3]. When formulating allocation schemes, it is essential to consider not only the UAVs' inherent

Academic Editors: Jie Yang, Yuhe Shi and Yuanyuan Zhang

Received: 15 December 2025

Revised: 23 January 2026

Accepted: 2 February 2026

Published: 5 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

constraints but also the interdependencies among tasks. Given the rapidly changing battlefield dynamics, situational shifts may degrade the original allocation scheme's efficiency or even render it entirely ineffective. Thus, the multi-UAV cooperative target allocation system must exhibit high adaptability to dynamic conditions. This problem is a classic integer programming challenge [4], equivalent to the assignment problem in operations research. The primary research difficulties lie in modeling and solving methods. Most existing literature employs convex optimization, nonlinear programming (NLP), or intelligent optimization algorithms to solve static target allocation problems under the assumption of a stationary battlefield environment from which an optimal solution can be derived given current conditions. However, real-world combat environments are highly dynamic, making real-time dynamic target allocation strategies more practically valuable. Therefore, this study focuses on addressing the continuous decision-making challenge in dynamic target allocation scenarios.

Motivated by these concerns, our work aims to address the dual requirements of decision timeliness and dynamic adaptability in electronic countermeasure scenarios. The key contributions of this study on cooperative jamming target allocation can be outlined as follows.

1. Application-Specific Contributions

- We mathematically abstract and model a “many-to-many” scenario where multi-beam jammers cooperatively interfere with multiple targets, unlike previous works limited to one-to-one or many-to-one target allocation scenarios.
- Our model comprehensively considers factors such as the distance between jammers and targets, target firepower, and threat levels to calculate precise situational values of the target environment. By incorporating system platform constraints from both the jamming and target sides, we derive allocation schemes that maximize jamming effectiveness.

2. Methodological Innovations

- We employ a deep reinforcement learning (DRL) approach—the Proximal policy optimization (PPO) algorithm, to replace traditional static target allocation methods. Additionally, we creatively applied the multi-branch structure to the actor network to reduce the decision-making dimension.
- We introduce a heuristic method from resource allocation—the Vogel's approximation method (VAM)—which leverages pre-generated state-action pairs to accelerate the learning process of agents in the target domain, thereby enhancing overall system responsiveness and resource scheduling performance.

3. Through experiments, we demonstrate that this DRL algorithm, enhanced by heuristic-based experience optimization, improves the model's adaptability to new environments while reducing decision time. It achieves faster convergence and higher reward values, enabling real-time allocation of optimal jamming schemes for each jammer to maximize total jamming effectiveness while fully utilizing available resources.

The organization of this paper is as follows. In Section 2, we analyze and discuss the existing related work. In Section 3, we present the mathematical modeling of the “many-to-many” jamming system, and Section 4 provides detailed explanations of the VAM and proposed PPO algorithm. Section 5 presents experimental results to validate the effectiveness of the proposed method. Finally, we summarize the findings and discuss future work.

2. Related Works

2.1. Classical Optimization Methods

In recent years, mathematical programming (MP) [5,6], game theory [7], expert systems [8], and evolutionary algorithms (EA) [9,10] have been widely applied to solve weapon–target assignment (WTA) problems. However, due to conflicts between optimization objectives and constraints, these methods have gradually proven inadequate for multi-objective optimization.

One of the most representative works was proposed by Kline et al. [11], where the objective function was formulated using weighted target survivability. Building on this reference, numerous studies have incorporated additional allocation objectives, leading to various similar variants. Luo et al. [12] considered missile launch costs, while Volle et al. [13] introduced the concept of target priority. Ref. [14] proposed an improved reference-point-based non-dominated sorting genetic algorithm (NSGA) by leveraging threat and advantage matrices derived from battlefield situational assessments. The study employed binary coded attack schemes to optimize the initial population and introduced adaptive mutation, crossover strategies, and population update mechanisms. Ref. [15] designed a novel auxiliary metric based on relative integral distance to address cooperative attack and target allocation concurrently within a unified single-stage framework, ensuring that each target is destroyed by at least one attacker while maximizing the utilization of all attackers for optimal performance. Ref. [16] introduced a new method based on differential dynamic programming that embeds the target allocation problem within terminal constraints. To find locally optimal allocation schemes, a mixed-integer quadratic programming problem must be solved in each iteration.

Although effective under specific assumptions, the high dynamism of real combat environments renders these approaches inadequate for scenarios requiring continuous decision making, underscoring the urgent need for real-time dynamic target allocation strategies. This study acknowledges the contributions of classical methods to static WTA problems and shifts the focus to the challenge of continuous decision making in dynamic environments.

2.2. Classical Reinforcement Learning Methods

In terms of continuous decision-making, reinforcement learning (RL) iteratively adjusts policy parameters through an agent's real-time interaction with the environment, achieving learning by trial-and-error and self-optimization without requiring prior knowledge to maximize rewards for a given task [17]. This adaptability to dynamic environments and flexibility in decision-making have made DRL highly attractive for WTA applications.

Ref. [18] proposed a two-stage optimization algorithm, QL-GA, combining Q-learning and genetic algorithms to enhance global optimality. Ref. [19] introduced a hybrid Monte Carlo (MC) and Q-learning algorithm, which, despite slower convergence, achieved superior performance metrics. Ref. [20] developed an incremental search method for radar network target assignment within the deep Q-network (DQN) framework, demonstrating improved performance in multi-target and saturation attack scenarios. Ref. [21] employed a multi-head Q-value network to improve WTA model efficiency in exploration. Additionally, ref. [22] demonstrated significant results by solving dynamic WTA (DWTA) with DQN. Ref. [23] proposed a novel framework integrating DQN with a multi-objective artificial bee colony (MOABC) to address a multi-objective WTA (MO-WTA) with strong scalability. Ref. [24] introduced a multi-objective evolutionary algorithm (MOEA) incorporating a DQN-based adaptive mutation operator and a greedy

crossover operator, significantly improving solution quality for MO-WTA. These studies validate the feasibility of RL in applications with well-defined state and action spaces.

Existing RL research has predominantly focused on one-to-one or many-to-one target assignment, and systematic, specialized investigation of many-to-many target matching remains limited. Agents still face substantial challenges in more complex multi-objective assignment problems. RL methods are often hampered by the exploration–exploitation dilemma, particularly when problem characteristics are unknown or prior knowledge is insufficiently exploited [24]. Consequently, improving the generalization capability of RL models and designing more efficient training mechanisms is critical. Ref. [25] integrated DRL and long short-term memory (LSTM) networks to enhance many-to-many task allocation efficiency, effectively addressing the collaboration instability caused by unknown future information and offering a novel approach to achieving stable multi-agent collaborative task allocation in highly dynamic environments.

2.3. Knowledge-Guided Reinforcement Learning

As DRL relies on continuous interaction with the environment to optimize policies, its application in highly dynamic adversarial scenarios involves prolonged training cycles and substantial computational overhead, particularly during early training phases when agent policies remain immature. To address this challenge, incorporating human priors or knowledge transferred from related tasks into the learning process has emerged as an effective approach.

Ref. [26] proposed a hybrid evolutionary–RL framework combining the dung beetle optimizer algorithm and Q-learning (DBO-QL) to address the problem of cooperative jamming beam allocation and power configuration. This structured knowledge-embedding approach substantially reduces the optimization difficulty under multiple constraints. Using the EA to guide and compress the RL search space, this approach offers a new paradigm for achieving second-level resource allocation responses in dynamic adversarial scenarios. By combining Q-learning with distribution estimation learning (EoD) in a self-driven manner, performance on constrained multi-objective optimization problems with irregular frontiers was significantly improved [27]. Ref. [28] proposed a knowledge-guided RL (KG-RL) method that integrates artificial potential fields (APFs) to improve sample efficiency and operational safety, where the APF generates concise demonstration trajectories during pretraining to provide foundational knowledge for the agent. Similarly, ref. [29] introduced an APF-based RL (APF-RL) approach that injects human knowledge to accelerate state feature learning, while generating intrinsic rewards to reduce invalid exploration and guide policy optimization. Ref. [30] developed a semantic knowledge-guided fuzzy DQN (KFDQN), embedding knowledge into learning objectives to reduce uncertainty, thereby enhancing learning efficiency, training stability, and performance. Ref. [31] proposed the multi-objective genetic programming-assisted stochastic DRL (MOGP-SDRL) framework, where genetic programming-generated intelligent features are used as priors to augment RL, further extending the depth of knowledge guidance.

These methods, by introducing prior knowledge, effectively improve learning efficiency and model performance of RL.

4. In conclusion, there are numerous works studying the WTA problem and the multi-agent collaboration problem. The core of this study is to address dynamic, many-to-many target assignment problems, which markedly differ from classical WTA formulations that focus on static, single-stage optimization. Classical WTA models (e.g., [5,6,9–11]) provide the theoretical foundation for problem definition and basic optimization objectives in this work; however, by introducing DRL, the paradigm shifts from seeking a single static optimum to making continuous decisions in dynamic environments. Unlike research that emphasizes complex coordination logic among

agents in multi-agent systems—typically concentrating on inter-agent communication, negotiation, and similar behaviors—this study, within a many-to-many framework, places greater emphasis on modeling and solving larger-scale, dynamic matching relationships between jamming resources and targets under multi-objective optimization constraints. Methodologically, therefore, this study constitutes an extension of classical WTA to dynamic scenarios; in terms of problem focus, it differs from purely multi-agent coordination research by concentrating on efficient, real-time solutions for many-to-many mapping relationships.

3. System Model

In this work, we investigate the WTA problem involving multiple jammers allocating interference resources to multiple targets in modern localized battlefields. As illustrated in Figure 1, we consider a scenario where the jamming party deploys N multi-beam jamming UAVs, each equipped with J beams capable of simultaneously interfering with up to J target links using spot jamming techniques. The set of jammers is denoted as $n = \{n_1, n_2, \dots, n_N\}$. Each jammer operates under limited maximum power, which is equally distributed among its jamming beams. The communication party consists of M transmitter–receiver pairs, represented as $m = \{m_1, m_2, \dots, m_M\}$, where each communication link can be simultaneously jammed by a maximum of K beams. The priority of jamming resource allocation is positively correlated with the threat coefficient of each target, denoted as $\{\omega_1, \omega_2, \dots, \omega_M\}$, which is assumed to be obtained through intelligence reconnaissance.

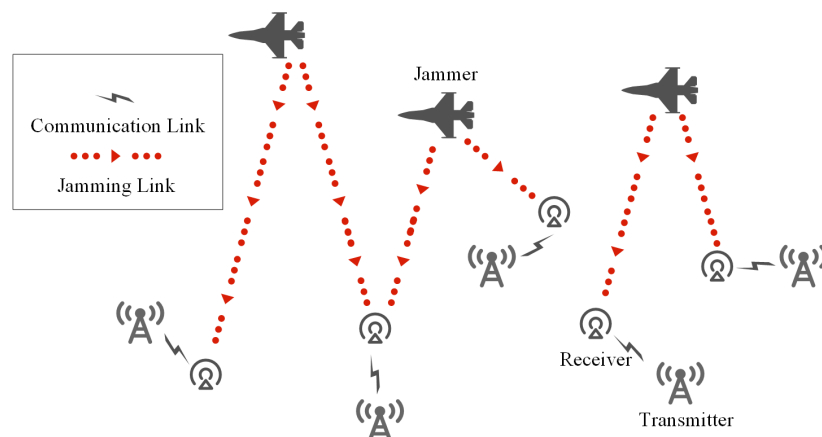


Figure 1. Multi-beam jammers cooperatively interfere with multiple targets.

During the jamming task allocation phase, we comprehensively consider multiple factors, including the threat level of target links, the distance between UAVs and target receivers, and the firepower configuration of targets. Under the constraint of limited transmission power for the UAV swarm, we formulate a “many-to-many” target allocation problem for cooperative jamming resource deployment with the objective of maximizing jamming effectiveness. This leads to the following optimization model:

$$\max \sum_{i=1}^N \sum_{j=1}^M x_{ij} \cdot F(\omega_j, d_{ij}^{3D}, c_{ij}^{2A}), \quad (1)$$

$$s. t. \quad C1: x_{ij} \in \{0, 1, 2, \dots, J\}, i = 1, 2, \dots, N; j = 1, 2, \dots, M. \quad (2)$$

$$C2: \sum_{i=1}^N x_{ij} \leq K, j = 1, 2, \dots, M. \quad (3)$$

$$C3: \sum_{j=1}^M x_{ij} \leq J, i = 1, 2, \dots, N. \quad (4)$$

$$C4: P_{ij} \leq P_{max}, i = 1, 2, \dots, N; j = 1, 2, \dots, M. \quad (5)$$

where x_{ij} serves as the jamming resource allocation factor, representing the number of beams allocated by the i -th UAV to jam the j -th target link. $x_{ij} = 0$ indicates no jamming. The composite jamming effectiveness function $F(\omega_j, d_{ij}^{3D}, c_{ij}^{2A})$ incorporates three key parameters. The parameters include ω_j for the threat level of the j -th target, d_{ij}^{3D} for the 3D spatial distance between the i -th UAV and j -th target, and c_{ij}^{2A} for the survivability index of the i -th UAV when engaging the j -th target, reflecting the probability of UAV survival and for which a higher value indicates better survivability. P_{ij} represents the jamming power allocated and P_{max} represents the maximum available jamming power of UAVs.

This multi-target allocation problem belongs to the NP-hard class, where traditional optimization algorithms often fail to find optimal solutions within reasonable timeframes due to high computational dimensionality. To balance solution quality and computation time, we propose a novel approach combining heuristic algorithms from operations research with DRL. Specifically, we employ VAM to initialize the DRL model, using empirical solutions to guide the learning process of the PPO algorithm. An innovative action space design is implemented to reduce problem dimensionality, thereby improving convergence speed and training effectiveness.

In the studied adversarial scenario involving the cooperative jamming of multiple ground targets, each target plays a distinct role with unique importance and operational value. To address this, we quantify target situational awareness through a comprehensive evaluation matrix that serves as the basis for jamming task allocation. This mechanism aims to simultaneously minimize UAV damage while maximizing jamming operational effectiveness. The target situational matrix [32] is constructed based on three key elements:

- **Survivability:** Considering potential countermeasures against UAVs, we assume targets may employ precision fire attacks (with default attack frequencies of 1, 2, or 3). The higher the probability of attack damage, the smaller the survivability.
- **Distance-based threat:** Closer proximity to UAVs indicates higher threat levels.
- **Operational value:** Incorporating reconnaissance and jamming capabilities, the threat level of targets is assessed through expert evaluation of prior information.

The mathematical formulation is expressed as follows:

$$c_{ij} = a \cdot c_{ij}^{2A} + b \cdot c_{ij}^{2D} + c \cdot c_j^{2E}, \quad (6)$$

$$c_{ij}^{2A} = (1 - P_j^{hit} \cdot P_i^{dam})^\vartheta, \quad (7)$$

$$c_{ij}^{2D} = 1 - e^{-1/d_{ij}^{3D}}, \quad (8)$$

$$c_j^{2E} = \{\omega_1, \omega_2, \dots, \omega_j, \dots, \omega_M\}, \quad (9)$$

$$Sv_{N \times M} = \begin{bmatrix} c_{11} & \cdots & c_{1j} & \cdots & c_{1M} \\ \vdots & & \vdots & & \vdots \\ c_{i1} & \cdots & c_{ij} & \cdots & c_{iM} \\ \vdots & & \vdots & & \vdots \\ c_{N1} & \cdots & c_{Nj} & \cdots & c_{NM} \end{bmatrix}, \quad (10)$$

where c_{ij} represents the battlefield situational value, $a/b/c$ are weighting factors, P_j^{hit} denotes weapon hit probability, P_i^{dam} indicates UAV destruction probability upon hit, ϑ

represents attack frequency, and $Sv_{N \times M}$ constitutes the situational matrix. Based on the derived situational matrix $Sv_{N \times M}$, we formulate the jamming target allocation problem with the objective of maximizing overall battlefield situational value, as follows:

$$\max \sum_{i=1}^N \sum_{j=1}^M x_{ij} c_{ij}, \quad (11)$$

s.t. $C1, C2, C3, C4$;

where $C1, C2, C3, C4$ represent the operational constraints for UAV jamming platforms as previously discussed. Note that the objective function in Equation (1) is generalized, and in Equation (11), we use the situational value c_{ij} to represent the integrated effect of $F(\omega_j, d_{ij}^{3D}, c_{ij}^{2A})$, where c_{ij} is derived from the factors in F via the situational matrix. This clarifies that Equation (11) is a concrete form of Equation (1) for optimization.

4. Deep Reinforcement Learning Approach with Heuristic Algorithm Optimization

4.1. Improved Vogel's Approximation Method Algorithm Based on Situational Matrix

VAM is a heuristic algorithm for transportation problems (TPs) that allocates resources between multiple supply and demand points to minimize total transportation costs [33]. It is commonly used to find near-optimal feasible solutions in logistics or resource allocation scenarios where transportation cost minimization is required. The algorithm calculates penalties (the difference between the smallest and second-smallest costs) for each row and column, then prioritizes resource allocation in the row or column with the highest penalty. Through this process of local optimal selection, it progressively reduces the problem scale to quickly generate feasible solutions [34]. While this method aims to produce a good initial solution, it may be further refined using other optimization techniques like the stepping-stone method or modified distribution method [35].

The rationale for selecting VAM over other transportation or assignment heuristics is grounded in its specific suitability for the asymmetric nature of our “many-to-many” jamming resource allocation problem. Unlike the Hungarian algorithm, which is designed for balanced assignment problems requiring a square cost matrix (i.e., equal number of agents and tasks) [36], our scenario involves N jammers and M targets, where N and M may be unequal. VAM, originally developed for transportation problems, naturally handles such asymmetries by considering supply and demand constraints. Compared with greedy initialization strategies that may converge quickly to local optima due to their myopic cost minimization, VAM's mechanism of calculating row and column penalties ensures a more globally informed initial solution by prioritizing allocations with the highest opportunity cost differences. This approach provides a higher-quality starting point for the subsequent DRL training, balancing solution quality and computational efficiency. In contrast, Hungarian algorithm variants for non-square matrices often require complex extensions that increase computational overhead, while pure greedy methods lack the global perspective needed for effective exploration guidance in high-dimensional spaces. Thus, VAM is ideal for initializing the PPO algorithm in dynamic environments where real-time adaptability is critical.

In our specific application scenario, the many-to-many resource allocation problem features an unequal number of jammers and targets, making the TP approach more appropriate. Here, multiple jammers serve as “Supply” points providing jamming beams, while multiple targets act as “Demand” points consuming these beams, with the situational matrix serving as the “Cost” basis for resource allocation. Our objective is to maximize total jamming effectiveness. To apply VAM, we reformulate (11) as follows:

$$\min Z = - \sum_{i=1}^N \sum_{j=1}^M x_{ij} c_{ij} \quad (12)$$

s.t. $C1, C2, C3, C4$;

where the supply vector represents the number of available jamming beams per UAV, and the demand vector indicates the maximum allowable jamming beams per target. Using the negative values of the situational matrix as the cost matrix, the VAM implementation follows the procedure detailed in Algorithm 1.

Algorithm 1: Vogel's Approximation Method for Initial Basic Feasible Solution

Input: Cost matrix $Sv_{N \times M} = [c_{ij}]_{N \times M}$, Supply vector $\alpha = [a_1, a_2, \dots, a_N]$, Demand vector $\beta = [b_1, b_2, \dots, b_M]$

Output: Initial basic feasible solution $X = x_{ij}_{N \times M}$, Total cost

```

1: Initialize:  $X \leftarrow 0_{N \times M}, TotalCost \leftarrow 0, UnassignedSupply \leftarrow \alpha,$ 
    $UnassignedDemand \leftarrow \beta$ 
2: While  $\exists UnassignedSupply_i > 0$  and  $\exists UnassignedDemand_j > 0$ :
3:   a. Compute Penalty Values:
4:   For each row  $i$ :
5:      $RowPenalty_i = SecondMin(C[i, :]) - Min(C[i, :])$ 
6:   For each column  $j$ :
7:      $ColPenalty_j = SecondMin(C[:, j]) - Min(C[:, j])$ 
8:   b. Select Maximum Penalty:
9:    $MaxPenalty \leftarrow \max(RowPenalty \cup ColPenalty)$ 
10:  If  $MaxPenalty$  corresponds to row  $i$ :
11:     $j^* \leftarrow \arg \min_j C[i, j]$ 
12:  Else (column  $j$ ):
13:     $i^* \leftarrow \arg \min_i C[i, j]$ 
14:  c. Allocate Resources:
15:   $x_{i^* j^*} \leftarrow \min(UnassignedSupply_{i^*}, UnassignedDemand_{j^*})$ 
16:  Update  $UnassignedSupply_{i^*} \leftarrow UnassignedSupply_{i^*} - x_{i^* j^*}$ 
17:  Update  $UnassignedDemand_{j^*} \leftarrow UnassignedDemand_{j^*} - x_{i^* j^*}$ 
18:  If  $UnassignedSupply_{i^*} = 0$ :
19:    Set  $C[i^*, :] \leftarrow \infty$ 
20:  Else:
21:    Set  $C[:, j^*] \leftarrow \infty$ 
22:  d. Update Total Cost:
23:   $TotalCost \leftarrow TotalCost + c_{i^* j^*} \cdot x_{i^* j^*}$ 
24: Return  $X, TotalCost$ 

```

4.2. VAM-Enhanced Multi-Discrete Proximal Policy Optimization Algorithm

For the multiple UAVs jamming resource allocation problem, the PPO algorithm offers several advantages. First, it can improve policy performance through multiple updates, enabling better adaptation to the dynamic nature of jamming resource allocation. Second, by introducing a clipping parameter to limit policy update magnitudes, PPO

ensures policy stability [37]. In practical electronic countermeasure applications, frequent adjustments to jamming resource allocations are often required to accommodate varying environments and mission requirements. Therefore, PPO's high stability and strong robustness make it particularly suitable for solving cooperative multi-UAV jamming resource allocation problems in complex dynamic scenarios.

We model the multi-UAV system as an agent system, formulating the cooperative jamming target allocation problem as a Markov decision process (MDP) represented by the tuple $(S_t, A_t, R_t, p(S_{t+1}|S_t, A_t), \gamma)$, where S_t denotes the state space, A_t represents the action space, R_t is the reward function, $p(S_{t+1}|S_t, A_t)$ gives the state transition probability, and $\gamma \in (0,1)$ is the reward discount factor. At time t , the agent takes actions A_t according to its own policy, interacting with the environment to obtain the next state S_{t+1} and immediate rewards R_t . Combined with the jamming target allocation problem, the relevant designs of the key components are as follows.

5. State space S_t : The state incorporates three critical factors influencing jamming allocation: the survivability index matrix of UAVs $C_{N \times M}^{2A}(t)$, constituted by $c_{ij}^{2A}(t)$; the distance threat matrix between targets and UAVs $C_{N \times M}^{2D}(t)$, constituted by $c_{ij}^{2D}(t)$; and the target threat levels vector $C_{1 \times M}^{2E}(t)$, constituted by $c_j^{2E}(t)$. To facilitate stacking of the feature matrices, the vector is broadcast-expanded into an $N \times M$ matrix. Then, we integrate all these parameters into a three-dimensional situational awareness matrix $Sv'_{N \times M \times 3}$, which serves as the state space at time t :

$$S_t = Sv'_{N \times M \times 3}(C_{N \times M}^{2A}(t), C_{N \times M}^{2D}(t), C_{N \times M}^{2E}(t)). \quad (13)$$

6. Action space A_t : With N jammers each having J beams, i.e., total $N \times J$ beams, and M potential targets per beam, the naive action space would be $M^{N \times J}$. Considering that there is such a large space for action, this can be a hindrance to algorithm convergence. To address dimensionality challenges, we propose an enhanced actor network architecture for the PPO algorithm, where the multi-dimensional discrete action space is decomposed into $N \times J$ independent action branches. Each branch corresponds to a dedicated policy head that determines the jamming target selection for individual beams. This design models the problem as a single-agent system with a factored action space, where the agent represents the centralized controller for all UAVs, enabling parallel computation to accelerate both sampling and loss calculation across multiple action dimensions, effectively reducing the action space dimensionality to $J \times N \times M$. The multi-discrete network architecture is shown in Figure 2.

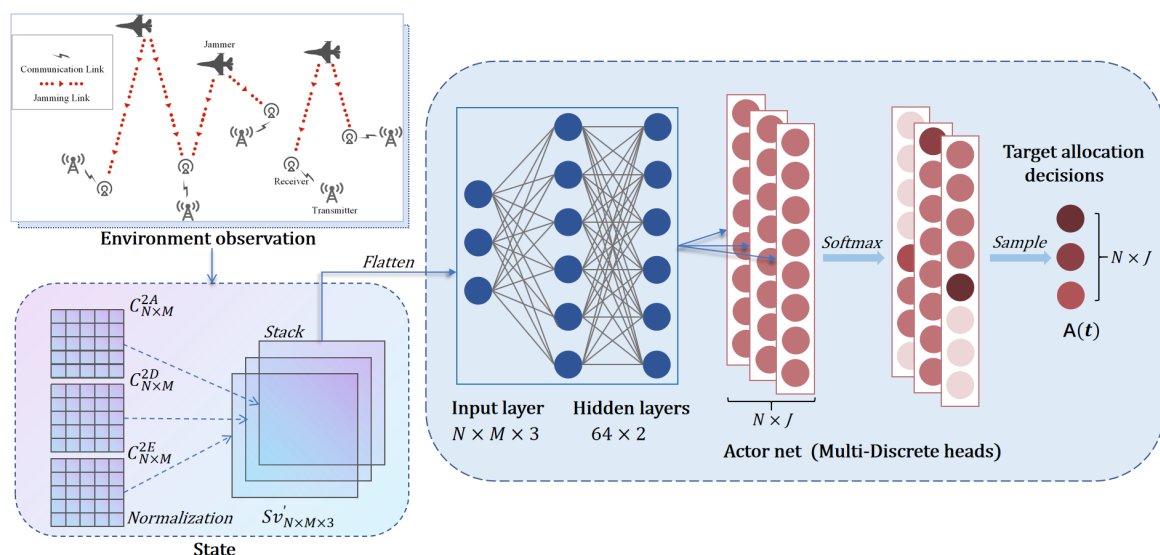


Figure 2. The multi-discrete network architecture for PPO algorithm.

7. Reward function R_t : The reward function should guide the agent to reduce target threats while improving resource allocation efficiency and avoiding local optima. The reward function is divided into three components. The first part is base reward, which reflects the effectiveness of executed jamming operations, incentivizing greater coverage of high-threat targets:

$$R_{bs} = \sum_{i=1}^N \sum_{j=1}^M x_{ij} \cdot c_{ij}. \quad (14)$$

In order to encourage scattered jamming and reduce the waste of resources caused by meaningless repetitive interference, the second part is set to discourage redundant interference by penalizing cases where three or more beams target the same object, with penalties scaling with repetition count:

$$R_{pl} = -\sigma \cdot \sum_{j=1}^M \max \left(\sum_{i=1}^N x_{ij} - 2, 0 \right), \quad (15)$$

where σ is the penalty coefficient. To enhance coverage effectiveness for networked targets, the third part is designed as jamming efficiency bonus:

$$R_{ef} = \delta \cdot \frac{\sum_{j=1}^M \min(\sum_{i=1}^N x_{ij}, 1)}{M}, \quad (16)$$

where δ is the reward coefficient. Therefore, the total reward for each agent action is calculated as follows:

$$R_t = R_{bs} + R_{pl} + R_{ef}. \quad (17)$$

The fundamental objective of RL is to enable agents to learn optimal policies that maximize expected returns without relying on prior knowledge. However, when encountering novel but similar environments, RL agents typically require learning from scratch, and this blind trial-and-error exploration mechanism often leads to prolonged training periods before convergence. Inspired by the approach in [38], where a differential method was employed to initialize Q-Learning, thereby significantly accelerating the training of RL models, we propose integrating VAM during the initial training phase of our PPO algorithm. This hybrid approach, termed the VAM-enhanced multi-discrete proximal policy optimization (VEMD-PPO) algorithm, provides experiential optimization to enhance training directionality. While the prior knowledge may not perfectly align with the new environment, it effectively accelerates the learning of state features. The synergistic combination enables our multi-target jamming system to not only leverage existing human knowledge but also maintain autonomous environmental adaptability, with the detailed algorithmic workflow presented in Algorithm 2.

The pseudocode explicitly incorporates independent policy heads for each action dimension, joint probability calculation for gradient updates, entropy regularization across all action branches, compatibility with generalized advantage estimation (GAE), and mini-batch optimization with multiple epochs. During the first τ training steps, VAM is used solely to generate action decisions for the experience buffer population, without updating neural network gradients. This experience pre-filling provides high-quality initial data for PPO optimization. To address potential distribution shift when control transitions to the agent's policy in remaining steps, we employ PPO's clipping mechanism and entropy regularization. The hybrid buffer sampling (mixing VAM-generated and agent-generated data) ensures smooth policy adaptation. This comprehensive implementation is expected to ensure both computational efficiency and algorithmic robustness in dynamic electronic warfare scenarios.

For practical implementation, the deep learning framework PyTorch is employed with parallelized categorical sampling across all action dimensions. It is crucial to note that, in the multi-branch architecture, the policy $\pi_\theta(s) = [\pi_\theta^1(s), \pi_\theta^2(s), \dots, \pi_\theta^{J*N}(s)]$ is structured such that each $\pi_\theta^i(s)$ generates logits specifically for its corresponding i -th action dimension. The joint probability calculation adheres to the logarithmic formulation, as follows:

$$\log \pi_\theta(a|s) = \sum_{i=1}^{J*N} \log \pi_\theta^i(a^i|s), \quad (18)$$

where θ denotes the parameter vector of the actor network. The joint probability ratio, which quantifies the divergence between updated and previous policies, is computed as follows:

$$r^{\text{total}} = \prod_{i=1}^{J*N} \frac{\pi_{\theta_{\text{new}}}^i(a^i|s)}{\pi_{\theta_{\text{old}}}^i(a^i|s)}. \quad (19)$$

To ensure stable policy updates, we implement the clipped surrogate objective function L_t^{CLIP} , as shown in Equation (20):

$$L_t^{\text{CLIP}} = -\mathbb{E}[\min(r_t^{\text{total}} A_t, \text{clip}(r_t^{\text{total}}, 1 - \epsilon, 1 + \epsilon) A_t)], \quad (20)$$

where A_t represents the GAE, $\text{clip}(\cdot)$ is the projection operator bounding the probability ratio within $[1 - \epsilon, 1 + \epsilon]$, and $\epsilon \in [0, 1]$ controls the update conservativeness.

The GAE computation builds upon the temporal difference (TD) error δ_t , as defined in Equation (21):

$$\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t), \quad (21)$$

with r_t being the immediate reward, $\gamma \in [0, 1]$ the discount factor, and $V_\phi(s_t)$ the state-value function of the critic network parameterized by ϕ . The GAE estimator combines multi-step TD errors through an exponentially weighted sum, as follows:

$$A_t^{\text{GAE}(\gamma, \lambda)} = \sum_{l=0}^{T-t} (\gamma \lambda)^l \delta_{t+l}, \quad (22)$$

where $\lambda \in [0, 1]$ regulates the bias–variance tradeoff in advantage estimation.

Finally, we incorporate entropy regularization to enhance exploration, as follows:

$$H = \sum_{i=1}^N \mathbb{E} \left[- \sum_{a^i} \pi_\theta^i(a^i|s) \log \pi_\theta^i(a^i|s) \right]. \quad (23)$$

This term encourages policy diversity while maintaining training stability, with the entropy coefficient automatically adjusted during optimization.

Algorithm 2: VAM-Enhanced Multi-Discrete Proximal Policy Optimization for Target Assignment

Start environment simulator, generating jammers and targets

Input: Initial policy parameters θ , value function parameters ϕ , clip range ϵ , entropy coefficient β , GAE parameter, discount γ , priori steps τ

1: Initialize: Actor network $\pi_\theta(s)$ with $J * N$ output heads $\{\pi_\theta^i(s)\}_{i=1}^{J*N}$, critic network $V_\phi(s)$, experience buffer D

2: **Collect Trajectories:**

3: For each environment episode:

```

4:   Observe state  $S_t$ 
5:   For each timestep  $t$  until terminal:
6:     a. Multi-discrete action selection:
7:       If  $t < \tau$ :
8:         Choose action  $A_t$  according Algorithm 1
9:       Else:
10:        For each action dimension  $i \in [1, J * N]$ :
11:          Logits  $l_t^i = \pi_\theta^i(S_t)$ 
12:          Distribution  $d_t^i = \text{Categorical}(\text{softmax}(l_t^i))$ 
13:          Sample action  $a_t^i \sim d_t^i$ 
14:          Log probability  $\log p_t^i = d_t^i \log \text{prob}(a_t^i)$ 
15:     b. Execute joint action  $A_t = \{a_t^1, \dots, a_t^{J*N}\}$ 
16:     c. Receive reward  $R_t$ , next state  $S_{t+1}$ 
17:     d. Store  $(S_t, A_t, R_t, \{\log p_t^i\})$  in  $D$ 
18:   Compute advantages  $A_t^{\text{GAE}(\gamma, \lambda)}$  according to Equation (21)
19: Policy Optimization:
20:   For  $K$  epochs:
21:     a. Shuffle mini-batches from  $D$ 
22:     b. For each batch  $(S_t, A_t, R_t, \{\log p_{old,t}^i\})$ :
23:       Recompute Policy:
24:       For each action dimension  $i$ :
25:         Logits  $l_t^i = \pi_\theta^i(S_t)$  , new distribution  $d_{new,t}^i =$ 
            $\text{Categorical}(\text{softmax}(l_t^i))$ 
26:         New log probability  $\log p_{new,t}^i = d_{new,t}^i \log \text{prob}(a_t^i)$ 
27:         Ratio  $r_t^i = \exp(\log p_{new,t}^i - \log p_{old,t}^i)$ 
28:         Compute joint policy loss according to Equation (20)
29:         Compute entropy regularization according to Equation (23)
30:       Value Loss:  $L_t^{\text{VF}} = (V_\phi(S_t) - V_t^{\text{target}})^2$ 
31:       Total Loss:  $L_t^{\text{total}} = L_t^{\text{CLIP}} + c_1 L_t^{\text{VF}} - c_2 H_t$ 
32:       c. Update  $\theta, \phi$  via gradient descent on  $L_t^{\text{total}}$ 
33:   Output: Optimized policy  $\pi_\theta(S)$ 

```

8. It needs to be clarified that the proposed VEMD-PPO algorithm shares similarities with demonstration-based RL and offline RL warm-start approaches in that it incorporates prior knowledge to accelerate training. However, it differs from these paradigms in several key aspects. Unlike behavior cloning [39], which typically learns a policy directly from demonstration data without online interaction, VEMD-PPO uses VAM-generated initial allocations only as a starting point to guide exploration during the early training phase, while still relying primarily on online policy optimization through environmental interactions. This avoids behavior cloning's limitation of potentially failing when demonstration data are suboptimal or the environment dynamics change significantly. In contrast to demonstration-based RL methods that often depend on human demonstrations [29], VEMD-PPO leverages algorithmic

heuristics to generate demonstrations automatically, making it more scalable and less dependent on costly human data collection. Furthermore, unlike offline RL warm-start approaches [40] that pre-train policies on fixed datasets before fine-tuning, VEMD-PPO operates in a fully online setting where the initial VAM-based policy is continuously improved through real-time interactions, ensuring better adaptability to dynamic electronic warfare environments. The VAM integration serves as an experience replay mechanism rather than a policy cloning target, preserving the exploration–exploitation balance essential for discovering optimal strategies in unknown states.

5. Experiments

5.1. Parameter Setting

In this section, we present and analyze the simulation results of our proposed VEMD-PPO algorithm. The simulation experiments were conducted using the following computer hardware and software: an Intel(R) Xeon(R) Gold 6230R CPU manufactured by Intel Corporation and sourced from Hefei, China, 128 GB RAM, an NVIDIA GeForce RTX 3090 GPU manufactured by PC Partner and sourced from Dongguan, China, Windows 10 Professional, Python 3.7.12, and Pytorch 1.12.0. To ensure statistical reliability, all experiments were conducted with five different random seeds, and results are presented as mean $\pm$ standard deviation over these independent runs.

The electronic countermeasure scenario consists of a communication network with $M = 5$ transmitter–receiver pairs being targeted by $N = 3$ jamming UAVs. We consider a three-dimensional engagement geometry where (1) the horizontal positions of target transmitters and our UAVs are randomly distributed within a $2 \text{ km} \times 2 \text{ km}$ rectangular area; (2) receiver positions are randomly generated within an annular region centered at each transmitter, with inner and outer radii of 0.3 km and 1 km respectively; and (3) while all transmitters and receivers maintain zero altitude, the UAVs' flight altitudes follow a normal distribution ($\mu = 1 \text{ km}$, $\sigma = 0.1 \text{ km}$). The threat coefficients for all five communication links are obtained through electronic reconnaissance. Table 1 summarizes the remaining environmental parameter configuration [41–43], and the selection of parameters penalty coefficient σ and reward coefficient δ will be explained in the subsequent analysis.

Table 1. The environmental parameter configuration.

Parameter	Value
Jamming beams per UAV	3
Maximum simultaneous jamming beams per target	2
Threat level of target	$N(0, 1)$
Jamming power/dBm	50
Attack probability	0.1
Damage probability	1
Number of target attacks	[1, 3]
Penalty coefficient	0.1
Reward coefficient	0.5

The neural network architecture of our proposed algorithm consists of fully connected hidden layers for both actor and critic networks, with each network containing two hidden layers of 64 neurons and ReLU activation functions. Specifically, the actor network employs Tanh activation in its output layer, while the Adam optimizer is utilized for neural network weight updates. To enhance training stability, we implement orthogonal initialization and apply normalization to both advantage functions and reward functions, thereby mitigating potential instabilities caused by magnitude variations in the data. To

further expedite the training process, VAM is employed for decision-making during the first 10% of training steps in each episode, with the resulting experiences stored in the replay buffer. Detailed parameter configurations are provided in Table 2.

To demonstrate the superiority of the proposed algorithm relative to existing approaches, we establish three baseline algorithms for comparison: standard PPO, modified VAM, and DBO-QL [26], which also combines the meta-heuristic method for action space exploration with RL for decision optimization. These comparative benchmarks are specifically designed to validate the effectiveness of both the introduced multi-discrete network architecture and VAM-enhanced mechanism while ensuring rigorous experimental control. All comparative benchmarks maintain equivalent computational conditions and environmental configurations. The PPO baseline maintains identical network architecture and hyperparameter configurations with our VEMD-PPO approach, including orthogonal initialization of network weights, and reward normalization procedures, while the Vogel benchmark is implemented as a sequential decision-making process that receives identical input states from the RL environment's situational awareness matrix and employs reward calculation mechanisms consistent with the RL environment's evaluation framework, which is also applied to the DBO-QL algorithm. This standardized implementation ensures the exclusion of confounding factors to the greatest extent possible, thereby enabling direct attribution of performance differences to the algorithmic enhancements rather than implementation variances, following established best practices for rigorous RL algorithm benchmarking.

Table 2. The simulation parameter configuration.

Parameter	Value
Episodes	1000
Steps per episode	100
Learning rate of actor network	3×10^{-4}
Learning rate of critic network	3×10^{-4}
Truncation factor	0.2
Discount factor	0.99
Replay buffer	10^6
Batch size	2048
GAE parameter	0.95
Entropy coefficient	0.01
Soft update coefficient	0.005

5.2. Performance Evaluation

5.2.1. Effectiveness Analysis of the VEMD-PPO Algorithm

Figure 3 illustrates the variation of average episode reward across 1000 training episodes for four algorithms, demonstrating the convergence characteristics of our proposed VEMD-PPO approach. The lines in the figure represent the mean cumulative reward across five different random seeds, and the shaded regions indicate the corresponding standard deviations. The results show that VEMD-PPO achieves progressively higher cumulative rewards with increasing iterations, confirming its effectiveness in performance enhancement. While the reward values begin to stabilize around 400 episodes, minor fluctuations persist due to the dynamic operational environment where rapidly changing situational conditions of networked targets cause variations in jamming effectiveness. Under identical conditions, the baseline PPO algorithm exhibits slower convergence around 700 episodes with significantly inferior final reward values compared with our method. Although VAM initially achieves relatively high rewards, its inherent inability to learn environmental dynamics prevents it from reaching the optimal performance level obtained by

our learning-based approach. Furthermore, the DBO-QL algorithm exhibits the lowest reward profile among all methods. It starts with a lower initial reward and shows minimal improvement throughout the training process, ultimately converging to a significantly suboptimal performance level. This suggests that, while the DBO component aids in exploration, the Q-Learning framework struggles to effectively determine a policy in this high-dimensional, multi-discrete action space compared with policy gradient methods like PPO. These comparisons provide conclusive evidence that VEMD-PPO offers substantial improvements over baseline methods in both accelerating training progress and enhancing final performance outcomes.

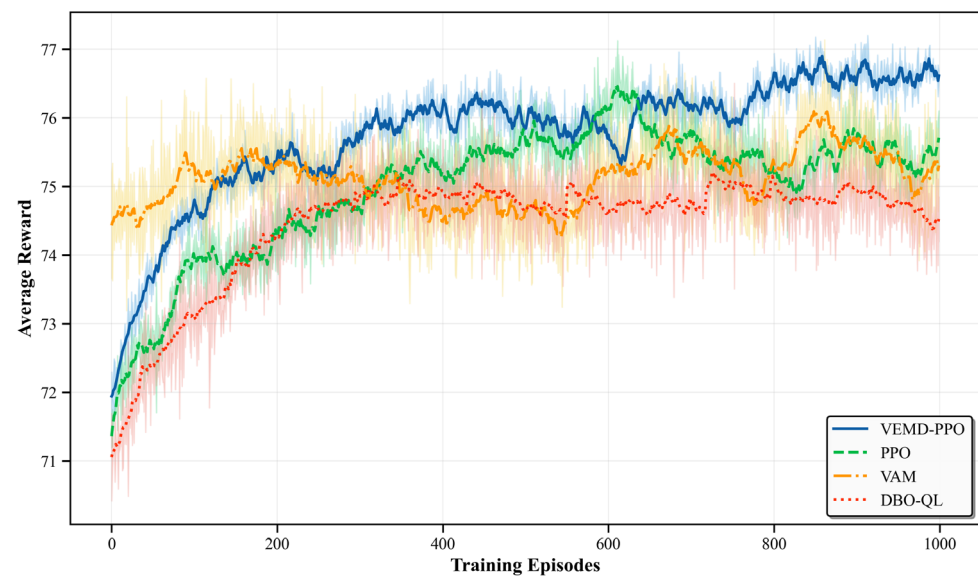


Figure 3. Comparison of the average reward values of four algorithms.

To further evaluate the stability and convergence quality of the algorithms, Figure 4 presents the box plots of the average reward over the final 100 training episodes for VEMD-PPO, PPO, VAM, and DBO-QL. The statistical analysis reveals a clear performance hierarchy. The proposed algorithm achieves the highest and most stable final performance, with a mean reward of 76.616 and the smallest standard deviation 0.104 among all methods. Its interquartile range (IQR) is narrowest (76.536 to 76.685), and it contains no outliers, indicating consistent and reliable policy execution upon convergence. The PPO algorithm ranks second with a mean of 75.468 and a slightly higher variance (std: 0.159). The VAM method shows the largest fluctuation (std: 0.238, mean: 75.327), reflecting its inherent limitation in adapting to dynamic scenarios without learning capability. Notably, the DBO-QL algorithm, included as an additional RL integrated with meta-heuristic, delivers the lowest average reward (74.739) and demonstrates a constrained reward range, suggesting limited effectiveness in this specific high-dimensional action space. Compared with the early stages of training, VEMD-PPO achieved a 13.86% greater increase in reward than PPO and a 38.45% greater increase than DBO-QL. The superior and stable performance of VEMD-PPO in this late-training phase statistically validates that the integration of VAM-based pre-training effectively guides the PPO agent towards a higher-performing and more robust policy basin compared with standard PPO and other benchmarks.

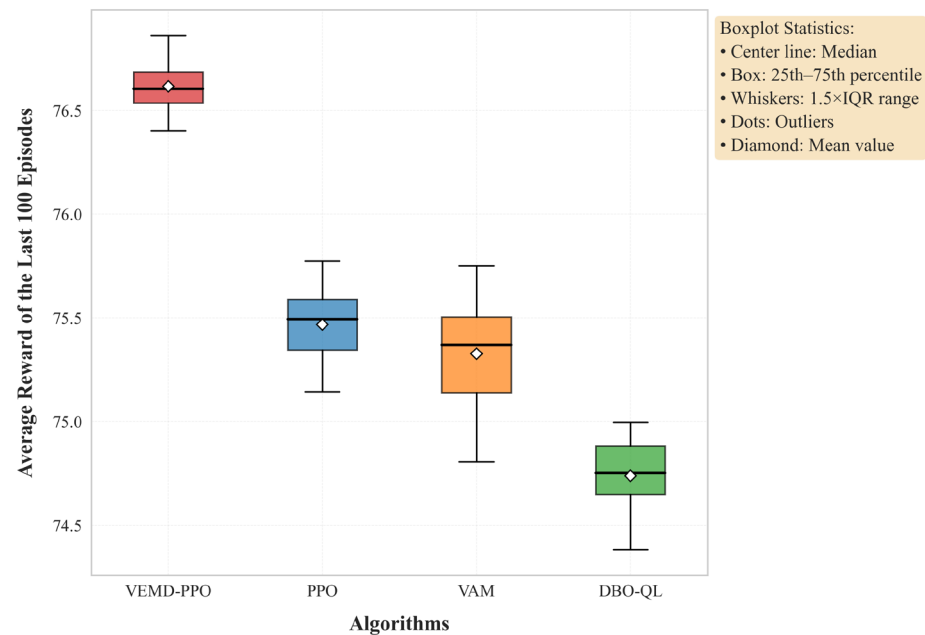


Figure 4. Box plots of rewards over the final 100 episodes for the four algorithms.

5.2.2. Analysis of Decision-Making Speed

In electronic warfare scenarios, where the timeliness of mission execution is critically important and decision speed directly impacts the probability of success, Figure 5 presents the computational time required for single-step decision-making by VEMD-PPO, PPO, VAM and DBO-QL as the scale of both jamming and communication parties increases, with the comparison based on average values obtained from 1000 decision-making trials conducted by each algorithm under three asymmetric engagement scales: 3 jammers interfering with 5 targets, 4 jammers with 7 targets, and 5 jammers with 9 targets.

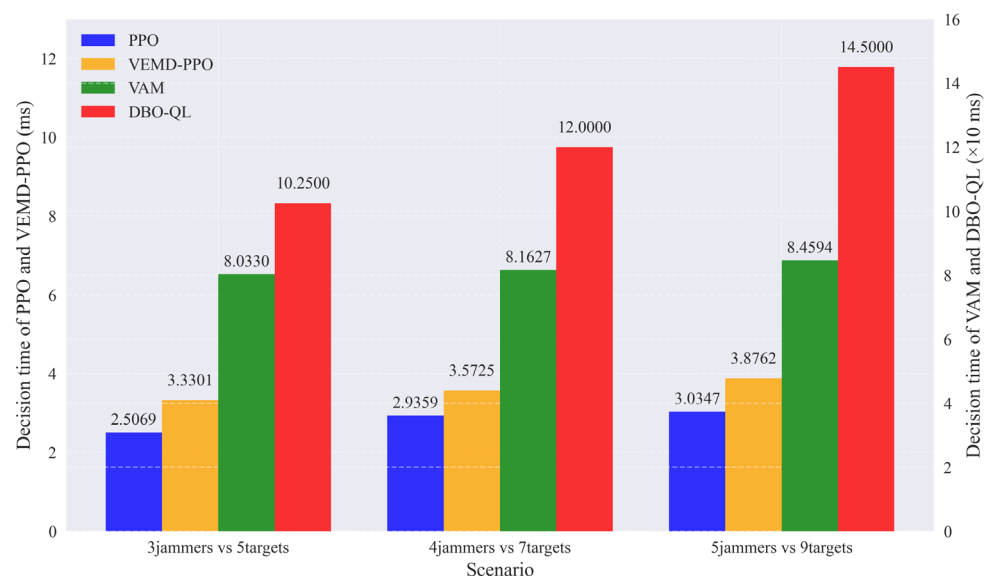


Figure 5. Comparison of the computational time required for single-step decision-making by four algorithms as the countermeasures scale increases.

The results demonstrate that, while all four algorithms exhibit increasing decision times with expanding operational scales, both DRL algorithms maintain consistently lower computational latency (measured in milliseconds) compared with the heuristic and RL combined meta-heuristic baselines. VAM requires tens of milliseconds per decision

due to its computationally intensive iterative calculation process, while DBO-QL incurs the highest decision latency across all tested scenarios, primarily because it combines the search process of the DBO with the tabular update mechanism of Q-Learning, resulting in computational overhead in dynamic environments that is substantially greater than that of the policy-gradient-based algorithm. This further proves the advantage of DRL algorithms' self-learning adaptability in dynamic environments for sequential decision-making—once the policy is successfully learned, near-optimal decisions can be made within extremely short timeframes. Although the proposed VEMD-PPO shows marginally higher latency (within 1 ms) than the baseline PPO algorithm due to the increased complexity introduced by the embedded heuristic method, we consider this an acceptable trade-off given VEMD-PPO's superior learning performance and faster convergence time in practical applications.

5.2.3. Ablation Experiment and Generalization Analysis

Figure 6 presents the ablation study comparing the training performance of the full VEMD-PPO algorithm against its key variants: the standard PPO baseline, VE-PPO (PPO with only the Vogel's approximation pre-training), and MD-PPO (PPO with only the multi-discrete action head architecture), as shown in Table 3. The purpose is to isolate and evaluate the individual contribution of each proposed component.

Table 3. Ablation experiment settings. “•” represents instances where the mechanism is included, and “✕” represents instances that do not include the mechanism.

Algorithm	VAM-Enhanced Mechanism	Multi-Discrete Architecture
VEMD-PPO	•	•
VE-PPO	•	✕
MD-PPO	✕	•
PPO	✕	✕

The results clearly demonstrate the synergistic effect of integrating both innovations. The full VEMD-PPO algorithm achieves the fastest learning speed and the most stable ascent to the highest reward level, indicating superior sample efficiency and convergence stability. The VE-PPO variant shows a noticeable improvement in early-stage learning compared with standard PPO, demonstrating that the VAM-based pre-training effectively provides a high-quality initial policy, accelerating the initial phase. Conversely, the MD-PPO variant, which structures the action space without heuristic pre-training, also outperforms standard PPO but converges slower than VEMD-PPO. This confirms that the multi-discrete architecture itself enhances the PPO's capability to handle the high-dimensional action space. Crucially, the fact that VEMD-PPO surpasses both VE-PPO and MD-PPO throughout the training process indicates that the VAM pre-training and the multi-discrete architecture are complementary. The pre-training offers a strategic starting point, while the specialized network architecture enables efficient learning and refinement of the complex jamming policy, leading to the best overall performance.

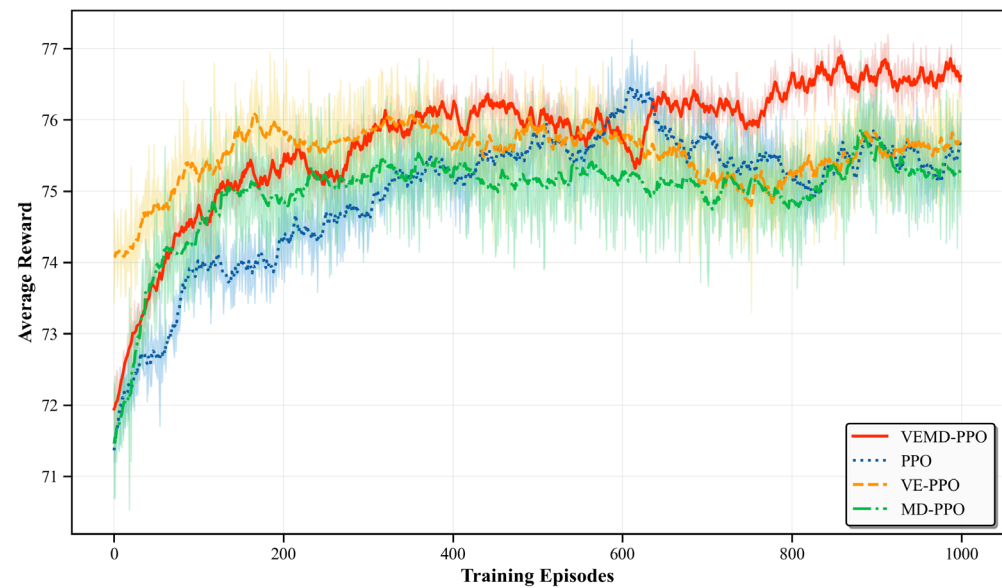


Figure 6. Ablation experiment of VEMD-PPO.

Although the VAM-enhanced PPO framework has been validated in cooperative electronic jamming scenarios, its underlying methodology possesses significant generalization potential for broader many-to-many resource allocation problems. The VAM component, rooted in classical transportation problems, is inherently applicable to any domain requiring efficient resource distribution between multiple suppliers and consumers under capacity constraints—such as logistics scheduling, cloud computing resource allocation, or multi-robot task assignment. Meanwhile, the PPO algorithm’s robustness in handling high-dimensional state–action spaces enables it to adapt to various dynamic environments. The multi-discrete action head design further allows scalable decomposition of complex allocation actions into parallel sub-policies. Therefore, with the appropriate reformulation of state representations (e.g., replacing situational matrices with domain-specific features like packet delay in communication networks or energy cost in sensor networks) and reward functions, the proposed architecture can be extended to other real-time decision-making scenarios involving multi-resource to multi-task assignments, provided the problem can be formulated as a sequential decision-making process under constraints.

5.2.4. Parameter Sensitivity Experiments

Figure 7 presents a sensitivity analysis investigating the impact of the penalty coefficient σ and the reward coefficient δ on the final performance of the VEMD-PPO algorithm. These coefficients, defined in Equations (15) and (16), balance the trade-off between encouraging effective jamming and penalizing constraint violations. We set the coefficient ranges for each reward according to their magnitudes to ensure the dominance of the base reward. The boxplot statistics of the average reward from the final 100 training episodes for four parameter combinations reveal a clear trend: the combination ($\sigma = 0.1, \delta = 0.5$) yields the highest mean performance 76.616, and the smallest standard deviation 0.104, indicating that, under this configuration, the algorithm not only attains the best performance but also converges most stably. The combination ($\sigma = 0.1, \delta = 1$) performs markedly worse than the other three. This may be because the excessively large reward coefficient amplifies fluctuations in the coverage reward, which disrupts stable policy optimization.

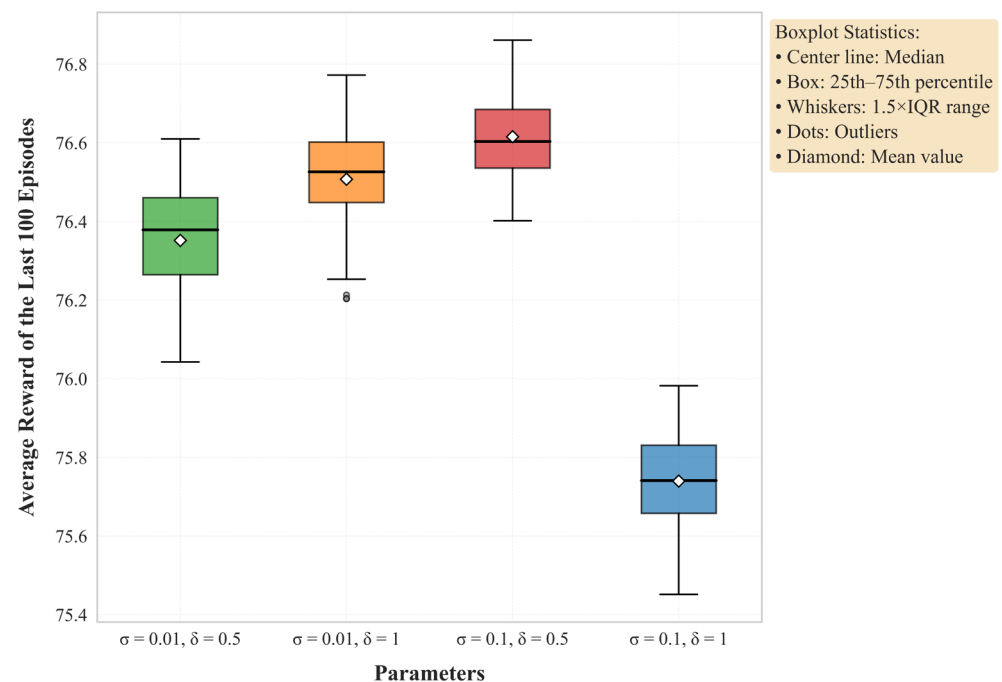


Figure 7. Parameter sensitivity experiments of penalty coefficient and reward coefficient.

Based on this analysis and practical experience, the selection of $\sigma = 0.1$ and $\delta = 0.5$ for our main experiments seems to be the best choice. Selecting $\sigma = 0.1$ rather than 0.01 imposes a stronger penalty signal on the constraints set by Equation (3). This helps ensure that the trained policy more reliably satisfies the hard constraints, which is critical for task safety and resource efficiency. The ratio $\frac{\delta}{\sigma} = 5$ establishes a practical scale where the positive reward for covering targets and jamming effectiveness meaningfully outweighs the penalty for minor constraint violations, guiding the agent towards the primary objective without being overly discouraged by initial exploratory failures—a common principle in reward shaping. Furthermore, the chosen values ensure that the magnitudes of the two reward terms are comparable, preventing numerical instability during gradient updates. Therefore, the choice ($\sigma = 0.1, \delta = 0.5$) represents a balanced and efficient parameter configuration determined after comprehensively weighing the reward magnitude, final performance, learning stability, and the algorithm’s exploration guidance.

5.2.5. Analysis of Scalability

To investigate whether the designed reward mechanism effectively guides both resource utilization and target coverage, we examine the jamming target allocation decisions made by the proposed RL model in larger-scale three-dimensional electronic warfare scenarios. The metrics target coverage is defined as the percentage of targets that receive at least one jamming beam, expressed mathematically as follows:

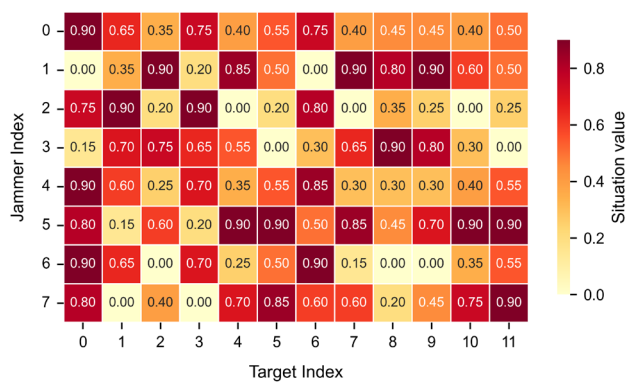
$$\frac{\sum_{j=1}^M \mathbb{I}\left(\sum_{i=1}^N x_{ij} > 0\right)}{M} \times 100\% \quad (24)$$

The metrics resource utilization refers to the percentage of utilized jamming beams, calculated using the following formula:

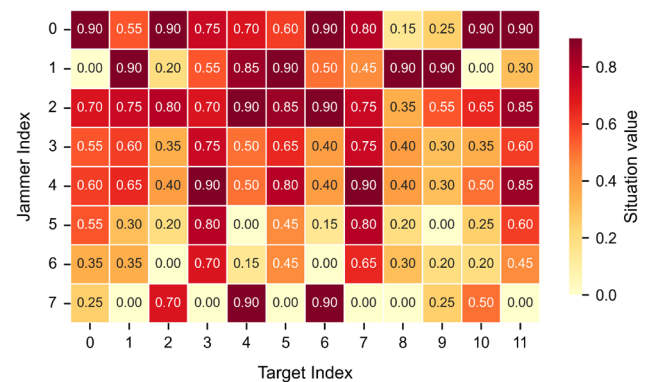
$$\frac{\sum_{i=1}^N \sum_{j=1}^M x_{ij}}{N \times J} \times 100\% \quad (25)$$

Figure 8 present the 3D jamming beam allocation diagrams generated by our algorithm under random different battlefield situational assessments when 8 jamming UAVs engage 12 networked targets, where blue arrows indicate single-beam jamming assignments, red arrows represent dual-beam allocations and black arrows indicate tri-beam jamming assignments.

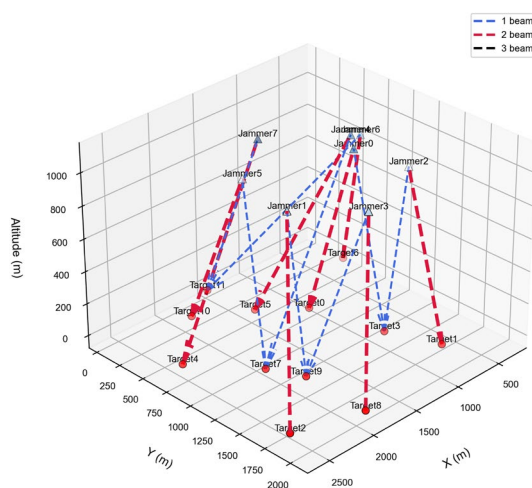
The beam distribution patterns demonstrate that all three beams of each jammer are optimally allocated based on comprehensive factors including geographical positioning and threat coefficients, achieving both 100% resource utilization and 100% target coverage without any instances of over-jamming, defined as three or more beams simultaneously interfering with a single target. These experimental results confirm that the trained RL model, properly guided by the reward mechanism, fully leverages available jamming resources to maximize operational effectiveness while maintaining complete target coverage, successfully avoiding both resource waste from excessive jamming and idle beam conditions, thereby optimizing the operational efficiency of multi-UAV cooperative jamming against multiple targets.



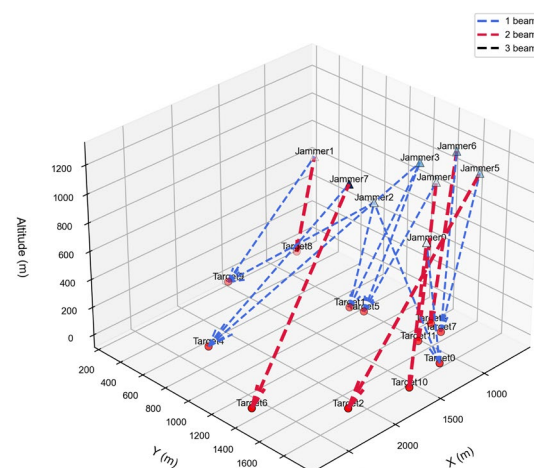
(a1)



(b1)



(a2)



(b2)

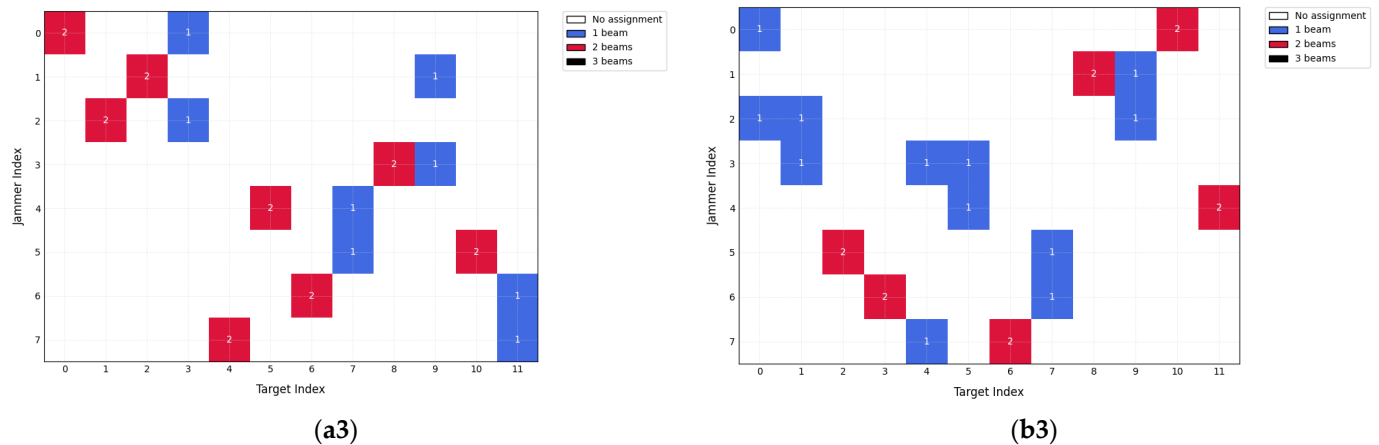


Figure 8. Two randomly selected frames of target allocation status during dynamic jamming of 12 targets by 8 UAVs: (a1,b1) Heatmap of environmental situational awareness for all current jamming targets; (a2,b2) 3D beam allocation diagram of UAVs under current battlefield conditions; (a3,b3) Statistical distribution of UAV beam allocation quantities.

It is meaningful to discuss the memory usage and training scalability of the proposed VEMD-PPO algorithm as the number of UAVs and targets increases, which is critical for practical deployment in large-scale scenarios. The memory consumption primarily stems from the neural network parameters and experience replay buffer. The actor and critic networks each employ two hidden layers with 64 neurons, resulting in a parameter count that scales linearly with the input dimension $3 \times N \times M$, derived from the flattened situational tensor, and the output dimensions $J \times N \times M$ for the multi-discrete action heads. Specifically, the total number of parameters is $O(N \times M)$, indicating that memory usage grows polynomially with the problem size. For example, doubling N and M would approximately quadruple the parameter memory footprint, but modern hardware (e.g., GPUs with sufficient VRAM) can accommodate such increases for moderate-scale problems as tested in our setup.

Regarding training scalability, the computational complexity per training step is dominated by forward/backward passes through the networks, scaling with $O(N \times M)$ due to matrix operations. Although larger N and M may prolong training convergence due to expanded state–action spaces, our multi-discrete architecture reduces the action space dimensionality, mitigating the curse of dimensionality. However, extreme scales (e.g., $N \geq 10, M \geq 15$) may necessitate distributed training or model compression techniques. Future work will empirically validate scalability on larger benchmarks and optimize memory efficiency via dynamic buffer sizing or quantization.

6. Conclusions

This paper addresses the challenges of low jamming resource utilization and inadequate coverage of networked targets caused by dynamic battlefield situations in multi-UAV cooperative jamming scenarios, proposing a multi-discrete PPO algorithm enhanced by heuristic optimization experience. By implementing branched output in the actor network, our approach effectively reduces the action space dimensionality while utilizing VAM to guide the model’s learning process. Experimental results demonstrate that the proposed VEMD-PPO algorithm achieves substantial improvements over baseline methods in both decision-making efficiency and training effectiveness, enabling real-time “many-to-many” jamming resource allocation decisions based on dynamic battlefield situations, which maximizes jamming effectiveness through optimal resource utilization while maintaining comprehensive target coverage.

Future research will focus on the joint optimization of target assignment and power allocation in “many-to-many” jamming scenarios, investigating hybrid action space optimization problems that combine discrete target assignment with continuous power control, while evaluating actual interference effects through metrics like jamming-to-signal ratio (JSR) to quantitatively evaluate the actual physical-layer interference effects, thereby refining the RL feedback mechanism and further improving jamming power resource utilization efficiency.

Author Contributions: Conceptualization, X.L.; software, Y.S.; formal analysis, X.L.; investigation, Y.P. and H.L.; resources, J.Y.; writing—original draft, Y.S.; writing—review and editing, X.L. and J.Y.; supervision, J.Y.; project administration, J.Y.; funding acquisition, X.L. and J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, grant number 62201601.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

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

Abbreviations

The following abbreviations are used in this manuscript:

DRL	Deep reinforcement learning
VAM	Vogel’s approximation method
PPO	Proximal policy optimization
UAV	Unmanned aerial vehicles
MP	Mathematical programming
EA	Evolutionary algorithms
WTA	Weapon–target assignment
NSGA	Non-dominated sorting genetic algorithm
NLP	Nonlinear programming
RL	Reinforcement learning
MC	Monte Carlo
DQN	Deep Q-network
DWTA	Dynamic weapon–target assignment
MOABC	Multi-objective artificial bee colony
MO-WTA	Multi-objective weapon–target assignment
MOEA	Multi-objective evolutionary algorithm
KG-RL	Knowledge-guided reinforcement learning
APF	Artificial potential field
KFDQN	Knowledge-guided fuzzy deep Q-network
TL	Transfer learning
TP	Transportation problems
MDP	Markov decision process
VEMD-PPO	VAM-enhanced multi-discrete PPO
GAE	Generalized advantage estimation
TD	Temporal difference

References

1. Fei, C.; Zheng, H. Research on UAV Cluster Control Technology. *J. Proj. Rocket. Missiles Guid.* **2023**, *43*, 45–55. <https://doi.org/10.15892/j.cnki.djzdx.2023.03.007>.
2. Kuai, W. The Development of Autonomous Control for UAV Clusters. *Informatiz. Res.* **2023**, *49*, 1–5+25.
3. Hu, X.; Ma, H.; Ye, Q.; Luo, H. Hierarchical Method of Task Assignment for Multiple Cooperating UAV Teams. *J. Syst. Eng. Electron.* **2015**, *26*, 1000–1009. <https://doi.org/10.1109/JSEE.2015.00109>.

4. Edison, E.; Shima, T. Integrated Task Assignment and Path Optimization for Cooperating Uninhabited Aerial Vehicles Using Genetic Algorithms. *Comput. Oper. Res.* **2011**, *38*, 340–356. <https://doi.org/10.1016/j.cor.2010.06.001>.
5. Lu, Y.; Chen, D.Z. A New Exact Algorithm for the Weapon-Target Assignment Problem. *Omega* **2021**, *98*, 102138. <https://doi.org/10.1016/j.omega.2019.102138>.
6. Ahuja, R.K.; Kuma, A.; Jha, K.C.; Odin, J.B. Exact and Heuristic Algorithms for the Weapon-Target Assignment Problem. *Oper. Res.* **2007**, *55*, 1136–1146. <https://doi.org/10.1287/opre.1070.0440>.
7. Akbel, O.; Kalaycıoğlu, A. A Solution to Dynamic Weapon Assignment Problem Based on Game Theory for Naval Platforms. *Games* **2024**, *15*, 33. <https://doi.org/10.3390/g15050033>.
8. Wang, C.; Chen, L.; Li, W.; Zeng, K. ANFIS-Based Self-Learning Expert System for Weapon Target Assignment Problem. In *Proceedings of the 2021 2nd International Conference on Control, Robotics and Intelligent System, Qingdao, China, 20–22 August 2021*; Association for Computing Machinery: New York, NY, USA, 2021; pp. 100–104.
9. Huang, J.; Li, X.; Yang, Z.; Kong, W.; Zhao, Y.; Zhou, D. A Novel Elitism Co-Evolutionary Algorithm for Antagonistic Weapon-Target Assignment. *IEEE Access* **2021**, *9*, 139668–139684. <https://doi.org/10.1109/ACCESS.2021.3119363>.
10. Wang, D.; Xin, B.; Wang, Y.; Zhang, J.; Deng, F.; Wang, X. Constraint-Feature-Guided Evolutionary Algorithms for Multi-Objective Multi-Stage Weapon-Target Assignment Problems. *J. Syst. Sci. Complex.* **2025**, *38*, 972–979. <https://doi.org/10.1007/s11424-025-4232-2>.
11. Kline, A.G.; Ahner, D.K.; Lunday, B.J. Real-Time Heuristic Algorithms for the Static Weapon Target Assignment Problem. *J. Heuristics* **2019**, *25*, 377–397. <https://doi.org/10.1007/s10732-018-9401-1>.
12. Luo, W.; Lu, J.; Liu, K.; Chen, L. Learning-Based Policy Optimization for Adversarial Missile-Target Assignment. *IEEE Trans. Syst. Man Cybern. Syst.* **2022**, *52*, 4426–4437. <https://doi.org/10.1109/TSMC.2021.3096997>.
13. Volle, K.; Rogers, J.; Brink, K. Decentralized Cooperative Control Methods for the Modified Weapon–Target Assignment Problem. *J. Guid. Control Dyn.* **2016**, *39*, 1934–1948. <https://doi.org/10.2514/1.G001752>.
14. Wang, S.; Sheng, Q.; Sun, M.; Tang, S.; Zhen, Z. Multi-UAV Cooperative Target Assignment Based on Improved NSGA-III Algorithm. *Aero Weapon.* **2024**, *31*, 100–111. <https://doi.org/10.12132/ISSN.1673-5048.2023.0222>.
15. Zhai, J.; Yang, J. An Integrated Cooperative Guidance Design for Target Assignment and Simultaneous Attack on Multiple Targets. *Int. J. Control* **2024**, *97*, 2175–2188. <https://doi.org/10.1080/00207179.2023.2260006>.
16. Jin, T.; Shin, H.-S.; Tsourdos, A.; He, S. Integrated Target Assignment and Trajectory Optimization for Many-to-Many Mid-course Guidance. *IEEE Trans. Aerosp. Electron. Syst.* **2025**, *61*, 853–867. <https://doi.org/10.1109/TAES.2024.3448408>.
17. Kafle, S.; Jagannath, J.; Kane, Z.; Biswas, N.; Kumar, P.S.V.; Jagannath, A. Generalization of Deep Reinforcement Learning for Jammer-Resilient Frequency and Power Allocation. *IEEE Commun. Lett.* **2023**, *27*, 1789–1793. <https://doi.org/10.1109/LCOMM.2023.3274594>.
18. Cheng, Q.; Chen, D.; Gong, J. Weapon-Target Assignment of Ballistic Missiles Based on Q-Learning and Genetic Algorithm. In *Proceedings of the 2021 IEEE International Conference on Unmanned Systems (ICUS), Beijing, China, 15–17 October 2021*; IEEE: Piscataway, NJ, USA, 2021; pp. 908–912.
19. Roux, H.L.; Roodt, J.; Mouton, H. Applying Reinforcement Learning to the Weapon Assignment Problem in Air Defence. *Sci. Mil.* **2011**, *39*, 99–116. <https://doi.org/10.5787/39-2-115>.
20. Meng, F.; Tian, K.; Wu, C. Deep Reinforcement Learning-Based Radar Network Target Assignment. *IEEE Sens. J.* **2021**, *21*, 16315–16327. <https://doi.org/10.1109/JSEN.2021.3074826>.
21. Li, S.; He, X.; Xu, X.; Zhao, T.; Song, C.; Li, J. Weapon-Target Assignment Strategy in Joint Combat Decision-Making Based on Multi-Head Deep Reinforcement Learning. *IEEE Access* **2023**, *11*, 113740–113751. <https://doi.org/10.1109/ACCESS.2023.3324193>.
22. Li, C.; Xin, B.; He, Y.; Wang, D.; Li, Y. Dynamic Weapon Target Assignment Based on Deep Q Network. In *Proceedings of the 2023 42nd Chinese Control Conference (CCC), Tianjin, China, 24–26 July 2023*; IEEE: Piscataway, NJ, USA, 2023; pp. 1773–1778.
23. Wang, T.; Fu, L.; Wei, Z.; Zhou, Y.; Gao, S. Unmanned Ground Weapon Target Assignment Based on Deep Q-Learning Network with an Improved Multi-Objective Artificial Bee Colony Algorithm. *Eng. Appl. Artif. Intell.* **2023**, *117*, 105612. <https://doi.org/10.1016/j.engappai.2022.105612>.
24. Zou, S.; Shi, X.; Song, S. MOEA with Adaptive Operator Based on Reinforcement Learning for Weapon Target Assignment. *Electron. Res. Arch.* **2024**, *32*, 1498–1532. <https://doi.org/10.3934/era.2024069>.
25. Zhao, L.; Feng, Y.; Hawbani, A.; Xu, L.; Liu, Z.; Bi, Y. Optimized Resource Allocation in Vehicle Edge Computing Through Platoon Collaboration. *IEEE Internet Things J.* **2025**, *12*, 16129–16141. <https://doi.org/10.1109/JIOT.2025.3530573>.
26. 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>.

27. Ma, L.; Li, N.; Guo, Y.; Wang, X.; Yang, S.; Huang, M.; Zhang, H. Learning to Optimize: Reference Vector Reinforcement Learning Adaption to Constrained Many-Objective Optimization of Industrial Copper Burdening System. *IEEE Trans. Cybern.* **2022**, *52*, 12698–12711. <https://doi.org/10.1109/TCYB.2021.3086501>.
28. Wang, Y.; Wang, W.; Chen, D. Knowledge-Guided Reinforcement Learning with Artificial Potential Field-Based Demonstrations for Multi-Autonomous Underwater Vehicle Cooperative Hunting. *J. Mar. Sci. Eng.* **2025**, *13*, 423. <https://doi.org/10.3390/jmse13030423>.
29. Jiang, H.; Li, S.; Zhang, J.; Zhu, Y.; Xu, X.; Liu, D. Efficient State Representation with Artificial Potential Fields for Reinforcement Learning. *Complex Intell. Syst.* **2023**, *9*, 4911–4922. <https://doi.org/10.1007/s40747-023-00995-8>.
30. Qin, P.; Zhao, T. Knowledge Guided Fuzzy Deep Reinforcement Learning. *Expert Syst. Appl.* **2025**, *264*, 125823. <https://doi.org/10.1016/j.eswa.2024.125823>.
31. Xue, X.; Mao, G.; Kumari, S.; Liu, Z. Multi-Objective Genetic Programming Assisted Stochastic Deep Reinforcement Learning for Dynamic Knowledge Integration in Transportation Networks. *IEEE Trans. Intell. Transport. Syst.* **2025**, *26*, 1–14. <https://doi.org/10.1109/TITS.2025.3545572>.
32. Jin, Q.; Gao, X.; Guo, Z.; Hou, Z. Autonomous Maneuver Decision of UAV in Air Combat Based on Scenario-Transfer Deep Reinforcement Learning. In *Proceedings of 2021 International Conference on Autonomous Unmanned Systems (ICAUS 2021)*, Changsha, China, 24–26 September 2021; Wu, M., Niu, Y., Gu, M., Cheng, J., Eds.; Lecture Notes in Electrical Engineering; Springer Singapore: Singapore, 2022; Volume 861, pp. 2616–2625.
33. Chen, H.; Zheng, J.; Lao, L. Optimizing the Distribution of Food Based on the Vogel Method. *MATEC Web Conf.* **2022**, *355*, 02039. <https://doi.org/10.1051/mateconf/202235502039>.
34. Kankarofi, R.H.; Ayakubu, U.; Sulaiman, I.M.; Mamat, M.; Sukono; Saputra, M.P.A. Fertilizer Transportation Problem Using Vogel Approximation Method. *IOP Conf. Ser. Mater. Sci. Eng.* **2021**, *1115*, 012005. <https://doi.org/10.1088/1757-899X/1115/1/012005>.
35. Pratihari, J.; Kumar, R.; Edalatpanah, S.A.; Dey, A. Modified Vogel's Approximation Method for Transportation Problem under Uncertain Environment. *Complex Intell. Syst.* **2021**, *7*, 29–40. <https://doi.org/10.1007/s40747-020-00153-4>.
36. Samiei, A.; Sun, L. Distributed Matching-By-Clone Hungarian-Based Algorithm for Task Allocation of Multiagent Systems. *IEEE Trans. Robot.* **2024**, *40*, 851–863. <https://doi.org/10.1109/TRO.2023.3335656>.
37. Liu, Y.; Li, X.; Wang, J.; Wei, F.; Yang, J. Reinforcement-Learning-Based Multi-UAV Cooperative Search for Moving Targets in 3D Scenarios. *Drones* **2024**, *8*, 378. <https://doi.org/10.3390/drones8080378>.
38. Zhang, B.; Zhu, W. A Cognitive Jamming Decision Method for Multi-functional Radar Based on Q-Learning. *Telecommun. Eng.* **2020**, *60*, 129–136. <https://doi.org/10.3969/j.issn.1001-893x.2020.02.001>.
39. Qin, H.; Li, S.; Zhao, D.; Meng, H.; Song, B. Spectrum Resource Allocation for High-Throughput Satellite Communications Based on Behavior Cloning. *J. Commun.* **2024**, *45*, 101–114.
40. Wang, H.; Lin, S.; Zhang, J. Warm-Start Actor-Critic: From Approximation Error to Sub-optimality Gap. In *Proceedings of 40th International Conference on Machine Learning (ICML 2023)*, Honolulu, HI, USA, 23–29 July 2023; PMLR: Cambridge, MA, USA, 2023; pp. 35989–36019.
41. Rao, N.; Xu, H.; Qi, Z.; Song, B.; Shi, Y. Allocation Method of Communication Interference Resource Based on Deep Reinforcement Learning of Maximum Policy Entropy. *J. Northwestern Polytech. Univ.* **2021**, *39*, 1077–1086. <https://doi.org/10.1051/jnwpu/20213951077>.
42. Liu, Y.; Li, X.; Yang, J.; Yang, D.; Wang, J. SANER-PPO Algorithm-Based Jamming Resource Allocation for UAV Swarm. *Control Decis.* **2024**, *39*, 3937–3945. <https://doi.org/10.13195/j.kzyjc.2023.1206>.
43. Peng, X.; Xu, H.; Jiang, L.; Rao, N.; Song, B. A Deep Reinforcement Learning Communication Jamming Resource Allocation Algorithm Fused with Noise Network. *J. Electron. Inf. Technol.* **2023**, *45*, 1043–1054.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.