electronics

MDPI

Article

Bio-Inspired Intelligent Swarm Confrontation Algorithm for a Complex Urban Scenario

He Cai [1,2,3], Yaoguo Luo [1], Huanli Gao [1,2,3,*] and Guangbin Wang [1]

[1] School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, China; caihe@scut.edu.cn (H.C.); 202120117704@mail.scut.edu.cn (Y.L.); 202130462554@mail.scut.edu.cn (G.W.)
[2] Key Laboratory of Autonomous Systems and Networked Control, Ministry of Education, Guangzhou 510640, China
[3] Guangdong Engineering Technology Research Center of Unmanned Aerial Vehicle Systems, Guangzhou 510640, China
* Correspondence: hlgao@scut.edu.cn

Abstract: This paper considers the confrontation problem for two tank swarms of equal size and capability in a complex urban scenario. Based on the Unity platform (2022.3.20f1c1), the confrontation scenario is constructed featuring multiple crossing roads. Through the analysis of a substantial amount of biological data and wildlife videos regarding animal behavioral strategies during confrontations for hunting or food competition, two strategies are been utilized to design a novel bio-inspired intelligent swarm confrontation algorithm. The first one is the "fire concentration" strategy, which assigns a target for each tank in a way that the isolated opponent will be preferentially attacked with concentrated firepower. The second one is the "back and forth maneuver" strategy, which makes the tank tactically retreat after firing in order to avoid being hit when the shell is reloading. Two state-of-the-art swarm confrontation algorithms, namely the reinforcement learning algorithm and the assign nearest algorithm, are chosen as the opponents for the bio-inspired swarm confrontation algorithm proposed in this paper. Data of comprehensive confrontation tests show that the bio-inspired swarm confrontation algorithm has significant advantages over its opponents from the aspects of both win rate and efficiency. Moreover, we discuss how vital algorithm parameters would influence the performance indices.

Keywords: swarm confrontation algorithm; bio-inspired intelligent algorithm; complex urban scenario; electronic game

check for updates

Citation: Cai , H.; Luo, Y.; Gao, H.; Wang, G. Bio-Inspired Intelligent Swarm Confrontation Algorithm for a Complex Urban Scenario. *Electronics* **2024**, *13*, 1848. https://doi.org/10.3390/electronics13101848

Academic Editor: Carlos Tavares Calafate

Received: 20 March 2024
Revised: 7 May 2024
Accepted: 7 May 2024
Published: 9 May 2024

Copyright: © 2024 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) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The research on swarm confrontation with a special focus on the competition and conflict resolution among multiple autonomous entities in complex environments has found applications in both civilian and military domains, such as electronic games [1–3] and unmanned aerial vehicle (UAV) swarm combats [4–7]. The core challenge of swarm confrontation lies in effectively coordinating and managing a group of autonomous entities. This coordination is crucial for optimizing decision-making and action strategies in adversarial environments, as well as for effectively countering the strategies and actions of opponents. Thus far, there have been three major ways to design swarm confrontation algorithms, namely the game theory approach, the evolution computation approach, and the artificial intelligence (AI)-based approach. The game theory method, through simulating interactive strategic games between opposing forces, achieves optimal decision-making and coordination among swarm agents using concepts like Nash equilibrium. The evolutionary computation method, through applying principles of genetic evolution such as mutation, selection, and crossover, achieves the continual adaptation and enhancement of strategies to improve swarm efficiency and adaptability in competitive scenarios. The AI-based method, through employing machine learning algorithms to analyze and predict opponent behaviors, achieves the dynamic and intelligent adjustment of strategies to enhance the overall tactical performance of the swarm.

Besides the aforementioned results, recently, a rule-based swarm confrontation algorithm was proposed in [8], employing the assign nearest tactic. Two scenarios were taken into consideration for the drone swarm confrontation problem. It concludes that the assign nearest rule, though simple, outperforms various swarm confrontation algorithms. This paper also proposes a rule-based swarm confrontation algorithm for the confrontation between two tank swarms of equal size and capability in a complex urban scenario. The most typical characteristic of an urban scenario is the interlocking of roads, and the Unity platform is utilized to construct the confrontation scenario featuring multiple crossing roads. Instead of relying on intuition, we seek answers from nature on how to fight as a swarm. Through the analysis of a substantial amount of biological data and wildlife videos regarding animal behavioral strategies during confrontations for hunting or food competition, two strategies have been utilized to design a novel bio-inspired swarm confrontation algorithm. The first one is the "fire concentration" strategy, which assigns a target for each tank in a way that the isolated opponent will be preferentially attacked with concentrated firepower. The second one is the "back and forth maneuver" strategy, which makes the tank tactically retreat after firing in order to avoid being hit when the shell is reloading. Integrating these two bio-inspired tactics with a guidance system for path planning in the urban environment leads to an entire bio-inspired swarm confrontation algorithm. Two swarm confrontation algorithms, namely the reinforcement learning algorithm [9] and the assign nearest algorithm [8], are chosen as the opponents for the bio-inspired swarm confrontation algorithm proposed in this paper. Data of comprehensive confrontation tests show that the bio-inspired swarm confrontation algorithm has significant advantages over its opponents from the aspects of both win rate and efficiency. Moreover, we discuss how vital algorithm parameters would affect the performance indices.

The main contributions of this paper are threefold:

1. In most of the existing results on swarm confrontation, the environment is an open field with no or a few simple obstacles. Meanwhile, in this paper, the urban scenario featuring multiple crossing roads is considered, which is much more complicated.
2. Different from all the other existing results, the swarm confrontation algorithm proposed in this paper is inspired by the animal behavioral strategies in nature during confrontations for hunting or food competition.
3. Besides the win rate, we have also defined another performance index, i.e., efficiency, to evaluate algorithm performance. Compared to state-of-the-art swarm confrontation algorithms, the bio-inspired swarm confrontation algorithm proposed in this paper exhibits significant advantages over both win rate and efficiency.

2. Related Work

Game theory [10,11] exemplifies a quintessential research methodology in swarm confrontation, which concentrates on the analysis and prediction of adversaries' behavior. Cruz et al. [12] proposed a game-theory-based discrete-time dynamic model that takes into account command and control hierarchies, strategic objectives, and operational constraints. Additionally, they highlighted that in military aerial combat, the highly nonlinear nature of discrete-time dynamic systems complicates the decision-making process. To tackle this challenge, Cruz et al. [13] introduced a method that confines the controllers' computations to a short-term horizon. Nowak [14] developed a complex game-theoretic strategy that includes task allocation for defensive UAVs and the analysis of the best course of action against offensive UAVs. By optimizing the interception efficiency and tactical response of the defensive UAVs when facing multiple intruders, the strategy ensures the security of sensitive areas. Bhattacharya et al. [15] formalized the challenge of developing strategies for UAVs to evade aerial jammer attacks into a zero-sum evasion pursuit game, applying Isaacs' differential game theory to derive control strategies for UAVs. Yao et al. [16] developed a method for multi-aircraft collaborative attacks on multiple targets, enhancing decision accuracy and cooperation efficiency by combining situational assessment, Dempster–Shafer evidence theory, and game theory. Özpala et al. [17] also proposed a multi-UAV air combat

decision-making method for environments with incomplete information. Notably, this study introduced a simplified approach to the mixed Nash equilibrium strategy when involving a large number of agents. Ma et al. [18] introduced the DO-NS algorithm, a hybrid of the double oracle algorithm and neighborhood search methods. This algorithm significantly reduces the search space and enhances computational efficiency, thus more effectively solving mixed-strategy Nash equilibrium problems involving a large number of agents. However, swarm decision-making based on game theory still faces challenges, such as excessive computational complexity with large-scale agents and poor adaptability to dynamic environments.

Evolution computation strategy [19–21] is another traditional research method for swarm confrontation, which primarily employs heuristic algorithms such as the pheromone algorithm,the particle swarm optimization (PSO) algorithm, and the genetic algorithm for path planning and decision-making. Sauter et al. [22] introduced a distributed pheromone algorithm. This algorithm overlays multiple digital pheromone maps in layers, and each individual in the swarm maintains its own digital pheromone map, facilitating the control of unmanned units on the ground and in the air. Foo et al. [23] proposed a solution for three-dimensional path planning problems. This method employs the PSO algorithm for the path planning of UAV swarms, minimizing the risk posed by enemy threats while also reducing fuel consumption. Li et al. [24] improved the Gravitational Search Algorithm (GSA) for UAV swarm path planning, incorporating PSO and social information concepts. This approach effectively mitigates GSA's tendency to fall into local optima in complex global optimization tasks. Duan et al. [25] proposed a hybrid PSO and genetic algorithm to address the formation reconfiguration problem. Later, Duan et al. [26] also applied the PSO method to solve mixed Nash equilibrium problems using game theory, proposing a UAV air combat game theory based on a predator–prey PSO model, which effectively optimizes the air combat mission assignments of UAV swarms. Dolicanin et al. [27] developed an adjusted brainstorm optimization algorithm inspired by human brainstorming processes for path planning in unmanned aerial combat, which demonstrated superior performance compared to eleven other methods in various test scenarios. Li [28] proposed an ant colony optimization-based algorithm for cooperative mission assignment among multiple UAVs tailored to meet the operational requirements of cooperative UAV combat tasks. Ye et al. [29] proposed a modified genetic algorithm with a multi-type gene chromosome encoding strategy. This algorithm, by incorporating a multi-type gene-encoding scheme, generates feasible chromosomes that meet the capabilities, task coupling, and priority constraints of UAVs, which effectively balances the search capability of the algorithm with population diversity. However, evolutionary computation strategies tend to be inefficient in handling large search space problems, potentially leading to local optima in complex environments. They are usually sensitive to parameter selection and are not well suited for precisely fine-tuning solutions.

In recent years, an increasing number of scholars have applied AI methods to swarm confrontation research [30–34]. Compared to traditional game theory and evolutionary computation strategies, AI methods offer advantages in handling complex, high-dimensional environments and adaptively learning optimal strategies from raw observational data. Liu et al. [35] developed a deep-reinforcement-learning-based method for UAV air combat decision-making, utilizing deep neural networks (DQNs) combined with Q-learning to effectively fit action-value functions, thereby reducing the dimensionality issue in complex air combat scenarios. Yang et al. [36] also trained a neural network model based on DQN for UAV air combat decision-making, specifically addressing the dynamic and uncertain maneuvers of enemy aircraft. Lee et al. [37] proposed an innovative autonomous control method for combat UAVs to evade surface-to-air missiles, introducing the amplification of the imitation effect algorithm. By integrating self-imitation learning with random network distillation, this algorithm enhances UAVs' evasion capabilities in complex environments. Zhang et al. [38] introduced a bidirectional recurrent neural network for communication between UAVs and trained a collaborative decision-making model for UAV swarms. This model is capable of integrating task allocation and situational assessment, generating cooperative tactical maneuver policy based on actual combat conditions.

Hu et al. [39] proposed an autonomous maneuver decision-making method for dual UAV cooperative air combat based on specific formation strategies. By incorporating a situational assessment model, a discretized combat state space, and predefined action commands, the effectiveness and convergence speed of model training were enhanced. Nonetheless, AI-based methods still face significant challenges, including large data requirements, limited robustness, and poor generalization capabilities.

3. Problem Description

In this paper, we consider the confrontation of two tank swarms in a complex urban scenario. The confrontation environment is developed by Unity, which is a popular platform for creating 2D and 3D games. The map of the urban scenario is shown in Figure 1, which is 2400 m long and 2000 m wide. The green areas represent impassable zones for both the tanks and the shells fired by the tanks, whereas the white areas allow the traveling of tanks and the flying of shells. The two tank swarms have the same number of tanks, denoted by N, and all the tanks have the same capability. The kinematic equation of the tank is given as follows:

$$\begin{aligned} x_{t+1} &= x_t + v_t \cos(\theta_t)\Delta t \\ y_{t+1} &= y_t + v_t \sin(\theta_t)\Delta t \\ \theta_{t+1} &= \theta_t + \omega_t \Delta t \\ \Theta_{t+1} &= \Theta_t + \Omega_t \Delta t \end{aligned} \tag{1}$$

where (x_t, y_t) and θ_t denote the position and heading angle of the tank at time t, respectively. Θ_t represents the turret angle of the tank. v_t and ω_t are the linear and angular velocities of the tank at time t, respectively, while Ω_t is the angular velocity of the tank's turret. The sampling time is denoted by Δt, set to be $\Delta t = 0.02$ s. v_t, ω_t, and Ω_t are constrained by $v_t \leq v_{\max}$, $\omega_t \leq \omega_{\max}$, and $\Omega_t \leq \Omega_{\max}$, respectively. In this paper, we assume $v_{\max} = 20$ m/s, $\omega_{\max} = 100^\circ/\text{s}$ and $\Omega_{\max} = 100^\circ/\text{s}$. The NavMesh navigation system [40] is adopted in this paper, which calculates v_t and ω_t for the tank according to the navigation path. Similar to [8,26,27,37,41–43], in this paper, it is assumed that the information of the confrontation is complete, i.e., that the tanks can access

1. Complete information about themselves and their teammates, including positions, linear velocities, angular velocities, heading angles, and ammunition load.
2. The current positions of opponents.
3. The map of the environment.

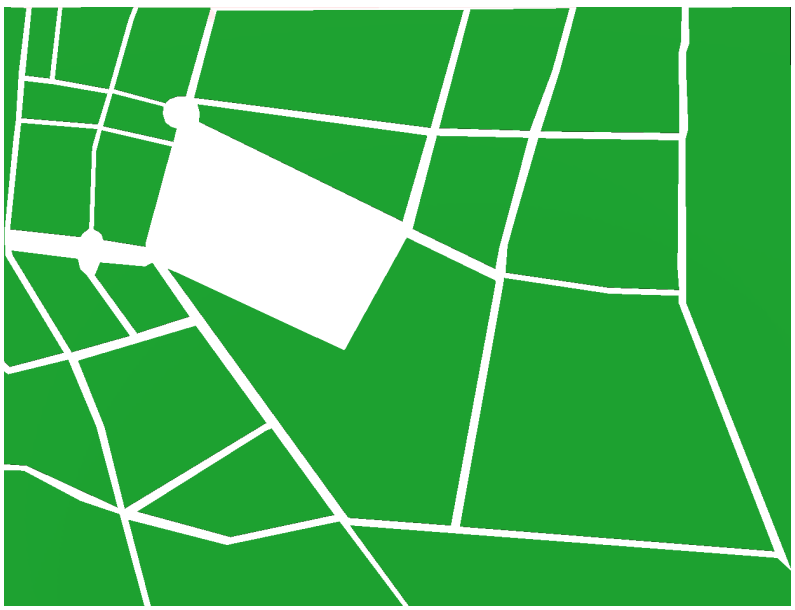


Figure 1. Map of the urban scenario for tank swarm confrontation.

Tanks attack opponents by firing shells, and a tank will be destroyed once it is hit by a shell. The range of the shell is set to be $d_{fire} = 1200$ m, and the shell reloading time is set to be $t_{rld} = 5$ s. In addition to the mobility of the chassis, the cannon of each agent is also capable of rotation. The cannon remains parallel to the xoy plane and is restricted to rotational movements within this plane. Upon acquiring a new target, the agent rotates its cannon at a fixed angular velocity of $\Omega_{\max}$ until the cannon is precisely aligned with the target.

Initially, the two tank swarms, identified, respectively, as the red team and the blue team, are located in different areas of the map. We set the maximal match duration as $t_{end} = 600$ s. A team is called fully destroyed if all its members are destroyed. Either team will win the match if, within t_{end}, the other team is fully destroyed while it is not. Note that the match will end when either one team wins or the time reaches the end. The objective of the swarm confrontation algorithm is to win the match.

The confrontation algorithm is evaluated by two performing indices, namely win rate and efficiency. Over a set of M matches, suppose a team wins M_w matches. Then, the win rate of the algorithm adopted by the team is M_w/M. Moreover, within these M_r winning matches, let n_i denote the number of surviving tanks of the team by the end of the ith match. Then, the efficiency of the algorithm adopted by the team is defined as

$$\psi = \begin{cases} \frac{1}{M_w} \sum_{i=1}^{M_w} \frac{n_i}{N} & M_w > 0, \\ 0 & M_w = 0. \end{cases} \tag{2}$$

The efficiency ψ represents straightforwardly the average survival rate of a team counting in all winning matches. A higher value of ψ implies that the team can win the match at a lower cost, thereby exhibiting better performance.

4. Swarm Confrontation Algorithm Design

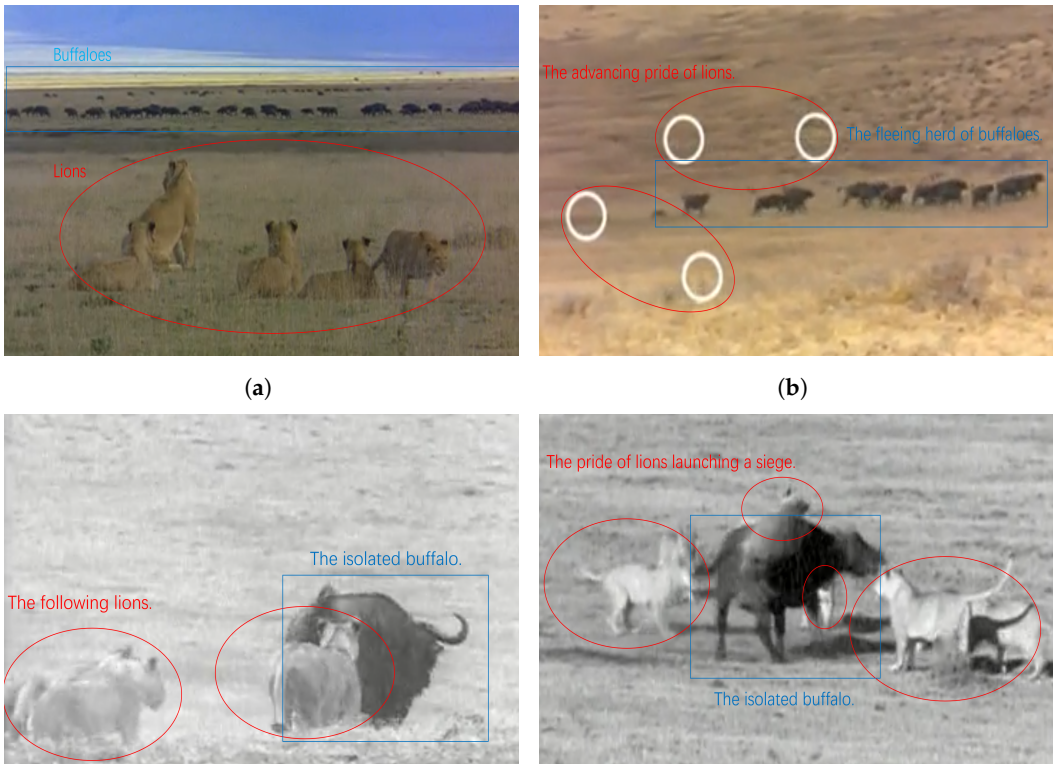
The swarm confrontation algorithm needs to answer two key questions:

1. Given a swarm of opponents, how does one assign a target for each member?
2. When both teams are within each other's firing range, how does one operate during combats?

In this paper, we try to seek the answers to these two questions from nature. Through the analysis of a substantial amount of biological data and wildlife videos, valuable animal behavioral strategies have been identified during confrontations for hunting or food competition. These strategies enable them to form a local advantage in combat power and minimize casualties. By extracting key actions from these animal behaviors and adapting them to the swarm confrontation algorithm, the algorithm's win rate and efficiency are expected to be significantly enhanced.

4.1. Bio-Inspired Rules

We begin with the first question of how to assign a target for each member given a swarm of opponents. Figure 2 shows the confrontation process between a pride of lions and a herd of buffaloes [44]. To be more specific, (a) the lions are observing the distribution and movement of the buffalo herd; (b) the lions tighten the encirclement of an isolated buffalo; (c) the lions approach the isolated buffalo; (d) the lions launch a siege on the isolated buffalo. The priority of the lion pride is to concentrate firepower on isolated individuals under the ever-changing state of confrontation, thereby constantly forming local suppression and advantage. We refer to this tactic of the lions as the "fire concentration" strategy. Given the complex urban scenario considered in this paper, it is often the case that an opponent may find itself in a relatively isolated position. By prioritizing such an opponent as the target with concentrated firepower, the swarm confrontation algorithm would be more effective and efficient.

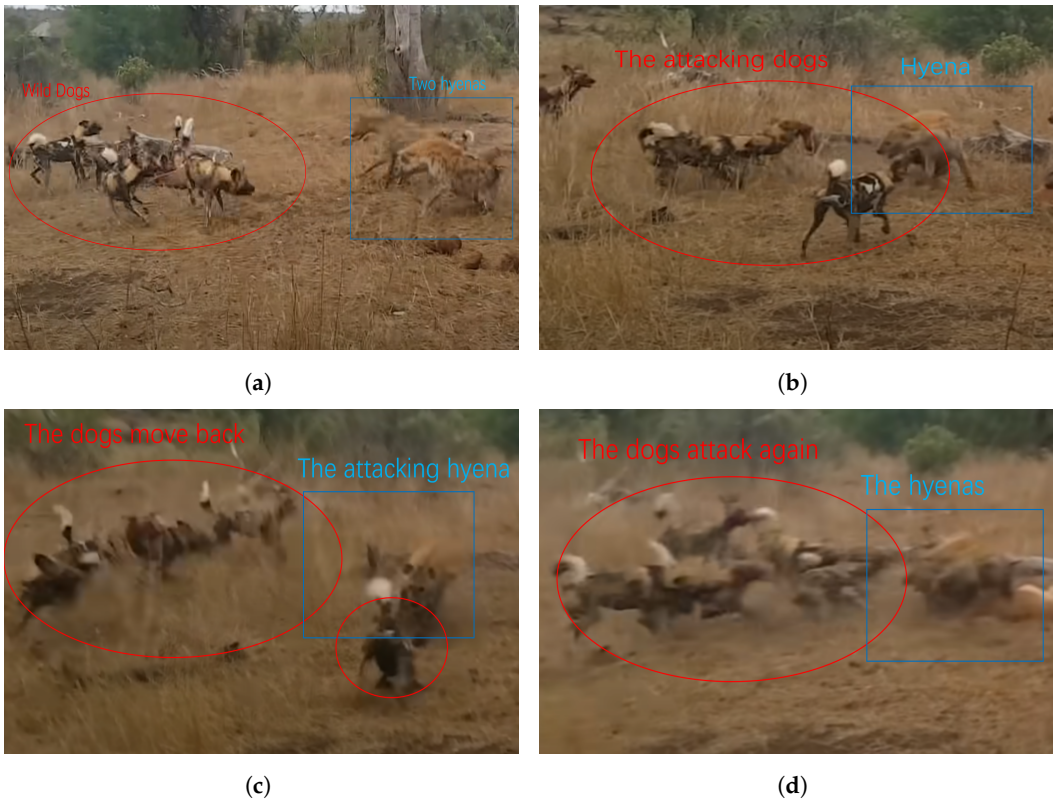


(**a**) (**b**)

(**c**) (**d**)

Figure 2. The "fire concentration" strategy taken from the confrontation between a lion pride and a buffalo herd [44]. (**a**) The lions are observing the distribution and movement of the buffalo herd. (**b**) The lions tighten the encirclement of an isolated buffalo. (**c**) The lions approach the isolated buffalo. (**d**) The lions launch a siege on the isolated buffalo.

Next, we continue with the second question on how to operate during combat when both teams are within each other's firing range. Figure 3 shows the confrontation scenario where a pack of wild dogs and a clan of hyenas compete for food [45]. Specifically, (a) the wild dogs are barking at the hyenas, closely observing their movements; (b) the wild dogs launch an attack when the hyenas are distracted; (c) the wild dogs retreat after attacking to avoid the hyenas' counteroffensive; (d) the wild dogs launch an attack again as the hyenas lower their guard. During confrontations, it is observed that the wild dogs strategically choose to attack only when the hyenas become distracted or lower their guard. Additionally, instead of engaging in direct combat with the advancing hyenas, the wild dogs opt for a rapid withdrawal. They constantly and dynamically adjust their positions, thereby increasing the chances of evading hyenas' counterstrikes and minimizing unnecessary casualties. Furthermore, the wild dogs maintain certain optimal distance from the hyenas, not retreating too far, preparing the next round of attack. We refer to this tactic of the wild dogs as the "back and forth maneuver" strategy. Noting that the tank considered in this paper is subject to the shell reloading time t_{rld}, the chance of being hit for the tank would be greatly lowered if the tank retreats to some safe position after firing, preparing for the next round of firing. By emulating this strategic withdrawal behavior of the wild dogs, the tanks can strike a balance between aggression and self-preservation.



(**a**) (**b**) (**c**) (**d**)

Figure 3. The "back and forth maneuver" strategy taken from the confrontation between a wild dog pack and a hyena pride [45]. (**a**) The wild dogs are barking at the hyenas, closely observing their movements. (**b**) The wild dogs launch an attack when the hyenas are distracted. (**c**) The wild dogs retreat after attacking to avoid the hyenas' counteroffensive. (**d**) The wild dogs launch an attack again as the hyenas lower their guard.

4.2. Navigation System

NavMesh navigation is an AI pathfinding and routing system offered by Unity, derived from improvements to the A* algorithm [46]. This system divides the three-dimensional map into several tile regions, with each tile composed of basic voxels marked with given cost values. This approach simplifies complex spatial path planning into a minimum cost-solving problem within a finite discrete grid, achieving a balance between navigation precision and computational complexity. In this paper, the NavMesh navigation system is employed to facilitate automatic path planning and obstacle avoidance for the tanks, and the baking map of the urban scenario is shown in Figure 4. The cost value for all navigable areas is set to be 1, while non-navigable areas are assigned a cost of 10,000. The navi-distance between two tanks is defined as the length of the path between the two points on the navigation mesh that are closest to these two tanks. To ensure the proper functioning of the navigation system, it is essential to pre-configure the general property settings for the NavMesh navigation system, the navigation component settings for the NavMesh agent, and the property settings for the NavMesh surface components mounted on the map. The specifics of these configurations are detailed in Tables 1–3.

Table 1. General property settings for the NavMesh navigation system.

Property Name	Property Explanation	Property Value
radius (m)	The radius of the agent	2.5
height (m)	The height of the agent	3
step height (m)	The maximum height the agent can climb	0.4

Table 1. *Cont.*

Property Name	Property Explanation	Property Value
maximum slope (°)	The maximum slope the agent can move	1
Drop height (m)	The maximum height an agent can fall to	0.5
Jump distance (m)	The maximum distance the agent can jump	0.5

Table 2. Navigation component settings for the NavMesh agent.

Property Name	Property Explanation	Property Value
speed (m/s)	Movement speed of the agent	1.0
angular speed (°/s)	Rotational speed of the agent	100
acceleration (m/s^2)	Acceleration of the agent	8
stopping distance (m)	Minimum distance to target for stopping	0.5
auto braking	Whether the agent automatically slows down when approaching the target	True
obstacle avoidance radius (m)	The radius used for avoiding obstacles	0.5

Table 3. Property settings for the NavMesh surface components.

Property Name	Property Explanation	Property Value
Layer	The layer covered by the navigation system	Everything
Default Area	The default area type in the map	Walkable
Voxel Size (m)	Edge length of the voxels (squares) for the NavMesh grid	0.833
Tile Size	Number of tiles the NavMesh grid is divided into	256

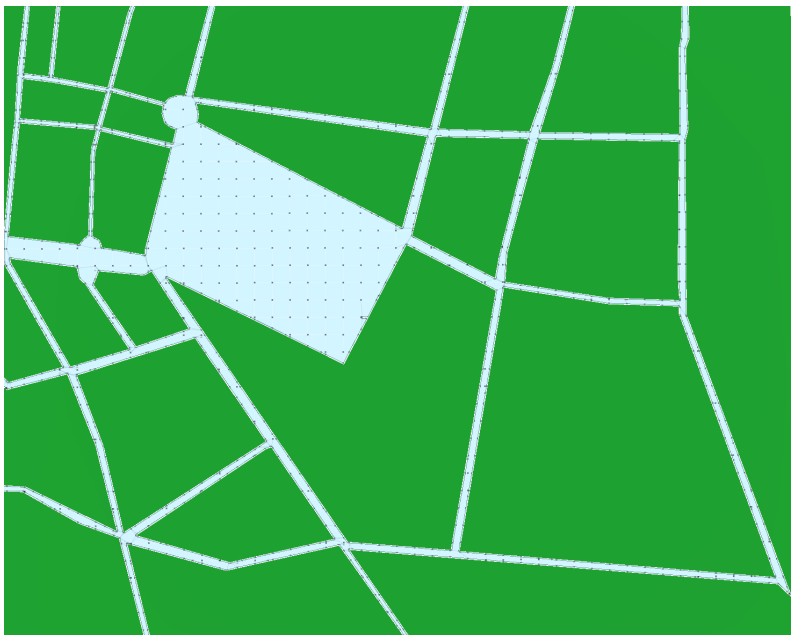


Figure 4. Baking map of the urban scenario.

4.3. Algorithm Design

In what follows, the team adopting the bio-inspired intelligent swarm confrontation algorithm is called the red team, whose members are labeled as $r_1, r_2, \ldots, r_N$, while the opponent team adopting another swarm confrontation algorithm is called the blue team, whose members are labeled as $b_1, b_2, \ldots, b_N$. At step k, for each surviving blue team member b_i, we define $D_{b_i}(k)$ as the sum of the navi-distances from b_i to all other surviving blue team members and $L_{b_i}(k)$ as the sum of the navi-distances from b_i to all surviving red team members. By definition, $D_{b_i}(k)$ measures how far away b_i is from other surviving blue

team members and $L_{b_i}(k)$ measures how far away b_i is from red team members. Moreover, let $D_{\text{avg}}(k)$ be the average of all $D_{b_i}(k)$, and

$$b_i^*(k) = \arg\min_{b_i}\{L_{b_i}(k)\}. \tag{3}$$

By definition, $b_i^*(k)$ refers to the blue team member that is nearest to the red team. If $D_{b_i^*(k)} \geq D_{\text{avg}}(k)$, i.e., the nearest blue team member to the red team happens to be sufficiently far away from other blue team members, then we further define the fire concentration set $S_{fc}(k)$ for the red team as follows. A red team member r_j belongs to $S_{fc}(k)$ if, at step k, it simultaneously satisfies the following three conditions:

1. r_j is surviving;
2. The navi-distance between r_j and $b_i^*(k)$ is less than ℓ;
3. The navi-distance between r_j and the surviving blue team members except $b_i^*(k)$ is more than d_{fire}.

Condition 2 collects red team members that are close to $b_i^*(k)$, and Condition 3 rules out red team members which have more suitable targets.

The bio-inspired intelligent swarm confrontation algorithm is given by Algorithm 1.

Algorithm 1 Bio-inspired intelligent swarm confrontation algorithm

for step k **do**
  if $k = 1$ **then**
    Select t_{sw}^* and d_{back}. Set $t_{back}^* = 300$. Initialize $t_{sw} > t_{sw}^*$ and $t_{back} > t_{back}^*$.
  end if
  for each surviving red team member r_i **do**
    if $t_{sw} \geq t_{sw}^*$ **then**
      if $S_{fc}(k)$ has at least two elements **and** $r_i \in S_{fc}(k)$ **then**
        Let the target of r_i be $b_i^*(k)$.
      else
        Let the target of r_i be b_i which is surviving at step k and has the shortest navi-distance to r_i.
      end if
      Reset t_{sw} to 0.
    else
      $t_{sw} = t_{sw} + 1$.
    end if
    if $t_{back} \geq t_{back}^*$ **then**
      if the shell is ready **and** the navi-distance from r_i to its target is more than d_{fire} **then**
        r_i moves toward the target.
      else if the shell is ready **and** the navi-distance from r_i to its target is less than d_{fire} **then**
        r_i fires at the target.
      else if the shell is not ready **and** the navi-distance from r_i to its target is less than d_{back} **then**
        Let where r_i was 6s ago be the retreat destination for r_i. Reset t_{back} to 0.
      end if
    else
      r_i moves towards the retreat destination. $t_{back} = t_{back} + 1$.
    end if
  end for
end for

In Algorithm 1, t_{sw}^* represents the minimal dwell time for target switching. With the constraint of t_{sw}^*, the tank shall not switch targets too frequently. d_{back} determines the

minimal safety firing distance. t^*_{back} represents the fixed time duration for executing a one-time "back and forth maneuver" strategy with the corresponding timer t_{back} in Algorithm 1. It is set at 300 to accommodate the 250 steps required for reloading shells. Algorithm 1 has two key bio-inspired features. First, the strategy "fire concentration" has higher priority in selecting targets for the red team members. Second, the red team members would retreat instantly after firing, following the "back and forth maneuver" strategy until the shell is ready again. Note that t^*_{sw} and d_{back} are the two key parameters for Algorithm 1.

5. Opponent Swarm Confrontation Algorithms

This section will introduce two benchmark opponent swarm confrontation algorithms for the bio-inspired intelligent swarm confrontation algorithm (abbreviated as the BIO algorithm hereafter), namely the reinforcement learning algorithm (abbreviated as the RL algorithm hereafter) [9] and the assign nearest algorithm (abbreviated as the AN algorithm hereafter) [8]. The RL swarm confrontation algorithm is an AI-based algorithm, trained by the MA-POCA method, while the AN algorithm is a rule-based algorithm, similar to the BIO algorithm proposed in this paper.

5.1. RL Algorithm

The MA-POCA method establishes a novel framework to address the posthumous credit-assignment problem in multi-agent reinforcement learning. It leverages a self-attention mechanism to accommodate dynamically varying numbers of agents, effectively distributing rewards to both active and deceased agents, thereby enhancing performance in dynamic multi-agent environments. Compared to the conventional approaches, the MA-POCA method significantly improves performance in scenarios involving the generation or elimination of agents during swarm confrontation processes. In this work, the MA-POCA method [9] is employed for model training and the RL algorithm is obtained after undergoing 150 million training steps. Note that the outputs of the RL Algorithm include v_t, ω_t and Ω_t for the tanks, as well as a binary parameter deciding whether to fire or not.

The reward and punishment settings for the training process of the MA-POCA method are summarized in Table 4. During the match, there are instant individual rewards for killing opponents and punishments for collisions and accidental injuries to teammates. Moreover, when the match ends, there are individual settlement rewards for winning, both alive or deceased; team settlement rewards for winning, counting also the amount of surviving members n_{svv}; and team settlement punishments for not winning.

Table 4. Reward and punishment settings.

During Match		
Event	**Subject**	**Reward/Punishment**
Hit teammate	Individual	−0.2
Kill opponent	Individual	+0.15
Collision with obstacle	Individual	−0.2
Collision with teammate or enemy	Individual	−0.1
End of Match		
Event	**Subject**	**Reward/Punishment**
Winning while alive	Individual	+0.6
Winning while deceased	Individual	+0.3
Not winning	Team	−0.2
Winning	Team	$0.6 + 0.4 \times (n_{svv}/N)$

Hyperparameters significantly influence the training performance of the reinforcement learning framework. The hyperparameter settings for the training of the RL algorithm is given in Table 5.

Table 5. Hyperparameter settings of the RL algorithm.

Property Name	Property Type	Property Explanation
Batch Size	Number of training examples utilized in one iteration	2048
Buffer Size	Size of the buffer for storing transitions	20,480
Learning Rate	Step size at each iteration while moving toward a minimum of a loss function	0.0003
Beta	The strength of the entropy regularization term in the training strategy.	0.005
Epsilon	The exploration rate that balances between exploring new actions and exploiting known rewards.	0.2
Lambda	The discount factor that determines the present value of future rewards.	0.95
Hidden Units	Number of units in each hidden layer	512
Num Layers	Number of hidden layers	3
Discount Factor	Discount factor used in the reward calculation	0.99
Strength	Parameter to adjust the strength of the reward	1.0

5.2. AN Algorithm

Reference [8] considers the drone swarm confrontation problem for two scenarios. The first scenario is a simple swarm vs. a swarm scene in an open playing field, whereas the second scenario is much more complex, involving ship attack and defense. It was pointed out that among various swarm confrontation algorithms, the algorithm employing the assign nearest strategy exhibits superior performance. The principle of the assign nearest algorithm is simple, i.e., to continuously pair the closest members from the two opposing teams until there are unpaired members remaining. At that point, these surplus members will directly target their nearest opponents even if these opponents have already been assigned as targets. Suppose the blue team members adopt the AN algorithm while the red team members adopt the other opponent swarm confrontation algorithm. The detailed AN algorithm adapted for the complex urban scenario considered in this paper is given by Algorithm 2.

Algorithm 2 Assign nearest swarm confrontation algorithm

for step k **do**
  for each surviving red team member r_i **do**
    Calculate $L_{r_i(k)}$, which is the sum of the navi-distances from r_i to all surviving blue team members.
    In ascending order of $L_{r_i(k)}$, for each surviving red team member r_i, find its nearest and unassigned surviving blue team member, if possible. If such blue team member exists, then the red team member will be assigned as the target for this blue team member.
  end for
  if the blue team has more surviving members than the red team **then**
    for each unassigned surviving blue team member b_i **do**
      Assign the closest red team member as the target for b_i, based on navi-distance.
    end for
  end if
end for

In contrast to the results of [8], two changes have been made to make the AN algorithm suitable for the complex urban scenario. First, the distance between two opposing members is measured by the navi-distance. Second, the AN algorithm also engages in path planning and obstacle avoidance based on the NavMesh navigation system. As a result, the very difference between the BIO algorithm and the AN algorithm lies in the two bio-inspired strategies adopted by the BIO algorithm, namely, "fire concentration" and "back and forth maneuver".

6. Swarm Confrontation Tests and Results Analysis

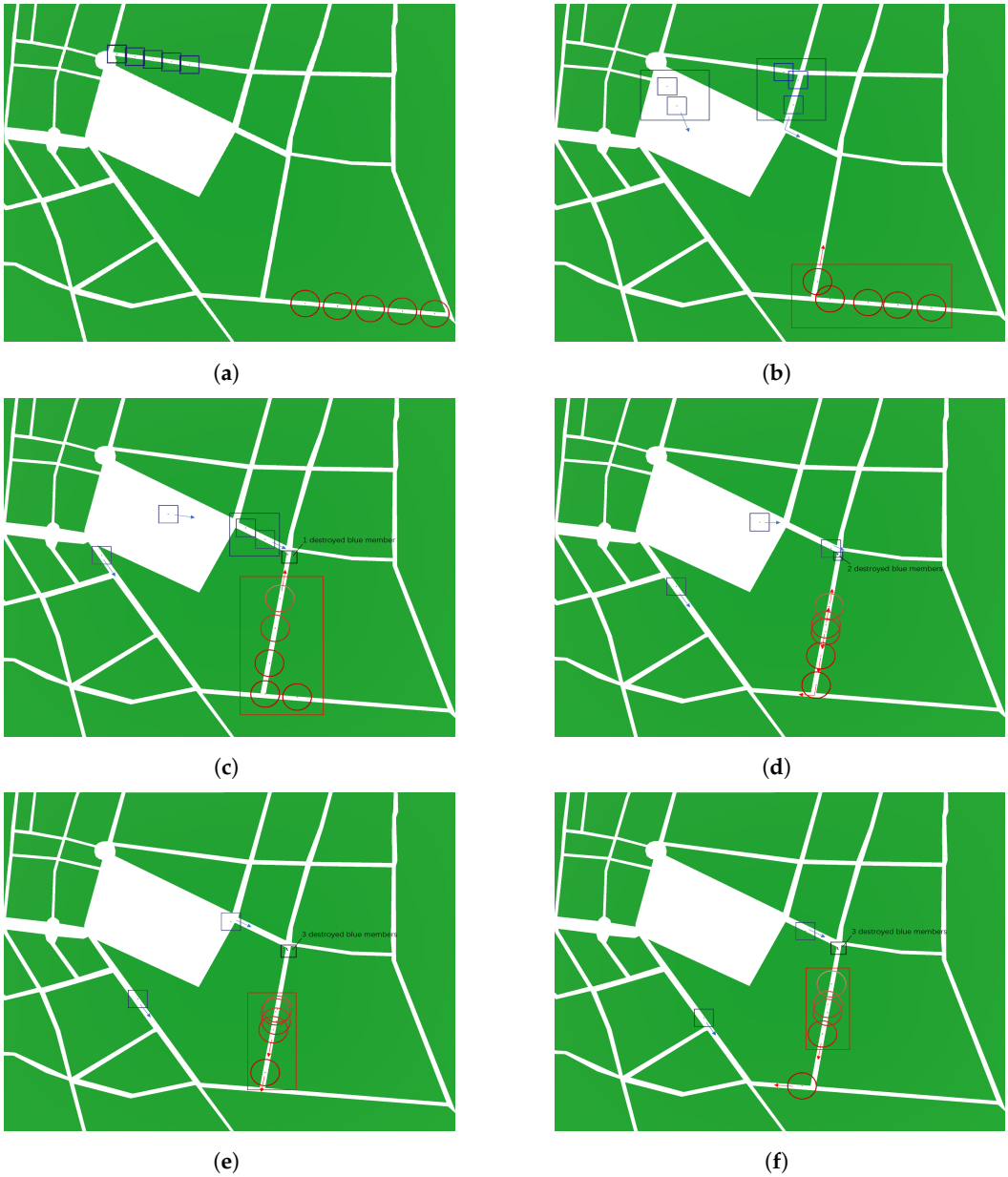
To evaluate the performance of the BIO algorithm, in this section, comprehensive confrontation tests have been conducted between the BIO algorithm and the RL algorithm, as well as between the BIO algorithm and the AN algorithm. In what follows, the analysis of the results for the single match will be first given to obtain an overall impression of the confrontation process. In particular, two typical scenes are presented to illustrate how the bio-inspired "fire concentration" and "back and forth maneuver" strategies work. After that,

the analysis of the results for the group matches is shown, and the confrontation algorithms involved are evaluated in terms of both the win rate and the efficiency. Moreover, we also discuss how algorithm parameters would affect the performance indices of the BIO algorithm.

6.1. Results Analysis for Single Match

6.1.1. Entire Confrontation Process of a Match

Figure 5 presents the entire confrontation process of a single match with $N = 5$. The red team members, adopting the BIO algorithm, are highlighted with circles, while the blue team members, adopting the AN algorithm, are highlighted with squares. Specifically, (a) the two teams are generated on different sides of the map; (b) both teams advance towards each other; (c) the red team initiates combat, and an opponent is destroyed; (d) the red team destroys another opponent, and then some members retreat tactically; (e) the red team destroys the third opponent; (f) the red team member at the bottom moves left to attack the opponent on the left; (g) the red team's main force destroys the fourth opponent but also loses a member; and (h) the red team destroys the last opponent and wins the match.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**)

Figure 5. *Cont.*



(**g**) (**h**)

Figure 5. The entire confrontation process of a single match with $N = 5$. The red team members, adopting the BIO algorithm, are highlighted with circles, while the blue team members, adopting the AN algorithm, are highlighted with squares. (**a**) The two teams are generated on different sides of the map. (**b**) Both teams advance towards each other. (**c**) The red team initiates combat, and an opponent is destroyed. (**d**) The red team destroys another opponent, and then some members retreat tactically. (**e**) The red team destroys the third opponent. (**f**) The red team member at the bottom moves left to attack the opponent on the left. (**g**) The red team's main force destroys the fourth opponent but also loses a member. (**h**) The red team destroys the last opponent and wins the match.

6.1.2. Typical Scenes for the Bio-Inspired Strategies

Two scene segments have been taken to illustrate the bio-inspired strategies. Similar to before, the red team members adopt the BIO algorithm, while the blue team members adopt the AN algorithm.

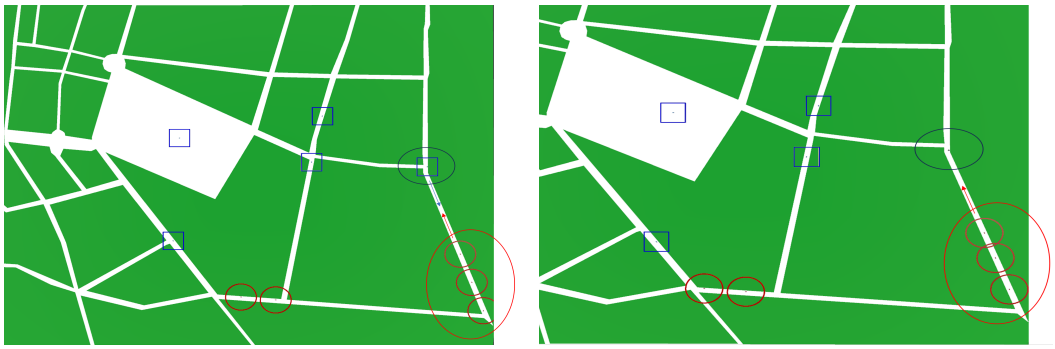
Figure 6 illustrates the "fire concentration" strategy. In particular, (a) the conditions for the "fire concentration" strategy are all satisfied, and three red team members are assigned a common target; (b) the three red team members move toward the target; (c) the three red team members attack the target; and (d) the target is destroyed.

Figure 7 illustrates the "back and forth maneuver" strategy. In particular, (a) the two red team members move toward the target; (b) the two red team members attack the target; (c) the two red team members retreat a little bit after firing to avoid being hit; and (d) the two red team members go back and move toward the target again when the shells are ready, and one opponent is destroyed. This scenario represents a special case in which the red team, having entered the road first, sets up an ambush for the blue team's tanks. As soon as the blue team exposes part of their tanks, they are immediately attacked by the red team, which employs precise sniper tactics. The attack is timed to prevent the blue team from retaliating, resulting in their tanks being destroyed before they can respond. Simultaneously, the red team executes a tactical retreat, firing while maneuvering to avoid incoming damage.



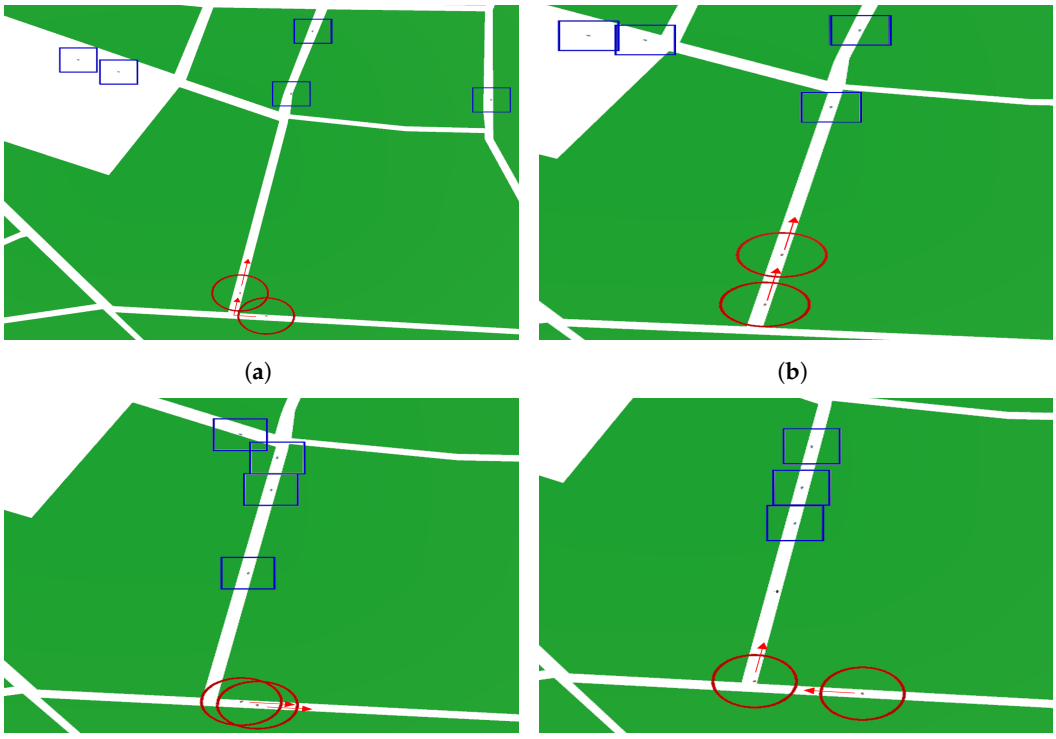
(**a**) (**b**)

Figure 6. *Cont.*



(**c**) (**d**)

Figure 6. Illustration of the "fire concentration" strategy, where the red team members adopt the BIO algorithm while the blue team members adopt the AN algorithm. (**a**) The conditions for the "fire concentration" strategy are all satisfied, and three red team members are assigned a common target. (**b**) The three red team members move toward the target. (**c**) The three red team members attack the target. (**d**) The target is destroyed.



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

(**c**) (**d**)

Figure 7. Illustration of the "back and forth maneuver" strategy, where the red team members adopt the BIO algorithm, while the blue team members adopt the AN algorithm. (**a**) The two red team members move toward the target. (**b**) The two red team members attack the target. (**c**) The two red team members retreat a little bit after firing to avoid being hit. (**d**) The two red team members go back and move toward the target again when the shells are ready, and one opponent is destroyed.

6.1.3. Algorithm Computational Complexity Analysis

The computational resource consumption of the BIO algorithm is shown by Figure 8. The main functions of the BIO algorithm are implemented through the `FixedUpdate` function. The scripts `Fire_Concentration`, `Pathcalculate`, and `ManControl` corresponding to the "fire concentration" strategy, the "back and forth maneuver" strategy, and the basic algorithmic functionalities, respectively. It can be observed that they utilize 8.0%, 0.6%, and 4.3% of CPU computational resources, respectively. The basic functionalities encompass a wide array

of operations, with a reasonably managed computational complexity. In contrast, the computational complexity of the "back and forth maneuver" strategy is quite low, whereas the "fire concentration" operation exhibits a significantly higher complexity. The reason for the high CPU utilization by the "fire concentration" strategy primarily stems from extensive computations of navigation distances, which involve a substantial number of subdivided grid details of the navigation mesh. This process significantly increases the consumption of computational resources. However, overall, the algorithm's demand for computing resources remains moderate. During the test, the algorithm operates on the host computer with the configuration specified in Table 6.

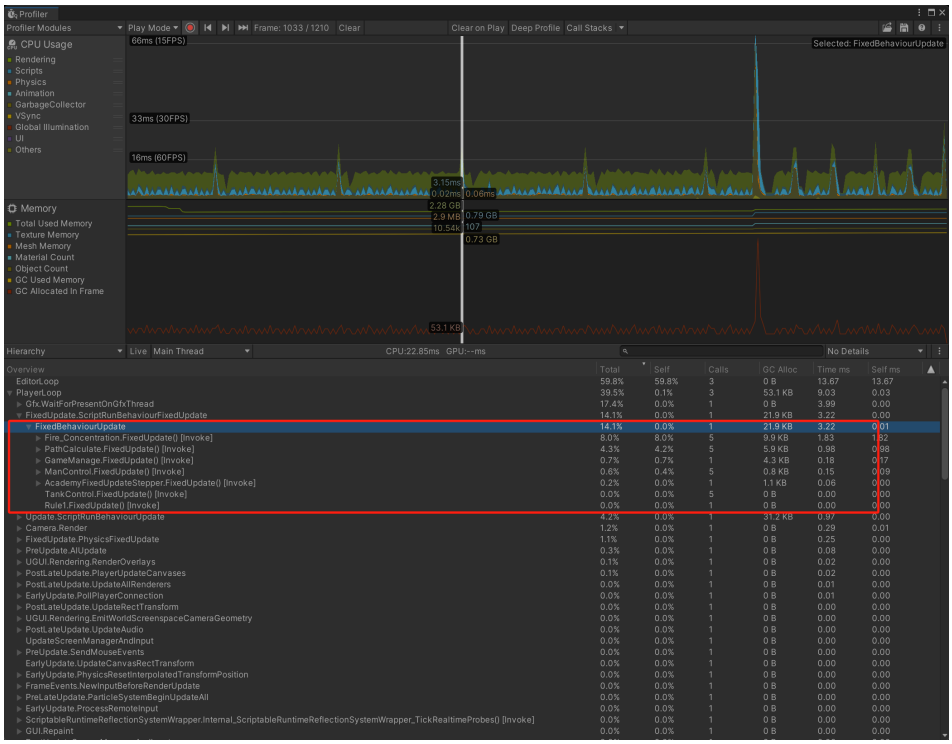


Figure 8. The algorithm performance consumption analysis.

Table 6. Computer configuration details.

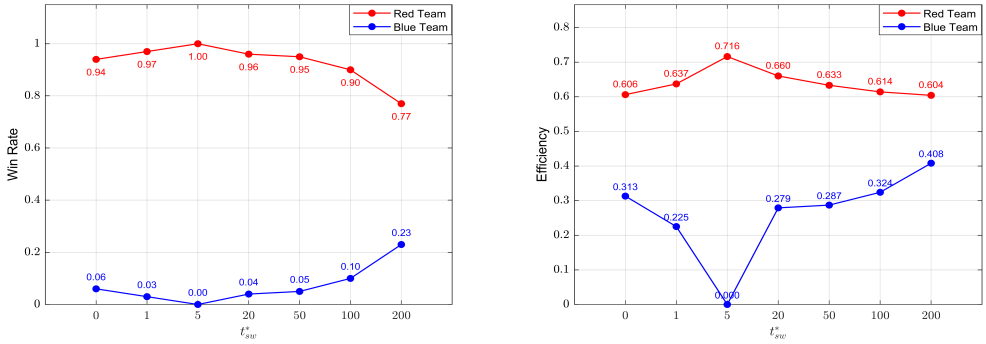
Component	Specification
CPU	AMD Ryzen 5 5600X 6-Core Processor, 3.70 GHz
GPU	Nvidia GeForce GT 1030
RAM	16 GB DDR4
Operation System	Windows 11 Family Version

6.2. Results Analysis for Group Matching

Next, we perform group matching between the BIO algorithm and the RL algorithm, as well as between the BIO algorithm and the AN algorithm. A group of matches consists of 100 matches, and both the cases of $N = 5$ and $N = 10$ are tested. Different swarm confrontation algorithms are compared in terms of win rate and efficiency. As mentioned before, the minimal dwell time for target switching t^*_{sw}, the minimal safety firing distance for the back and forth maneuver strategy d_{back}, and the triggering distance of the "fire concentration" strategy ℓ are the three key parameters for the BIO algorithm. Thereby, a comprehensive series of comparative tests have been conducted to reveal the influence of these three parameters on the performance indices. In particular, by setting $d_{back} = 1200$ m and $\ell = 1800$ m, we let $t^*_{sw} = 0, 1, 5, 20, 50, 100, 200$ steps; by setting $t^*_{sw} = 20$ steps and $\ell = 1800$ m, we let $d_{back} = 0, 100, 300, 500, 700, 900, 1100, 1200$ m; and by setting $t^*_{sw} = 20$ steps and $d_{back} = 1200$ m, we let $\ell = 1300, 1400, 1500, 1600, 1700, 1800, 1900$ m.

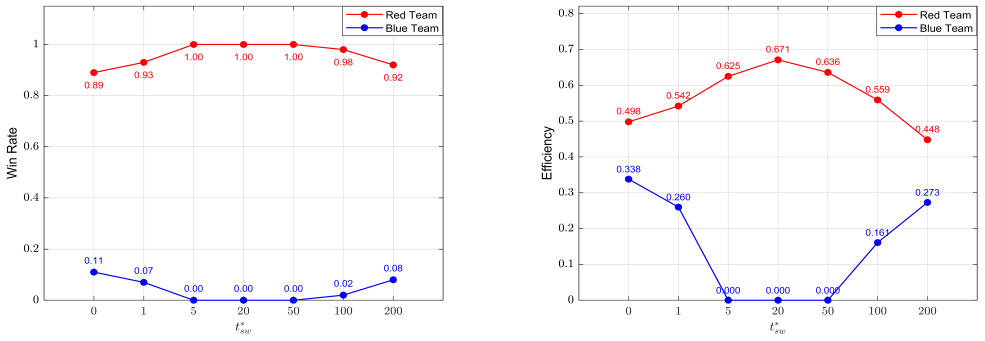
6.2.1. The BIO Algorithm vs. The RL Algorithm

The confrontation results between the BIO algorithm and the RL algorithm are illustrated in Figures 9–12, which demonstrate that in all cases, the BIO algorithm outperforms the RL algorithm.



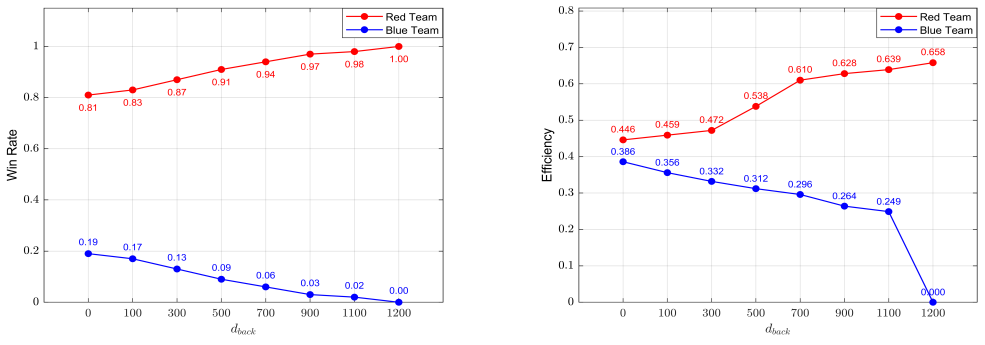
(**a**) (**b**)

Figure 9. Confrontation results for the case of $N = 5$ subject to different values of t^*_{sw}, with the red team adopting the BIO algorithm and the blue team adopting the RL algorithm. (**a**) Win rate. (**b**) Efficiency.



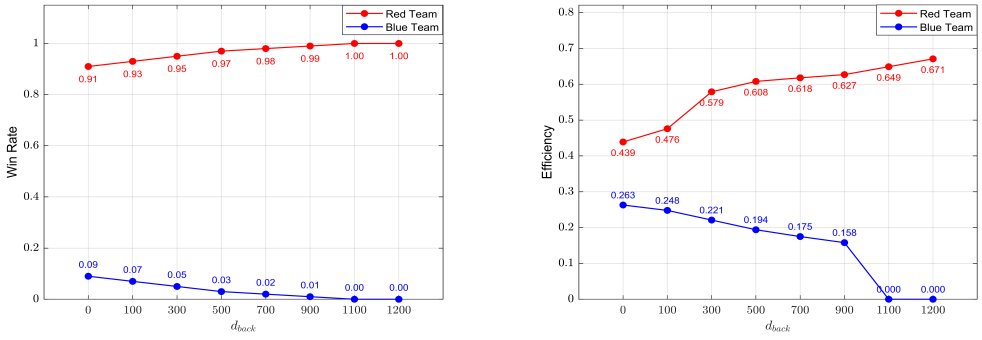
(**a**) (**b**)

Figure 10. Confrontation results for the case of $N = 10$ subject to different values of t^*_{sw}, with the red team adopting the BIO algorithm and the blue team adopting the RL algorithm. (**a**) Win rate. (**b**) Efficiency.



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

Figure 11. Confrontation results for the case of $N = 5$ subject to different d_{back}, with the red team adopting the BIO algorithm and the blue team adopting the RL algorithm. (**a**) win rate (**b**) efficiency.



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

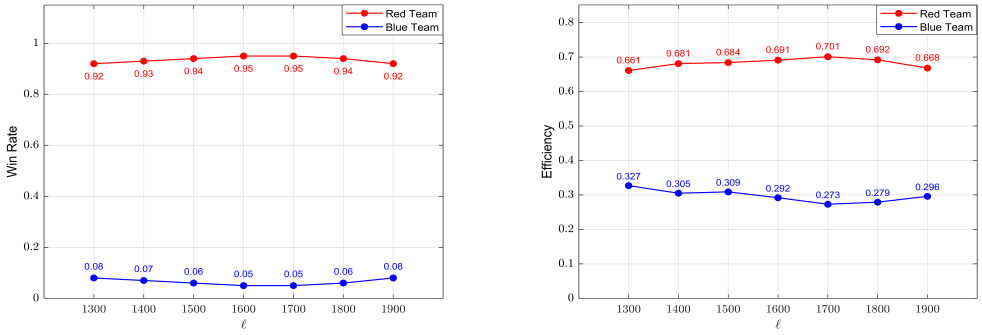
Figure 12. Confrontation results for the case of $N = 10$ subject to different values of d_{back}, with the red team adopting the BIO algorithm and the blue team adopting the RL algorithm. (**a**) Win rate. (**b**) Efficiency.

In Figure 9, it is observed that when $N = 5$, the BIO algorithm achieves its highest win rate and efficiency against the RL algorithm when $t_{sw}^{*} = 5$ steps. Similarly, Figure 10 shows that the BIO algorithm reaches its peak performance in terms of win rate and efficiency against the RL algorithm when $t_{sw}^{*} = 20$ steps. These results indicate that there might exist an optimal minimal dwell time for a target switching t_{sw}^{*} regarding algorithm performance, and the optimal value may vary with N. Notably, when $t_{sw}^{*} < 5$ steps, a significant overall decrease in win rate and efficiency becomes apparent. Though frequent target switching may enable the tank to better adapt to the dynamic confrontation scenario, occasionally, the tank may get stuck in selecting targets and thus miss the opportunity to attack. On the other hand, $t_{sw}^{*} > 20$ steps also results in a similar reduction in win rate and efficiency as the decision-making would lag far behind the current confrontation situation. Furthermore, compared to $t_{sw}^{*} = 0$ steps, which corresponds to the absence of a dwell time for target switching, the maximum win rates for the cases of $N = 5$ and $N = 10$ are improved by 6% and 11%, respectively, and the maximum efficiencies for the cases of $N = 5$ and $N = 10$ are enhanced by 0.112 and 0.223, respectively. These enhancements underscore the significant impact of the target switching mechanism in boosting the BIO algorithm's win rate and efficiency.

In Figures 11 and 12, both the win rate and the efficiency peak at around $d_{back} = 1200$ m, and this is because the range of the shell is set to be 1200 m. The test results confirm that being at the critical point of the shell range is actually the optimal choice, which allows prompt firing and effective retreating. Moreover, in comparison to the case where $d_{back} = 0$ m, indicating the absence of the "back and forth maneuver" strategy, the incorporation of this strategy leads to significant performance improvements in confrontations. Specifically, for the cases of $N = 5$ and $N = 10$, the maximum win rate enhancements are 19% and 9%, respectively, and the increases in maximum efficiency are notably 0.212 and 0.232, respectively.

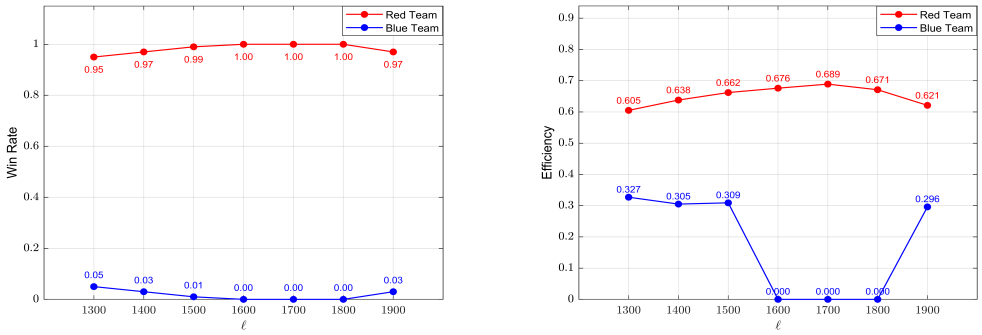
In Figures 13 and 14, the win rate trend of the BIO algorithm as ℓ varies first increases and then achieves its maximum value at $\ell = 1700$ m. The efficiency trend also improves with increasing ℓ, peaking at $\ell = 1700$ m. Beyond this point, both metrics begin to decline slightly, indicating that $\ell = 1700$ m is the optimal engagement distance under these testing conditions. When $\ell < 1700$ m, both the win rate and the efficiency increase as ℓ rises. This improvement is primarily due to the fact that the increasing distance allows the BIO team to engage more effectively without entering into the immediate firing range of the RL opponent, which is crucial in avoiding direct confrontations that are less favorable in a conservative combat scenario. As the distance increases, the BIO algorithm leverages its strategic capabilities better, resulting in improved targeting and maneuvering space, which contributes to the overall combat effectiveness. Additionally, for smaller values of ℓ, such as 1300 m and 1400 m, the "fire concentration" strategy defined by the BIO algorithm becomes challenging to trigger. When $\ell > 1700$ m, both the win rate and the efficiency begin to decrease, primarily due to the inefficiency in tactical execution in complex environments. At these larger distances, tanks under the BIO algorithm often need to move significantly

further to participate in concentrated fire efforts, which can be especially challenging in environmental settings with numerous obstacles. This increased movement reduces not only the speed of engagement but also the overall tactical responsiveness. At $\ell = 1700$ m, compared to $\ell = 1300$ m, the win rates for the 5V5 and 10V10 configurations increase by 3% and 5%, respectively, while the efficiency improves by 0.040 and 0.084, respectively.



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

Figure 13. Confrontation results for the case of $N = 5$ subject to different values of ℓ, with the red team adopting the BIO algorithm and the blue team adopting the RL algorithm. (**a**) Win rate. (**b**) Efficiency.

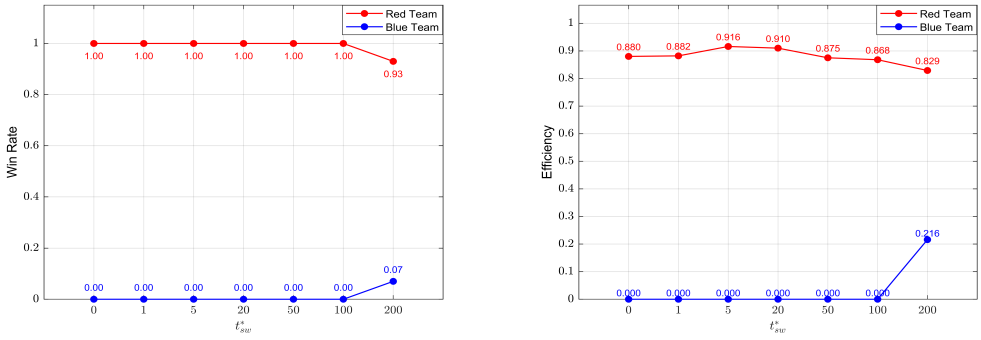


(**a**) (**b**)

Figure 14. Confrontation results for the case of $N = 10$ subject to different values of ℓ, with the red team adopting the BIO algorithm and the blue team adopting the RL algorithm. (**a**) Win rate. (**b**) Efficiency.

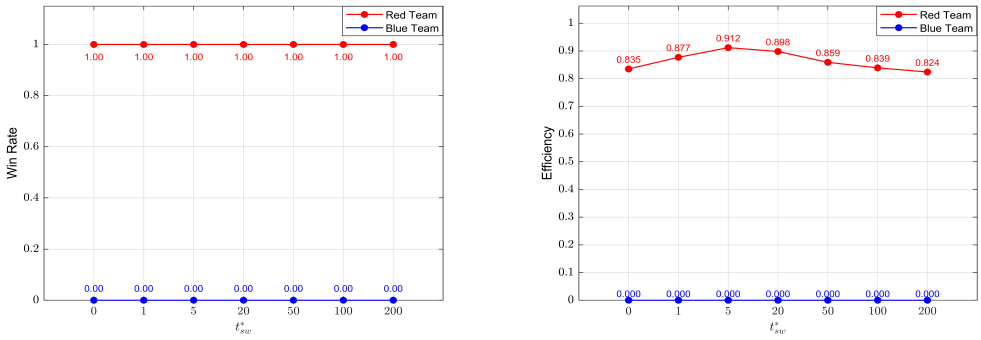
6.2.2. The BIO Algorithm vs. The AN Algorithm

The confrontation results between the BIO algorithm and the AN algorithm are illustrated in Figures 13–16. It can be seen that the BIO algorithm achieves a win rate of over 91% against the AN algorithm in all scenarios, demonstrating the robust performance of the BIO algorithm. Moreover, if we compare the results of Figures 9–16, it can be found out that the RL algorithm has a better performance than the AN algorithm when facing the same opponent, revealing the fact that simple rules might not be suitable for complex confrontation scenarios.



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

Figure 15. Confrontation results for the case of $N = 5$ subject to different values of t_{sw}^*, with the red team adopting the BIO algorithm and the blue team adopting the AN algorithm. (**a**) Win rate. (**b**) Efficiency.

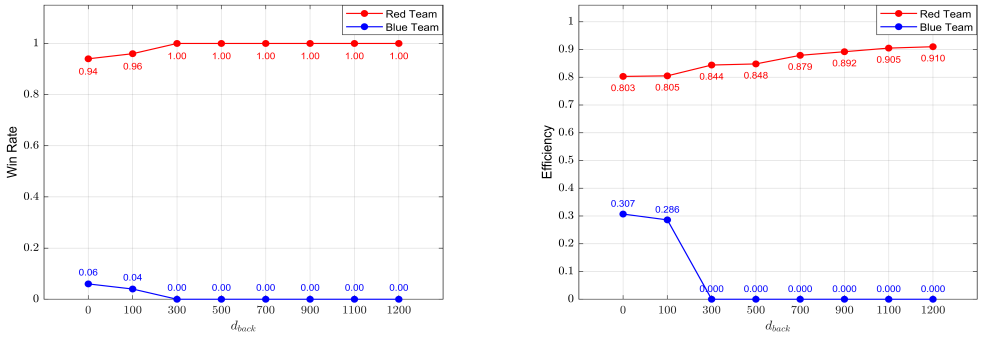


(**a**) (**b**)

Figure 16. Confrontation results for the case of $N = 10$ subject to different values of t_{sw}^*, with the red team adopting the BIO algorithm and the blue team adopting the AN algorithm. (**a**) Win rate. (**b**) Efficiency.

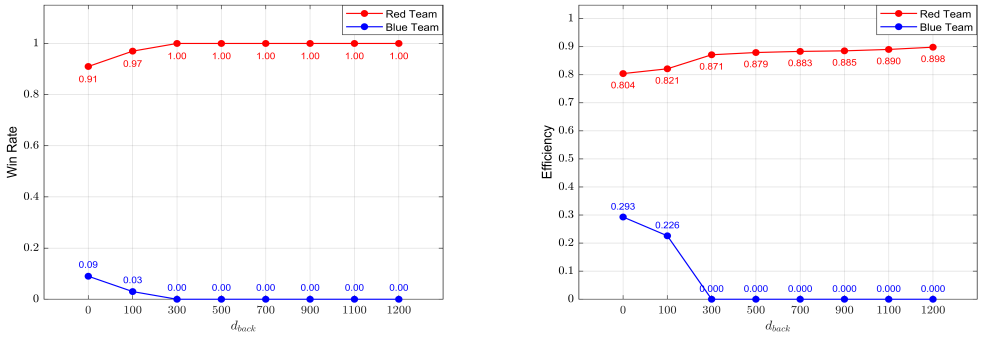
In Figures 15 and 16, the win rate of the BIO algorithm is almost 100%, except for a slightly lower rate of 93% in the 5V5 scenario with $t_{sw}^* = 200$ steps. In both the 5V5 and 10V10 scenarios, the BIO algorithm's efficiency peaks at $t_{sw}^* = 5$ steps. Compared to $t_{sw}^* = 0$ step, setting $t_{sw}^* = 5$ steps results in an efficiency increase of 0.051 and 0.088 for the BIO algorithm in the 5V5 and 10V10 configurations, respectively.

In Figures 17 and 18, the win rate and efficiency of the BIO algorithm overall exhibit an increasing trend with the rise in d_{back}. In both the 5V5 and 10V10 configurations, the win rates remain at 100% for $d_{back} \geq 300$, and the maximum efficiency is achieved at $d_{back} = 1200$ m. At $d_{back} = 1200$ m, compared to $d_{back} = 0$ m, the win rates for the 5V5 and 10V10 configurations increase by 6% and 4%, respectively, while the efficiency improves by 0.107 and 0.094, respectively. We believe the reasons behind these phenomena are largely consistent with those observed in Figures 11 and 12. Compared to the BIO vs. AN scenario, the decline in the win rate and efficiency for $d_{back} < 300$ m is more pronounced in the BIO vs. RL scenario, probably because the RL algorithm has a higher hit rate.



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

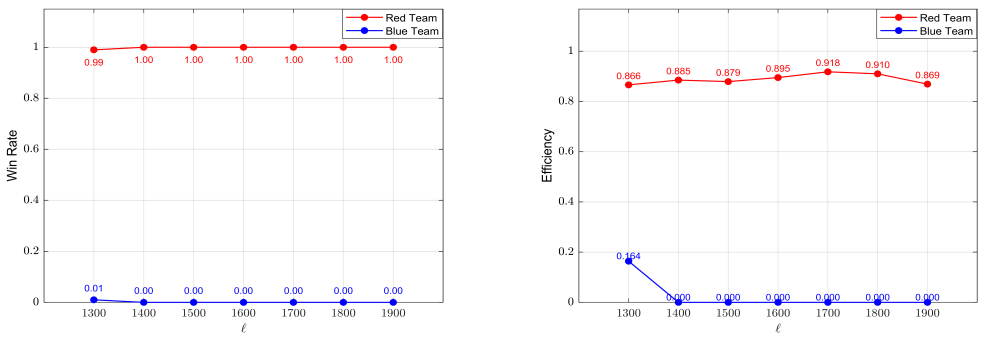
Figure 17. Confrontation results for the case of $N = 5$ subject to different values of d_{back}, with the red team adopting the BIO algorithm and the blue team adopting the AN algorithm. (**a**) Win rate. (**b**) Efficiency.



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

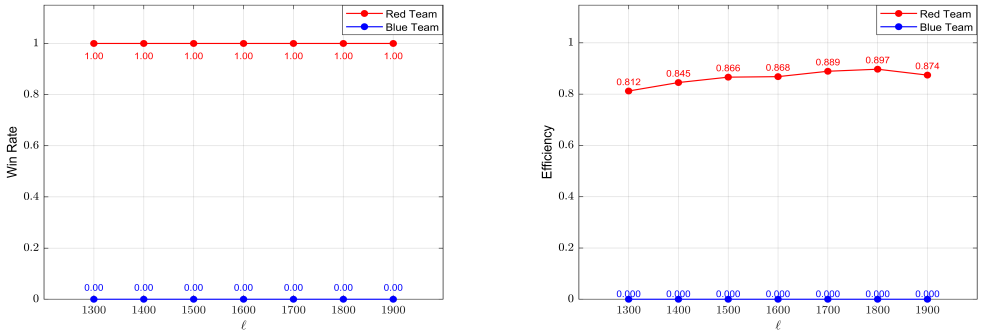
Figure 18. Confrontation results for the case of $N = 10$ subject to different values of d_{back}, with the red team adopting the BIO algorithm and the blue team adopting the AN algorithm. (**a**) Win rate. (**b**) Efficiency.

In Figures 19 and 20, it is observed that the win rate shows minor variation, and the efficiency is also marginally affected by changes in ℓ, although the general trend still increases before decreasing. The reasons for these results are consistent with those discussed in Figures 13 and 14. Specifically, when $N = 5$, the maximum efficiency is achieved at $\ell = 1700$ m. In contrast, for a larger swarm $N = 10$, peak efficiency is reached at $\ell = 1800$ m, although the differences between these settings are slight. Compared to a triggering distance of $\ell = 1300$ m, when ℓ is optimized, the win rates for the 5V5 and 10V10 scenarios are increased by 1% and 0%, respectively, with corresponding gains in efficiency of 0.052 and 0.085.



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

Figure 19. Confrontation results for the case of $N = 5$ subject to different values of ℓ, with the red team adopting the BIO algorithm and the blue team adopting the AN algorithm. (**a**) Win rate. (**b**) Efficiency.



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

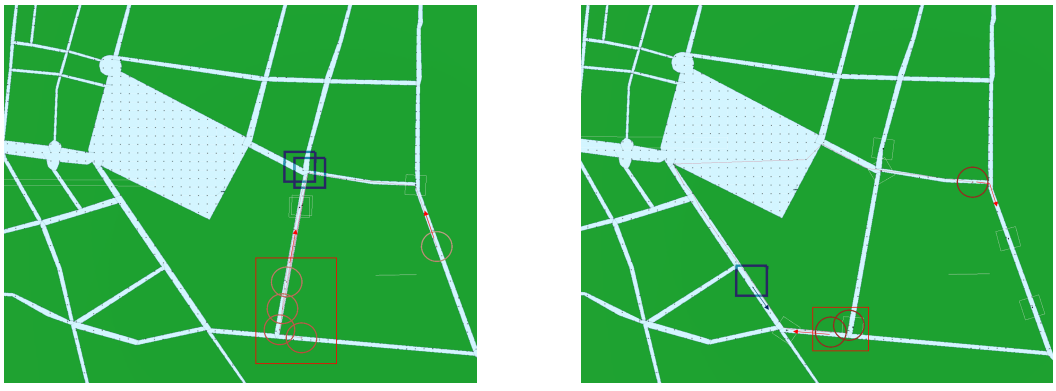
Figure 20. Confrontation results for the case of $N = 10$ subject to different values of ℓ, with the red team adopting the BIO algorithm and the blue team adopting the AN algorithm. (**a**) Win rate. (**b**) Efficiency.

However, synthesizing the data from Figures 13, 14, 17 and 18 shows that the variability in efficiency due to changes in ℓ is more pronounced for the team size $N = 10$ than for $N = 5$. This increased sensitivity in larger teams can be attributed to the fact that with more team members, the coordination and alignment for effective "fire concentration" becomes more challenging. The greater number of agents increases the potential for misalignment in positioning and timing, which can amplify the impact of suboptimal distances on overall efficiency. Additionally, in larger teams, the need to maintain formation and cover more ground to utilize the "fire concentration" strategy effectively at varying ℓ distances necessitates more complex and precise maneuvering, further contributing to the observed variability in efficiency.

6.2.3. Analysis of Tank Behavior under the RL and AN Algorithms

In the confrontation process, tanks under the RL Algorithm exhibit significant behavioral uncertainty, a characteristic of the neural network models. Meanhile, in general, the following behavioral patterns can be observed during engagements.

1. Tanks do not necessarily choose their nearest opponents as targets, and the choice of targets may change constantly.
2. During the approach phase, tanks tend to move at a slow pace.
3. Tanks often position themselves at the edges of intersections of roads, strategically waiting to ambush opponents (Figure 21a).
4. Tanks frequently execute flanking maneuvers, aiming to stealthily attack from the sides (Figure 21b).



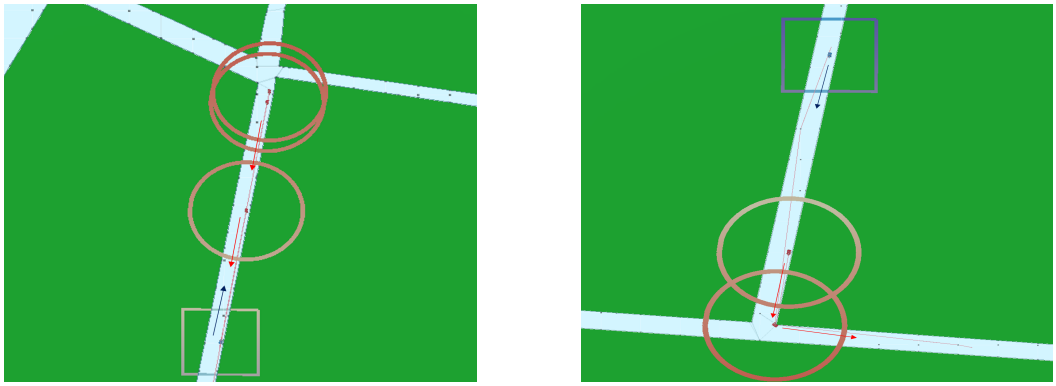
(**a**) (**b**)

Figure 21. Analysis of tank behavior under the RL algorithm (blue team). (**a**) Tanks often position themselves at the edges of intersections of roads, strategically waiting to ambush opponents. (**b**) Tanks frequently execute flanking maneuvers, aiming to stealthily attack from the sides.

A major drawback of the RL algorithm is that the tanks often perform in a manner that could be perceived as clumsy. Their movement is slow, and they occasionally collide with obstacles, demonstrating a lack of fluid path navigation and obstacle avoidance. In contrast, the tanks under the BIO algorithm, driven by scripted intelligence, show highly efficient decision-making and execution capabilities with smooth path-finding and obstacle avoidance, which explains the higher win rate and efficiency of the BIO algorithm.

In contrast to the RL algorithm, the AN algorithm is rule-based, which means that the tank behaviors are deterministic for all scenarios. Below are summaries of some typical behavioral patterns for tanks under the AN algorithm.

1. Tanks immediately lock onto and approach their targets at the beginning of the engagement.
2. Tanks advance towards their targets in an orderly manner, avoiding collisions with obstacles or teammates.
3. Tanks directly confront opponents, engaging them head-on in combat scenarios (Figure 22a).
4. Tanks actively pursue retreating opponents, showcasing persistent and strategic chasing behavior (Figure 22b).



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

Figure 22. Tanks behavior analysis under the AN algorithm (blue team). (**a**) Tanks directly confront opponents, engaging them head-on in combat scenarios. (**b**) Tanks actively pursue retreating opponents, showcasing persistent and strategic chasing behavior.

In contrast to the BIO Algorithm, the AN Algorithm is in general overly aggressive. This lack of strategic retreat and adaptability frequently exposes the tanks to enhanced risks of opponents' attacks and counterattacks, which lowers down the win rate and efficiency of the AN Algorithm.

7. Conclusions

This paper considers the swarm confrontation problem for two tank swarms in a complex urban scenario. A novel intelligent swarm confrontation algorithm is proposed featuring two bio-inspired strategies, namely the "fire concentration" strategy and the "back and forth maneuver" strategy. The former one helps to create a favorable combat situation, while the latter one may reduce the risk of being hit when the shell is reloading. Compared to two other state-of-the-art swarm confrontation algorithms, the bio-inspired swarm confrontation algorithm exhibits significant advantages in terms of the win rate and efficiency. Moreover, it is also revealed how algorithm parameters would affect algorithm performance.

In this paper, we have considered the cases of $N = 5$ and $N = 10$, while, for larger swarms in similar urban environments, it might be necessary for the BIO algorithm to employ a hierarchical structure so that the swarm can be divided into several small subswarms. The reason lies in the fact that the effectiveness of the "fire concentration" strategy may be compromised since the roads are narrow, which cannot accommodate a large number of tanks for focused fire. Moreover, regarding the "back and forth maneuver" strategy, the retreating of tanks during the "back and forth maneuver" could cause severe

congestion for large swarms. On the other hand, the BIO algorithm would be well suited for different environments with similar urban characteristics since it is based on the foundation of navigation. There will be no difference in how to select targets for different environments.

Regarding future work, techniques such as deep reinforcement learning and evolutionary strategies may offer avenues for the further optimization of the algorithm's parameters. Furthermore, due to limitations of space, we cannot conduct comparative study with more existing works within this paper. We may further do this in the future with, say, [47–51], which involves RL-based methods, visual language models, game-theoretic approaches, and evolutionary strategies. Additionally, it is necessary to further investigate the dynamics of swarm confrontation under conditions of incomplete information in order to enhance the broader applicability of the bio-inspired confrontation algorithm.

Author Contributions: Conceptualization, H.G. and H.C.; methodology, Y.L. and H.C.; software, Y.L. and G.W.; validation, Y.L.; investigation, H.G., Y.L. and H.C.; writing—original draft preparation, Y.L. and H.C.; visualization, Y.L. and G.W.; supervision, H.C.; funding acquisition, H.G. and H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by National Natural Science Foundation of China under grant number 62173149, 62276104, U22A2062, in part by Guangdong Natural Science Foundation under grant number 2021A1515012584, 2022A1515011262, and in part by Fundamental Research Funds for the Central Universities.

Data Availability Statement: The data of this paper are available upon request.

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

References

1. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
2. Jaderberg, M.; Czarnecki, W.M.; Dunning, I.; Marris, L.; Lever, G.; Castañeda, A.G.; Beattie, C.; Rabinowitz, N.C.; Morcos, A.S.; Ruderman, A.; et al. Human-level performance in 3D multiplayer games with population-based reinforcement learning. *Science* **2019**, *364*, 859–865. [CrossRef] [PubMed]
3. Vinyals, O.; Babuschkin, I.; Czarnecki, W.M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D.H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* **2019**, *575*, 350–354. [CrossRef] [PubMed]
4. Day, M. Multi-Agent Task Negotiation Among UAVs to Defend Against Swarm Attacks. Ph.D. Thesis, Naval Postgraduate School, Monterey, CA, USA, 2012.
5. Tsach, S.; Peled, A.; Penn, D.; Keshales, B.; Guedj, R. Development Trends for Next Generation of UAV Systems. In Proceedings of the AIAA Infotech@ Aerospace 2007 Conference and Exhibit, Rohnert Park, CA, USA, 7–10 May 2007; p. 2762.
6. Ayamga, M.; Akaba, S.; Nyaaba, A.A. Multifaceted Applicability of Drones: A Review. *Technol. Forecast. Soc. Change* **2021**, *167*, 120677. [CrossRef]
7. Zuo, Z.; Liu, C.; Han, Q.-L.; Song, J. Unmanned Aerial Vehicles: Control Methods and Future Challenges. *IEEE/CAA J. Autom. Sin.* **2022**, *9*, 601–614. [CrossRef]
8. Gergal, E.K. *Drone Swarming Tactics Using Reinforcement Learning and Policy Optimization*; United States Naval Academy: Annapolis, MD, USA, 2021.
9. Cohen, A.; Teng, E.; Berges, V.-P.; Dong, R.-P.; Henry, H.; Mattar, M.; Zook, A.; Ganguly, S. On the Use and Misuse of Absorbing States in Multi-Agent Reinforcement Learning. *arXiv* **2021**, arXiv:2111.05992.
10. Thakoor, O.; Garg, J.; Nagi, R. Multiagent UAV Routing: A Game Theory Analysis with Tight Price of Anarchy Bounds. *IEEE Trans. Autom. Sci. Eng.* **2019**, *17*, 100–116. [CrossRef]
11. Zhou, Y.; Rao, B.; Wang, W. UAV Swarm Intelligence: Recent Advances and Future Trends. *IEEE Access* **2020**, *8*, 183856–183878. [CrossRef]
12. Cruz, J.B.; Simaan, M.A.; Gacic, A.; Jiang, H.; Letelliier, B.; Li, M.; Liu, Y. Game-Theoretic Modeling and Control of a Military Air Operation. *IEEE Trans. Aerosp. Electron. Syst.* **2001**, *37*, 1393–1405. [CrossRef]
13. Cruz, J.B.; Simaan, M.A.; Gacic, A.; Liu, Y. Moving Horizon Nash Strategies for a Military Air Operation. *IEEE Trans. Aerosp. Electron. Syst.* **2002**, *38*, 989–999. [CrossRef]
14. Nowak, D.J. Exploitation of Self Organization in UAV Swarms for Optimization in Combat Environments. Master's Thesis, Air Force Institute of Technology, Kawo, Kaduna, 2008.
15. Bhattacharya, S.; Başar, T. Game-Theoretic Analysis of an Aerial Jamming Attack on a UAV Communication Network. In Proceedings of the 2010 American Control Conference, Baltimore, MD, USA, 30 June–2 July 2010; pp. 818–823.

16. Yao, Z.; Li, M.; Chen, Z.; Zhou, R. Mission Decision-Making Method of Multi-Aircraft Cooperatively Attacking Multi-Target Based on Game Theoretic Framework. *Chin. J. Aeronaut.* **2016**, *29*, 1685–1694. [CrossRef]
17. Özpala, A.; Efe, M.Ö.; Sever, H. Multiple UAV Engagement Decision by Game Theory. *Int. J. Comput. Electr. Eng.* **2017**, *9*, 384–392. [CrossRef]
18. Ma, Y.; Wang, G.; Hu, X.; Luo, H.; Lei, X. Cooperative Occupancy Decision Making of Multi-UAV in Beyond-Visual-Range Air Combat: A Game Theory Approach. *IEEE Access* **2019**, *8*, 11624–11634. [CrossRef]
19. Foo, J.L.; Knutzon, J.; Oliver, J.; Winer, E. Three-Dimensional Path Planning of Unmanned Aerial Vehicles Using Particle Swarm Optimization. In Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Portsmouth, VA, USA, 6 September 2006–8 September 2006; p. 6995.
20. Zhang, Y.; Wang, S.; Ji, G. A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications. *Math. Probl. Eng.* **2015**, *2015*, 931256. [CrossRef]
21. Yu, X.; Zhang, Y. Sense and Avoid Technologies with Applications to Unmanned Aircraft Systems: Review and Prospects. *Prog. Aerosp. Sci.* **2015**, *74*, 152–166. [CrossRef]
22. Sauter, J.A.; Mathews, R.S.; Yinger, A.; Robinson, J.S.; Moody, J.; Riddle, S. Distributed Pheromone-Based Swarming Control of Unmanned Air and Ground Vehicles for RSTA. In Proceedings of the Unmanned Systems Technology X, Orlando, FL, USA, 17–20 March 2008; Volume 6962, pp. 109–120.
23. Foo, J.L.; Knutzon, J.; Kalivarapu, V.; Oliver, J.; Winer, E. Path Planning of Unmanned Aerial Vehicles Using B-Splines and Particle Swarm Optimization. *J. Aerosp. Comput. Inf. Commun.* **2009**, *6*, 271–290. [CrossRef]
24. Li, P.; Duan, H.B. Path Planning of Unmanned Aerial Vehicle Based on Improved Gravitational Search Algorithm. *Sci. China Technol. Sci.* **2012**, *55*, 2712–2719. [CrossRef]
25. Duan, H.; Luo, Q.; Shi, Y.; Ma, G. Hybrid Particle Swarm Optimization and Genetic Algorithm for Multi-UAV Formation Reconfiguration. *IEEE Comput. Intell. Mag.* **2013**, *8*, 16–27. [CrossRef]
26. Duan, H.; Li, P.; Yu, Y. A predator–prey Particle Swarm Optimization Approach to Multiple UCAV Air Combat Modeled by Dynamic Game Theory. *IEEE/CAA J. Autom. Sin.* **2015**, *2*, 11–18. [CrossRef]
27. Dolicanin, E.; Fetahovic, I.; Tuba, E.; Capor-Hrosik, R.; Tuba, M. Unmanned Combat Aerial Vehicle Path Planning by Brain Storm Optimization Algorithm. *Stud. Inform. Control* **2018**, *27*, 15–24. [CrossRef]
28. Li, Y.; Zhang, S.; Chen, J.; Jiang, T.; Ye, F. Multi-UAV Cooperative Mission Assignment Algorithm Based on ACO Method. In Proceedings of the 2020 International Conference on Computing, Networking and Communications (ICNC), Big Island, HI, USA, 17–20 February 2020; pp. 304–308.
29. Ye, F.; Chen, J.; Tian, Y.; Jiang, T. Cooperative Multiple Task Assignment of Heterogeneous UAVs Using a Modified Genetic Algorithm with Multi-Type-Gene Chromosome Encoding Strategy. *J. Intell. Robot. Syst.* **2020**, *100*, 615–627. [CrossRef]
30. Li, S.; Wang, Y.; Wu, C.; Chen, Z. Artificial Intelligence and Unmanned Warfare. In Proceedings of the 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), Nanjing, China, 23–25 November 2018; pp. 336–339.
31. Ren, Y.; Cao, X.-Q.; Guo, Y.-N.; Peng, K.-C.; Xiao, C.-H.; Tian, W.-L. Application of Machine Learning in UAV Combat. In Proceedings of the International Conference on Autonomous Unmanned Systems, Changsha, China, 24–26 September 2021; pp. 2963–2969.
32. Chaurasia, R.; Mohindru, V. Unmanned Aerial Vehicle (UAV): A Comprehensive Survey. In *Unmanned Aerial Vehicles for Internet of Things (IoT) Concepts, Techniques, and Applications*; Wiley: Hoboken, NJ, USA, 2021; pp. 1–27.
33. Rezwan, S.; Choi, W. Artificial Intelligence Approaches for UAV Navigation: Recent Advances and Future Challenges. *IEEE Access* **2022**, *10*, 26320–26339. [CrossRef]
34. Kurunathan, H.; Huang, H.; Li, K.; Ni, W.; Hossain, E. Machine Learning-Aided Operations and Communications of Unmanned Aerial Vehicles: A Contemporary Survey. *IEEE Commun. Surv. Tutor.* **2023**, *26*, 496–533. [CrossRef]
35. Liu, P.; Ma, Y. A Deep Reinforcement Learning Based Intelligent Decision Method for UCAV Air Combat. In Proceedings of the Modeling, Design and Simulation of Systems: 17th Asia Simulation Conference, AsiaSim 2017, Melaka, Malaysia, 27–29 August 2017; Part I 17, pp. 274–286.
36. Yang, Q.; Zhang, J.; Shi, G.; Hu, J.; Wu, Y. Maneuver Decision of UAV in Short-Range Air Combat Based on Deep Reinforcement Learning. *IEEE Access* **2019**, *8*, 363–378. [CrossRef]
37. Lee, G.T.; Kim, C.O. Autonomous Control of Combat Unmanned Aerial Vehicles to Evade Surface-to-Air Missiles Using Deep Reinforcement Learning. *IEEE Access* **2020**, *8*, 226724–226736. [CrossRef]
38. Zhang, J.; Yang, Q.; Shi, G.; Lu, Y.; Wu, Y. UAV Cooperative Air Combat Maneuver Decision Based on Multi-Agent Reinforcement Learning. *J. Syst. Eng. Electron.* **2021**, *32*, 1421–1438.
39. Hu, J.; Wang, L.; Hu, T. Autonomous Maneuver Decision Making of Dual-UAV Cooperative Air Combat Based on Deep Reinforcement Learning. *Electronics* **2022**, *11*, 467. [CrossRef]
40. Navigation System in Unity. Available online: https://docs.unity3d.com/2023.1/Documentation/Manual/nav-NavigationSystem.html (accessed on 29 April 2023).
41. Chi, P.; Wei, J.; Wu, K.; Di, B.; Wang, Y. A Bio-Inspired Decision-Making Method of UAV Swarm for Attack-Defense Confrontation via Multi-Agent Reinforcement Learning. *Biomimetics* **2023**, *8*, 222. [CrossRef] [PubMed]
42. Xiang, L.; Xie, T. Research on UAV Swarm Confrontation Task Based on MADDPG Algorithm. In Proceedings of the 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), Harbin, China, 25–27 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1513–1518.

43. Zhou, L.; Leng, S.; Liu, Q.; Wang, Q. Intelligent UAV Swarm Cooperation for Multiple Targets Tracking. *IEEE Internet Things J.* **2021**, *9*, 743–754. [CrossRef]
44. CCTV. Animal Kings: Lords of the Grassland. Available online: https://tv.cctv.com/2010/10/04/VIDE1355511938817391.shtml (accessed on 22 January 2024).
45. YouTube. Leopard vs. Wild Dogs vs Hyenas vs. Impala. Available online: https://www.youtube.com/watch?v=p49Vtl5KQJE (accessed on 22 January 2024).
46. Hart, P.E.; Nilsson, N.J.; Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
47. Nian, X.; Li, M.; Wang, H.; Gong, Y.; Xiong, H. Large-scale UAV swarm confrontation based on hierarchical attention actor-critic algorithm. *Appl. Intell.* **2024**, *54*, 3279–3294. [CrossRef]
48. Liu, H.; Li, Z.; Huang, K.; Wang, R.; Cheng, G.; Li, T. Evolutionary reinforcement learning algorithm for large-scale multi-agent cooperation and confrontation applications. *J. Supercomput.* **2024**, *80*, 2319–2346. [CrossRef]
49. De Zarzà, I.; de Curtò, J.; Roig, G.; Calafate, C.T. LLM Multimodal Traffic Accident Forecasting. *Sensors* **2023**, *23*, 9225. [CrossRef] [PubMed]
50. Jin, B.; Zhao, X.; Yuan, D. Attack–Defense Confrontation Analysis and Optimal Defense Strategy Selection Using Hybrid Game Theoretic Methods. *Symmetry* **2024**, *16*, 156. [CrossRef]
51. Liu, H.; Wu, K.; Huang, K.; Cheng, G.; Wang, R.; Liu, G. Optimization of large-scale UAV cluster confrontation game based on integrated evolution strategy. *Clust. Comput.* **2024**, *27*, 515–529. [CrossRef]

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.