drones

MDPI

Article

Task Allocation and Saturation Attack Approach for Unmanned Underwater Vehicles

Qiangqiang Chen [1,†], Baisheng Liu [2,†], Changdong Yu [3,*], Mingkai Yang [4] and Haonan Guo [5]

[1] Faculty of Science, Kunming University of Science and Technology, Kunming 650500, China; chenqiangqiang@stu.kust.edu.cn
[2] College of Software, Liaoning Technical University, Huludao 125000, China; 2120011013@stu.lntu.edu.cn
[3] College of Artificial Intelligence, Dalian Maritime University, Dalian 116026, China
[4] College of International Collaboration, Dalian Maritime University, Dalian 11602, China
[5] College of Business Administration, Liaoning Technical University, Huludao 125000, China; 472321195@stu.lntu.edu.cn
* Correspondence: ycd@dlmu.edu.cn
† These authors contributed equally to this work.

Abstract: In modern marine warfare, unmanned underwater vehicles (UUVs) have fast and efficient attack capabilities. However, existing research on UUV attack strategies is relatively limited, often ignoring the requirement for the effective allocation of different strategic value areas, which restricts its performance in the marine combat environment. To this end, this paper proposes an innovative UUV task allocation and saturation attack strategy. The strategy first divides the area according to the distribution density of enemy UUVs, and then reasonably allocates tasks according to the enemy's regional value and the attack capability of our UUVs. Our UUVs then sail to the enemy area and are evenly distributed in the encirclement to ensure accurate saturation attacks. In the task allocation link, the grey wolf optimizer is improved by introducing Logistic chaos mapping and differential evolution mechanism, which improves the search efficiency and allocation accuracy. At the same time, the combination of the optimal matching algorithm and Bezier curve dynamic path control ensures the accuracy and flexibility of a coordinated attack. The simulation experimental results show that the strategy shows high attack efficiency and practicality in marine combat scenarios, providing an effective solution for UUV attack tasks in complex marine environments.

Keywords: UUVs; task allocation; saturation attack; collaboration; swarm intelligence

Academic Editor: Oleg Yakimenko

Received: 28 December 2024
Revised: 31 January 2025
Accepted: 31 January 2025
Published: 4 February 2025

Citation: Chen, Q.; Liu, B.; Yu, C.; Yang, M.; Guo, H. Task Allocation and Saturation Attack Approach for Unmanned Underwater Vehicles. *Drones* **2025**, *9*, 115. https://doi.org/10.3390/drones9020115

1. Introduction

In recent years, with the advancement of maritime industries and artificial intelligence technologies, naval warfare has been shifting towards networking and intelligence. UUVs, as intelligent and autonomous maritime agents, have shown broad application potential [1–3]. Currently, as many researchers accelerate the study of unmanned system clusters, the field of UUV operations has seen rapid development [4–6]. Therefore, improving the adaptability and flexibility of UUVs to achieve collaborative combat purposes has become a research hotspot in the field of shipbuilding and ocean engineering. In particular, in the context of naval military operations, efficient task allocation and saturation attacks in complex environments have emerged as key research challenges.

In terms of task allocation, traditional methods for solving task planning include the Hungarian algorithm, centralized linear programming, etc. In recent years, with in-depth research on biological swarm intelligence and neural networks, scholars have begun to use these theories to study the field of multi-agent task allocation. For example, Chen et al. [7] proposed a genetic algorithm with genetic constraints (GAC) to address the inefficiency of traditional genetic algorithms in solving the target allocation problem. Zhuang et al. [8] introduced an offline task allocation method that combines an improved genetic algorithm (GA), fast marching method (FMM), and self-attention mechanism (SAM). This algorithm reduces the total travel distance of UUVs and balances the task load across the UUVs. Xue et al. [9] proposed an exact algorithm based on the Hungarian algorithm to optimize the task allocation of multiple UUVs, which effectively minimizes the maximum task completion time and reduces the overall task duration, making it suitable for time-sensitive collaborative environments. Bi et al. [10] presented a cooperative task allocation method based on an improved Non-dominated Sorting Genetic Algorithm II (NSGA-II), enhancing the convergence speed and quality of collaborative task allocation for heterogeneous air–sea unmanned systems. Zhang et al. [11] proposed a method that integrates the nonlinear function fitting ability of deep neural networks with the decision-making control ability of reinforcement learning to solve multi-UUV task planning problems. However, reinforcement learning methods often require substantial computational resources and time, which may become a limiting factor in practical applications, particularly in resource-constrained or time-critical environments. Xia et al. [12] introduced an improved self-organizing map (ISOM) network model for multi-task allocation in UUVs. However, this method has limitations in reasonably allocating tasks with different priorities.

Saturation attacks aim to adopt a large number of weapons in a short period to exceed the defense capabilities of the enemy system, thereby achieving the destruction of enemy targets. However, research on saturation attacks involving multiple unmanned surface vehicles is relatively scarce. Insights from the field of robot and unmanned aerial vehicle (UAV) swarm systems could offer valuable references for the UUV domain. For example, Wu et al. [13] analyzed several attack strategies for UAV swarms in urban combat scenarios, including saturation attacks, precision attacks, and distributed attacks, to accomplish various combat tasks. Marzoughi et al. [14] investigated how pursuers can effectively encircle fugitives within a limited area when their mobility is restricted. This study focuses on the dynamic interaction between pursuers and fugitives, particularly under resource and environmental constraints, and proposes methods to optimize pursuit strategies for effective encirclement. Sun et al. [15] introduced the Apollonius Circle as the ultimate form of attack mode and considered obstacle avoidance. Qian et al. [16] proposed a method based on deep reinforcement learning for task allocation in UAV swarm saturation attacks, treating the task allocation problem as a Markov Decision Process (MDP) and presenting a policy gradient-based training algorithm to enhance learning speed. Wen et al.[17] studied how to calculate the minimum number of UAVs required to ensure the completion of encirclement tasks without predefined formations and designed a cooperative fence controller for tracking mobile targets. Zhang et al. [18] addressed the saturation attack problem for UAV swarms by investigating autonomous maneuvering strategies. They constructed an autonomous decision-making model based on a Distributed Partially Observable Markov Decision Process (POMDP) and utilized a recursive Multi-Agent Deep Deterministic Policy Gradient (MADDPG) for learning. The effect of a suicide saturation attack is optimized by designing global and local reward functions.

Inspired by the above, this paper proposes a novel task allocation and saturation attack method for multiple unmanned vehicles. We first introduce the improved Grey Wolf Optimizer [19] to solve the task allocation problem of UUVs and adopt a dynamic path control method based on the stepwise optimal matching algorithm integrated with Bezier curves to establish a saturation attack model. The final simulation experiment shows that our UUVs can efficiently complete the saturation attack mission against the enemy through reasonable task allocation and planning. In general, the contributions of this paper are as follows:

- We introduce chaos initialization and generate the initial population position through the Logistic chaos mapping function, which enhances the population diversity and global search capability.
- We adopted the differential evolution strategy into the Grey Wolf Optimizer. For individuals with lower fitness, the update strategy of the three positions of α, β and ω, and used in the grey wolf optimization algorithm. For individuals with higher fitness, the differential evolution strategy is used for updating, which enhances the local search capability.
- A Bezier curve dynamic path control strategy is employed to enable the UUV to produce a smooth and controllable trajectory, ensuring a smooth path and avoiding local path jitter when performing attack missions.

The rest of the paper is organized as follows: In Section 2, we present a basic scenario description of UUVs in a maritime environment. Then, we propose the task allocation method and coordinated saturation attack strategy for UUVs in Section 3 and Section 4, respectively. Moreover, Section 5 describes the implementation details and reports the simulation experiment results. Finally, a brief conclusion is given in Section 6.

2. Basic Scenario Description

This paper focuses on the scenario of swarm operations of UUVs in a maritime environment. In a specified area, our forces (blue) deploy M UUVs, while the enemy deploys N UUVs (red). The enemy's UUVs are initially located at certain positions within the sea area. By assessing the concentration of enemy UUVs, we partition the region into target zones. Upon detecting these zones, our UUVs will quickly form a formation and execute a saturation attack on the enemy's target area through task allocation. To enhance the challenge of the attack mission, multiple obstacles are introduced in the scene.

To ensure the efficient execution of the saturation attack, a new deployment strategy is adopted by our UUVs: they are deployed on both sides of the enemy, as shown in Figure 1. The advantage of this strategy lies in its ability to comprehensively block the enemy: by approaching from both sides, our UUVs can quickly encircle the enemy's target region, effectively sealing it off. This approach increases the probability of a successful attack, minimizes the potential damage to our forces due to excessive distance from the enemy, and ensures more precise attacks. It thus improves both the safety and flexibility of the operation.

Assume that both the enemy and our forces consist of homogeneous UUVs, meaning they maintain identical performance parameters. Each UUV participating in the saturation attack shares its velocity and position with other UUVs in real time. The size of the UUVs is neglected, and they are treated as point particles. Initially, the distances between all friendly and enemy units are greater than the radius of the saturation attack formation formed by our forces. The motion equation of the UUVs [20] can be expressed as follows:

$$\begin{cases} \dot{x}_i = u_i \cos \psi_i - v_i \sin \psi_i \\ \dot{y}_i = u_i \cos \psi_i + v_i \sin \psi_i \\ \dot{\psi}_i = r_i \end{cases} \tag{1}$$

where (x_i, y_i) denotes the position vector and ψ_i denotes the heading angle of the i-th UUV in the earth-fixed frame. Additionally, u_i, v_i, and r_i are the forward velocity (surge), transverse velocity (sway), and angular velocity in the yaw direction in the body-fixed frame, respectively, $V = \sqrt{u_i^2 + v_i^2}$ is the actual movement velocity of the i-th UUV.

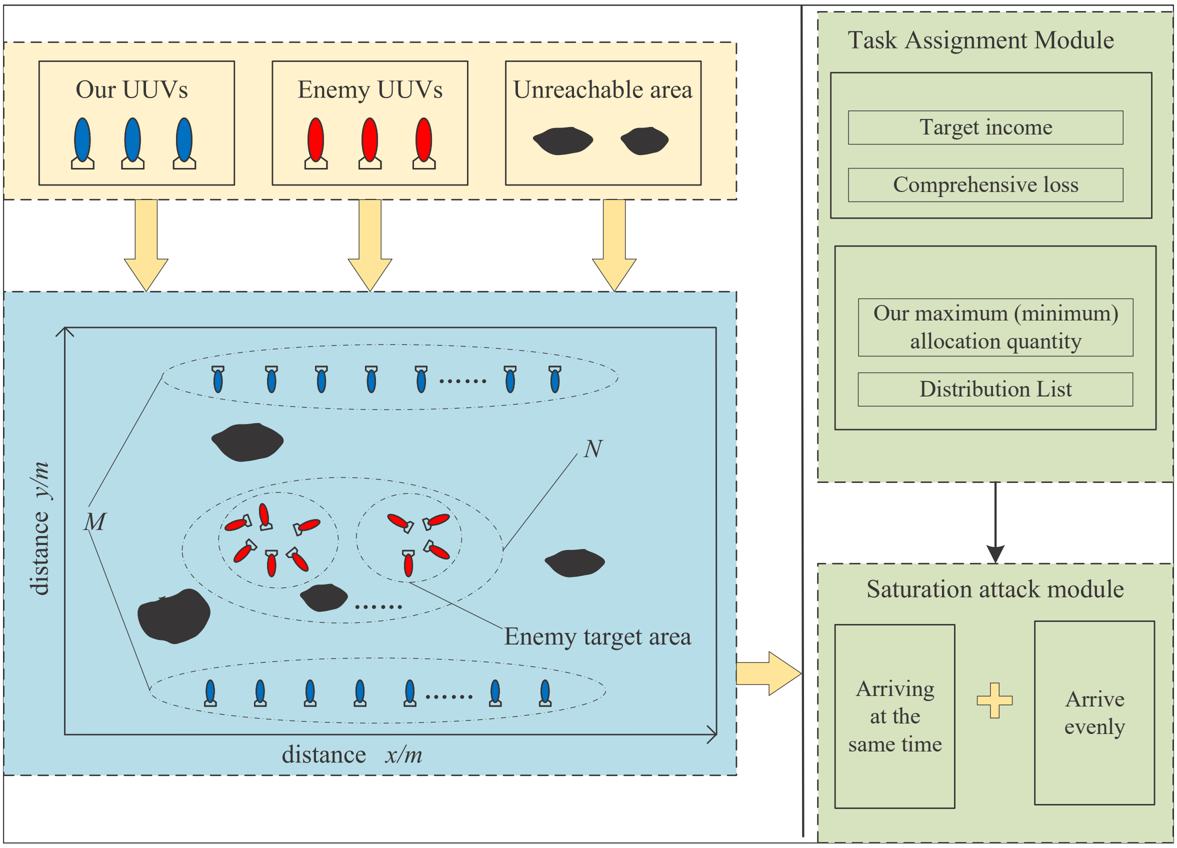


Figure 1. Schematic diagram of the overall framework of saturation attack for UUVs.

3. UUVs Task Allocation Method

The task allocation problem for UUVs can be viewed as a scenario in which multiple UUVs collaboratively perform multiple tasks within a specified area. In this process, multiple UUVs may cooperate to complete the same task. When our UUV detects an enemy target region, it quickly forms a formation to encircle the area. The allocation of tasks is influenced by factors such as the value, difficulty, and priority of the target area, as well as the capabilities, movement time, and execution costs of the UUVs. Further, if the enemy UUVs contained in a certain enemy area have the enemy's command role and the ability to cause potential great harm to us, then this area is considered to be a high-value area. Therefore, the objective is to develop an effective task allocation strategy that maximizes the overall benefit of our swarm or minimizes the total cost. In this context, inspired by the hunting behavior of wolf packs, this paper employs a wolf pack mechanism to allocate tasks to the UUVs, aiming to achieve an optimal solution.

To solve the UUVs task allocation problem, we build an accurate model to optimize the task allocation efficiency, this paper adopts a five-tuple $\{\mathrm{P}, \mathrm{U}, \ \mathrm{T}, \ \mathrm{F}, \mathrm{I}\}$ to represent the problem. Here, P denotes the set of enemy target areas, U denotes the set of our UUVs, T denotes the set of constraints, F denotes the set of objective functions, and I represents the maritime environment. To ensure a balanced distribution of tasks among the participating UUVs, the task allocation must consider the following constraints, costs, and benefits when assigning tasks to our fleet.

3.1. Constraint Condition

In the process of task allocation for UUV clusters, it is essential to establish appropriate constraints between UUVs to ensure efficiency and safety in subsequent operations. To achieve a saturation attack effect while avoiding over- or under-allocation of UUVs to a single target area, upper and lower limits on the number of UUVs assigned to each target area are necessary. This approach can enhance the overall efficiency and success rate of task allocation. For simplicity in describing the task allocation of the UUVs, each participating UUV is assigned a numerical identifier, facilitating the implementation of the task allocation strategy. The specific constraints are as follows:

Assume that the set of our UUVs assigned to a specific enemy area is denoted by $W = \{W_i, i = 1, 2, \ldots, m\}$. For each target area, segmented based on the density of enemy UUVs, the total number of our UUVs allocated should equal the total number of our UUVs, represented by M. The mathematical expression for this task allocation constraint is as follows:

$$W = \{W_i, i = 1, 2, \ldots, m\} \tag{2}$$

$$\sum_{i}^{m} \mathrm{num}(W_i) = M \tag{3}$$

$$X_i \in \{0, 1\}, i \in \{1, 2, 3, ..., M\} \tag{4}$$

$$\begin{cases} \mathrm{mum}(W_i) \geq S_{\min}(W) \\ \mathrm{num}(W_i) \leq S_{\max}(W) \end{cases}, \quad W_i \in W \tag{5}$$

$$\text{dList } = \left\{W_i \in W, P_j \in P \mid W_i \rightarrow P_j\right\}; i, j \in \{0, 1, 2, ..., m\} \tag{6}$$

In formula (2), specifically, we determine the number of our groups m based on the divided enemy areas and W_i represents the number of our UUVs deployed to an enemy target area. In formula (3), The sum of the number of UUVs in each group of our side is equal to the total number of our UUVs M. In formula (4), each UUV is only allowed to be assigned to a single enemy target area. X_i represents whether the i-th UUV is assigned to a target area (1 represents assigned, 0 represents unassigned, and M is our UUV quantity). In formula (5), $S_{\min}(W)$ and $S_{\max}(W)$ represent the minimum and maximum number of UUVs required in a certain enemy area, respectively. In formula (6), dList is the final allocation result and i is the number of a certain group of our UUVs, and j is the number of the enemy area.

3.2. Target Return

In the task allocation of UUVs, calculating the target payoff during attack operations is crucial as it serves as a key metric for evaluating the effectiveness of the allocation strategy. The target payoff includes not only the direct benefits of the mission but also factors in the efficiency of task completion and the optimal use of resources. The payoff assessment for our UUVs primarily depends on three factors: the enemy density, the intrinsic value of the enemy, and the payload capacity of our UUVs. Higher enemy density and intrinsic value elevate the target area's importance, leading to the allocation of more UUVs to concentrate force on that area, thereby maximizing the overall mission payoff. The payload capacity of our UUVs reflects their attack capability, determining the intensity of the missions they can undertake. A greater payload capacity enhances the UUV's attack power, providing a significant advantage in conducting saturation attacks on enemy targets. To accurately measure the target payoff achieved, the total payoff for assigning our i-th UUV to target j can be expressed as follows:

$$\begin{cases} \text{benefit}_{i}^{j} = k_d \times \text{ dens }_j + k_{\text{val}} \times \text{ val }_j + k_o \times \text{ load }_i \\ k_d + k_{\text{val}} + k_o = 1 \end{cases} \tag{7}$$

where the variable dens_j represents the enemy aggregation density, expressed in terms of quantity. For instance, when the number of target areas is 1, it signifies a low-value target area, while when the number is 5, it indicates a high-value target area. In addition,

k_d represents the density weight coefficient, which is used to assess the impact of the aggregation of enemy unmanned vehicles within a target area on the overall reward.val_j refers to the value of all enemy UUVs within the target area, while k_{val} denotes the target value weight coefficient, which reflects the influence of target value on task allocation. load_i represents the payload capacity of the i-th UUV, and k_o signifies the weight coefficient of hitting the target area.

3.3. Comprehensive Loss and Objective Function

In the task allocation of unmanned vehicles, the losses of our unmanned vehicles when executing the assigned target area can be mainly divided into range loss and attack loss, both of which have a significant impact on the allocation results. Range loss refers to the fuel consumption of our unmanned ship due to factors such as distance and complex paths when traveling to the target area. This loss is closely related to the distance between our UUV and the enemy's UUV. The farther the distance, the higher the range loss. Attack losses are the inevitable losses caused by the enemy's counterattack when our unmanned vessel approaches and attacks the target area. This loss usually depends on factors such as the enemy's ability to counterattack. The specific cost formula and constraints can be expressed as follows:

$$\begin{cases} \text{cost}_i = k_{dis} f\left(\text{ dis }_{i,j}\right) + k_s f(\text{damage}) \\ f\left(dis_i^j\right) = \sqrt{\left(x_i^{usv} - x_i^P\right)^2 + \left(y_i^{usv} - y_j^P\right)^2} \\ k_{dis} + k_s = 1 \end{cases} \tag{8}$$

where $f\left(dis_i^j\right)$ represents the Euclidean distance between the center coordinates of our i-th UUV and enemy area j, $f\left(dis_{i,j}\right)$ represents the weight coefficient of range loss, $f(\text{damage})$ represents the enemy's killing capability, and k_{dis}, k_s represents the weight coefficients of range loss and attack loss, respectively.

We consider the need to reasonably and efficiently allocate the number of our unmanned vehicles to each enemy area, laying the foundation for subsequent saturation attacks. Therefore, in the process of allocating our unmanned vehicles, the allocation strategy must consider multiple factors, one is the loss value, and the other is the benefit value. The reasonable allocation of our unmanned vehicles must not only ensure the successful implementation of the encirclement but also minimize the cost of our UUVs in the process of approaching the target area. The specific description of the function is as follows:

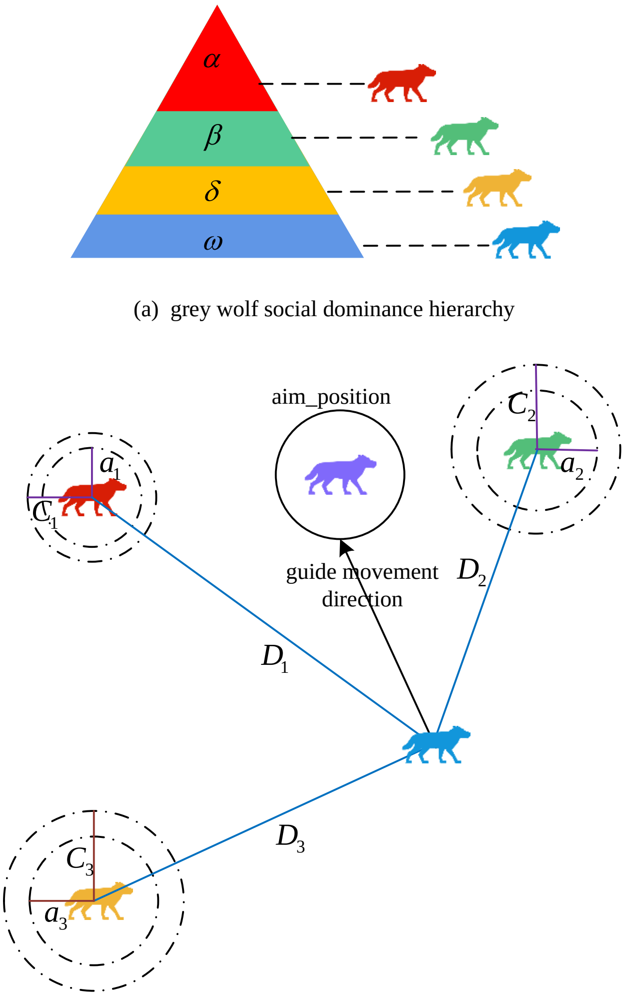
$$f = \sum_{i=1}^{U} \sum_{j=1}^{P} \left(\text{ benefit }_i^j - \text{ cost }_i\right) \tag{9}$$

In the formula benefit _i^j represents the benefit value generated by our i-th UUV in the process of attacking target area j, and cost_i represents the loss value.

3.4. Improved Grey Wolf Optimizer(IGWO)

3.4.1. Grey Wolf Optimizer

Grey Wolf Optimizer [19] is a swarm intelligence optimization algorithm inspired by wolf hunting behavior. The algorithm mainly imitates the social hierarchy and hunting strategy of grey wolves and is a bionic optimization algorithm (see Figure 2). It includes simple implementation, rapid convergence, and superior convergence outcomes, leading to its effective application in diverse fields for solving optimization issues [21]. The social structure of grey wolves consists of the following individuals of different levels: α wolf denotes the leader of the wolf pack and is responsible for major decision-making; β wolf assists α wolf in decision-making and may become a future α wolf; δ wolf follows the instructions of α wolf and β wolf, but has control over the subordinate ω wolves. ω wolves are the lowest individuals in the group and are responsible for obeying the instructions of all levels. In this algorithm, the solution in the optimization problem is represented as "wolf", and α wolf, β wolf, and δ wolf represent the current optimal solution, the second-best solution, and the third-best solution, respectively. In the algorithm, grey wolves hunt through three main behaviors:



(a) grey wolf social dominance hierarchy

(b) grey wolf hunting diagram

Figure 2. Schematic diagram of Grey Wolf Optimizer.

Tracking and surrounding prey, the process is described by the following formula:

$$\begin{aligned} \vec{D} &= \left| \vec{C} \cdot \vec{X}_{\text{prep}}\,(t) - \vec{X}(t) \right| \\ \vec{X}(t+1) &= \vec{X}_{\text{prep}}\,(t) - \vec{A} \cdot \vec{D} \end{aligned} \tag{10}$$

where t represents the number of iterations, $\vec{X}_{\text{prep}}\,(t)$ represents the location of the prey, and $\vec{X}(t)$ represents the current location of the wolf. In addition, the two coefficient vectors $\vec{A}$ and $\vec{C}$ are used to adjust the exploration and encirclement behaviors.

Chasing and trapping prey. During the hunting process, the grey wolf will not only surround the prey but also keep approaching the prey to get close to its position. In the algorithm, the current solutions of the three wolves α, β and δ are used to guide the grey wolves ω to update their position. The description formula of the whole process is as follows:

$$\begin{cases} \vec{D}_1 = \left|\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}\right| \\ \vec{D}_2 = \left|\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}\right| \\ \vec{D}_3 = \left|\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}\right| \end{cases} \tag{11}$$

$$\begin{cases} \vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_1 \\ \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_2 \\ \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_3 \end{cases} \tag{12}$$

Here, $\vec{X}_\alpha$, $\vec{X}_\beta$ and $\vec{X}_\delta$ represent the current positions of the three wolves α, β and δ, respectively. $\vec{D}_1$, $\vec{D}_2$, $\vec{D}_3$ represent the positions of the three wolves α, β and δ from the grey wolf, respectively.

Attack prey. Next, the positions of all grey wolves are updated using the following formula, and the algorithm gradually approaches the optimal solution.

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3} \tag{13}$$

3.4.2. Improved Grey Wolf Optimizer

Although the Grey Wolf Optimizer shows good convergence performance and searchability, its ability to obtain the global optimal solution for solving complex task allocation problems still needs to be improved. Based on this, we improve the GWO algorithm from two aspects according to [22]. The specific process of the improved Grey Wolf Optimizer (IGWO) is shown in Figure 3. We first replace the random initialization in the GWO algorithm with a Logistic chaotic initialization mechanism. Chaotic mapping is a nonlinear, quasi-random deterministic bounded system that exhibits neither periodicity nor convergence and is highly sensitive to initial parameters and conditions. The characteristics of chaotic systems—such as regularity, ergodicity, unpredictability, and sensitivity to initial conditions—enable the generation of 'quasi-random' sequences. Compared to traditional random number generation methods, chaotic sequences often lead to enhanced search performance, especially in global optimization problems, and offer the following advantages: (a) Chaotic sequences can cover the entire search space, ensuring no region is overlooked; (b) they help avoid local optima and facilitate exploration of a broader solution space; (c) they partially retain optimization directionality, aiding in faster convergence. In optimization algorithms, common chaotic mapping methods include the Logistic and Circle maps. The Logistic chaotic map, in particular, is a classical nonlinear map widely employed in population initialization and search space exploration within optimization algorithms. The description formula of Logistic mapping is as follows:

$$x_{n+1} = r \cdot x_n(1 - x_n) \tag{14}$$

where n is the length of the chaotic sequence to be generated, r is the control parameter of the Logistic mapping, and the common chaotic value range is $2 < r < 4$. Figure 4 shows a schematic diagram of different initializations. It can be seen that the initial individuals in the traditional Grey Wolf Optimizer are unevenly distributed in the random initialization wolf pack algorithm, which affects the global search ability of the algorithm. The introduction of Logistic chaotic mapping can increase the diversity of the population as shown in Figure 4b, which can solve the problem that the initial individuals appear in local areas.

Secondly, we introduce the Differential Evolution (DE) strategy to balance global exploration and local development capabilities, thereby improving the overall optimization performance. In other words, during the grey wolf's position update process, if the individual fitness is low the position is still updated, and if the individual fitness is high the differential evolution strategy is used to update its position. The differential evolution algorithm is essentially a population-based global optimization algorithm. Through mutation operations and crossover operations between individuals in the population, new candidate solutions are generated to further improve the algorithm's search depth. Specifically, the differential evolution strategy operation process is as follows: For mutation operation, it is to perform differential operation between individuals in the population, introduce randomness, expand the search range, and improve the global exploration ability. This helps the grey wolf jump out of the local optimal solution area. The differential mutation is the core step of the differential evolution algorithm, which uses individuals in the current population to generate new individuals. The information between individuals is borrowed from each other to push the solution closer to the optimal solution. Specifically, three different individuals x_{r1}, x_{r2} and x_{r3} are randomly selected to generate the expression of the new mutation vector λ_i as follows:

$$\lambda_i = x_{r1} + F \cdot (x_{r2} - x_{r3}) \tag{15}$$

where F is the scaling factor with a value between 0 and 2.

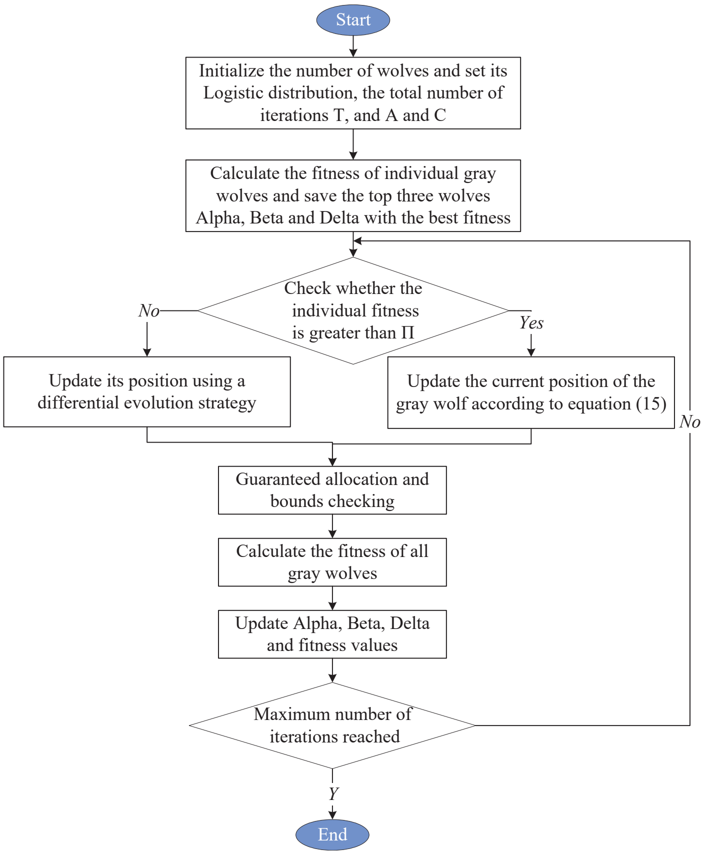

Start
Initialize the number of wolves and set its Logistic distribution, the total number of iterations T, and A and C
Calculate the fitness of individual gray wolves and save the top three wolves Alpha, Beta and Delta with the best fitness
Check whether the individual fitness is greater than Π
No
Yes
Update its position using a differential evolution strategy
Update the current position of the gray wolf according to equation (15)
No
Guaranteed allocation and bounds checking
Calculate the fitness of all gray wolves
Update Alpha, Beta, Delta and fitness values
Maximum number of iterations reached
Y
End

Figure 3. The specific process of the IGWO.

For the crossover operation, it generates new candidate solutions through crossover, so that the new solution can inherit the characteristics of multiple excellent individuals and enhance local development capabilities. The crossover operation ensures the diversity of the understanding space. The individuals generated by the mutation also need to cross with the current individuals to generate candidate solutions. The crossover operation determines whether to adopt the solution of the mutated individual based on the crossover probability (CR). The vector $\gamma_{i,j}$ after crossover is generated by the following description formula:

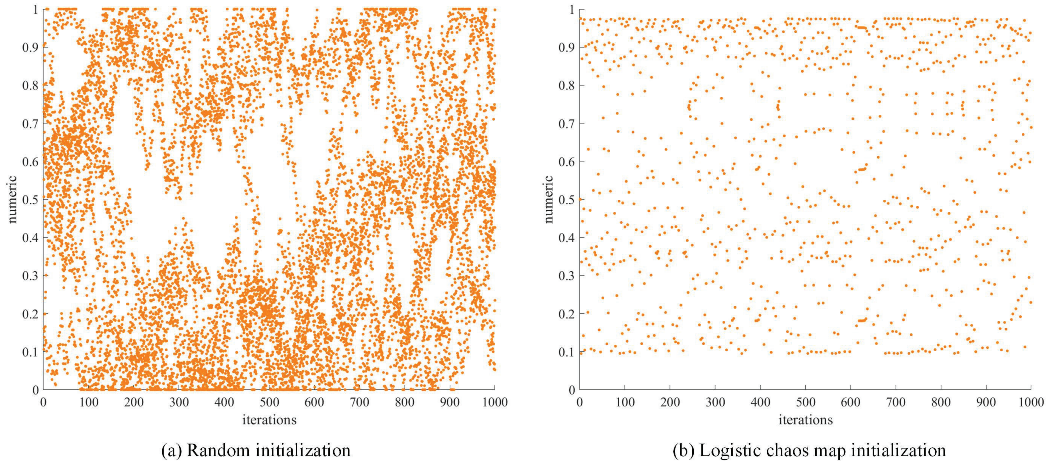
$$\gamma_{i,j} = \begin{cases} \lambda_{i,j} & \text{if rand } (0,1) \leq CR, \\ x_{i,j} & \text{otherwise} \end{cases} \tag{16}$$

where $CR \in (0,1)$ is the crossover probability, which is used to control the mixing ratio of the mutation vector and the original solution, and $\gamma_{i,j}$ is the value of the test vector generated after crossover in the jth dimension. In addition, $\lambda_{i,j}$ represents the value of the mutation vector in the jth dimension, and $x_{i,j}$ represents the value of the original individual in the jth dimension.

In order to better describe the distribution relationship between our unmanned vehicles cluster and the enemy area, this paper uses an $M \times P$ dimensional matrix to describe the specific distribution relationship. The corresponding distribution matrix is expressed as follows:

$$X = \begin{bmatrix} X_{1,1} & \dots & X_{1,p} \\ \dots & X_{i,j} & \dots \\ X_{m,1} & \dots & X_{m,p} \end{bmatrix} \tag{17}$$

where M represents the number of our UUVs and U_i is its number, P represents the number of enemy areas and j is its number. We define $X_{i,j} = 1$ when U_i is assigned to P_j, and 0 otherwise.



(a) Random initialization

(b) Logistic chaos map initialization

Figure 4. Schematic diagram of different initializations of the GWO algorithm.

4. Coordinated Saturation Attack Against UUVs

For a saturation attack by multiple UUVs, each UUV must carry out effective task allocation and coordinated control. In this way, each UUV is able to arrive at the target area synchronously, form a siege and carry out a continuous attack. This paper presents a method that integrates a stepwise optimal matching allocation algorithm with Bezier curve-based dynamic control. This approach enables saturation attacks even in complex environments. By ensuring precise path control and position adjustments, our UUVs can accurately reach predetermined encirclement points, thereby achieving the intended saturation attack effect. As Figure 5 is a flow chart.

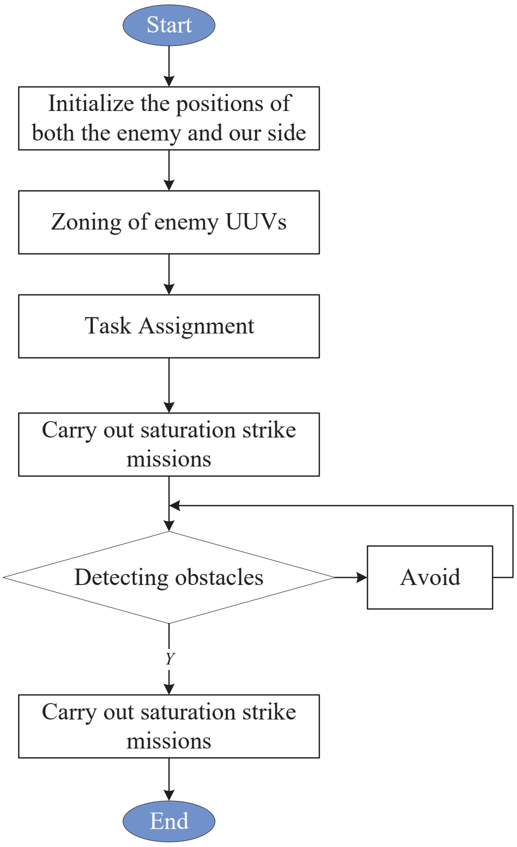

Start
Initialize the positions of both the enemy and our side
Zoning of enemy UUVs
Task Assignment
Carry out saturation strike missions
Detecting obstacles
Avoid
Y
Carry out saturation strike missions
End

Figure 5. Saturation strike flow chart.

4.1. Coordinate Calculation of Encirclement Points

In order to effectively encircle the enemy UUVs in a saturation attack mission, each UUV needs to reach its own predetermined encirclement point to implement the attack strategy (see Figure 6). The position of each encirclement point in the Cartesian coordinate system $\left(x_i^{(\delta)}, y_i^{(\delta)}\right)$ is given by the following formula:

$$\theta_i = \frac{2\pi(i-1)}{n}, i = 1, 2, \ldots, n \tag{18}$$

$$\begin{cases} x_i^{(\delta)} = R_m \cos(\theta_i) + x_c^{(\delta)} \\ y_i^{(\delta)} = R_m \sin(\theta_i) + y_c^{(\delta)} \end{cases} \tag{19}$$

$$U_i(t) \geq \mathrm{R}_m \tag{20}$$

In formula (18), n represents the number of our UUVs assigned to attack a certain enemy area. In formula (19), $\left(x_c^{(\delta)}, y_c^{(\delta)}\right)$ denotes the center position of the enemy area, R_m represents the distance between our UUVs and the center of the encirclement (i.e., the radius of the encirclement). In formula (20), $U_i(t)$ represents the distance between the i-th UUV and the center of the target area at time t. The constraints of this formula ensure that the UUV always maintains a distance from the center of the target area not less than R_m during navigation, thereby avoiding premature exposure or conflict with enemy UUVs. Until all our UUVs finally reach the encirclement at the same time and evenly, the distance will be equal to R_m to ensure that our UUVs maintain the best combat state in the attack mission to prevent the enemy UUVs from actively escaping after discovering our UUVs, and at the same time to prevent our UUVs from being seriously damaged by the enemy.

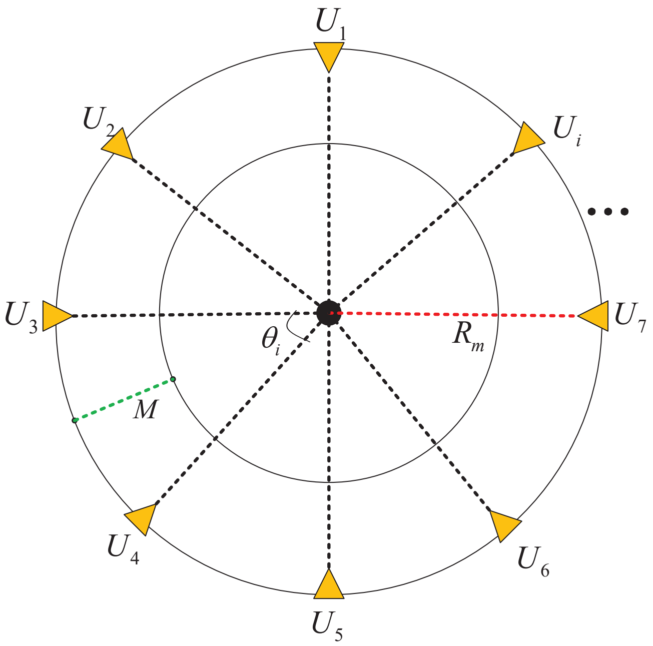


Figure 6. Schematic diagram of saturation attack encirclement.

4.2. Allocation of Encirclement Points Based on Optimal Matching Algorithm

After the encirclement points are determined, how to effectively allocate these encirclement points to our UUVs becomes a key issue. The stepwise optimal matching algorithm [23] is a simple and efficient task allocation method that is usually used to solve this type of problem. The algorithm achieves encirclement point allocation by minimizing the distance between the UUVs and the target encirclement point. The core idea is to approximate the global optimal solution by gradually selecting the local optimal solution in the process of task allocation. In the problem of allocating our UUVs and encirclement points, the Algorithm 1 performs task allocation according to the following steps:

Algorithm 1 Optimal matching algorithm

```
Require: $n_1$ is our UUVs heading to a certain enemy area
Ensure: UUV correspondence with assigned_goals$_{n_1 \times 1}$
Create a matrix assigned_goals $\in \mathbb{R}^{n_1 \times 2}$
Create a vector assigned_indices $\in \mathbb{R}^{n_1 \times 1}$
Command assigned_goals $= 0$ and assigned_indices $= 0$
$\left\{ C(i,j) \in \mathbb{R}^{p_1 \times x_1} \mid C(i,j) = \|USV_i - \text{target}_j\| \right\}$
for $i = 1$ to $n_1$ do
    $id_x = \arg\min_j \{C(i,j) \mid j \notin \text{assigned\_indices}\}$
    while $id_x \in$ assigned_indices do
        cost_matrix$(i, id_x) = \infty$
        $id_x = \arg\min_j \{C(i,j) \mid j \notin \text{assigned\_indices}\}$
    end while
    assigned_goals$(i,:)$ = target_positions$(id_x,:)$
    assigned_indices$(i) = id_x$
end for
return UUV$_i(i \in (1,2,\ldots,n_1)) \rightarrow$ assigned_goals$_{n_1 \times 1}$
```

4.3. Dynamic Control of Bezier Curves

This paper uses Bezier curves[24] to dynamically control our unmanned vehicles, which can help our unmanned vehicles achieve smooth path movement in complex environments so that they can reach the target deployment point faster and safer and then carry out saturation attacks. A Bezier curve is a parametric curve usually defined by a set of control points. Assuming the control points $P_0, P_1, \ldots, P_n$ are given, the mathematical representation of the Bezier curve is as follows:

$$B(t) = \sum_{k=0}^{n} P_k \cdot \binom{n}{k} (1-t)^{n-k} t^k, t \in [0,1] \tag{21}$$

Among them, P_k denotes the control point, n denotes the order of the Bezier curve, and t is the parameter of the curve. As shown in Figure 7, it is a fifth-order Bezier curve with a total of six control points [25]. Based on this, the control points in this paper are determined based on the initial position, target position, and target center point. The specific control point expression is as follows:

$$\begin{cases} \mathbf{d} = P_{goal} - P_{initial} \\ \hat{\mathbf{d}} = \frac{\mathbf{d}}{\|\mathbf{d}\|} \end{cases} \tag{22}$$

$$\begin{cases} P_1 = P_{initial} \\ P_2 = P_{initial} + scale_1 \cdot \hat{\mathbf{d}} \cdot \|P_{goal} - P_{center}\| \\ P_3 = P_{initial} + scale_2 \cdot \hat{\mathbf{d}} \cdot \|P_{goal} - P_{center}\| \\ P_4 = P_{goal} - scale_2 \cdot \hat{\mathbf{d}} \cdot \|P_{goal} - P_{center}\| \\ P_5 = P_{goal} - scale_2 \cdot \hat{\mathbf{d}} \cdot \|P_{goal} - P_{center}\| \\ P_6 = P_{goal} \end{cases} \tag{23}$$

In formula (22), the direction vector $\mathbf{d}$ is the vector from the initial position $P_{initial}$ to the target position P_{goal}, where $\|\mathbf{d}\|$ is the modulus of the vector $\mathbf{d}$ and $\hat{\mathbf{d}}$ is the normalized direction vector; in formula (23), the positions of the control points P_1 and P_6 are the initial position and the target position, respectively, and the remaining control points are determined based on the initial position $P_{initial}$, the target position P_{goal}, the direction vector $\mathbf{d}$, and the control coefficients $scale_1$ and $scale_2$. $scale_1$ controls a larger offset, which is usually used in the initial stage of the path. $scale_2$ controls a smaller offset, which is usually used in the middle stage of the path.

In the process of our UUV attack, this paper uses the envelope polygon [26] as the sea surface obstacle. The envelope polygon refers to the smallest convex polygon that encloses one or more point sets (the boundary of the obstacle). It is usually used in collision detection, path planning and other fields to ensure that the shape of the object can effectively represent the space it occupies. In the process of dynamic control of the Bezier curve, this paper also introduces an avoidance strategy to protect our UUVs. The specific avoidance strategy of this paper and the declaration of relevant variables during the driving process are as follows:

$$\text{grad} = \frac{B(t) - G}{\|B(t) - G\|} \tag{24}$$

$$B = B + k_1 \cdot \text{grad} \tag{25}$$

$$\begin{cases} \theta = \text{atan}\,2\left(B_y\left(t_j\right) - B_y\left(t_{j-1}\right), B_x\left(t_j\right) - B_x\left(t_{j-1}\right)\right) \\ \omega = k_2 \frac{\theta}{\Delta t} \quad, \Delta t = t_j - t_{j-1} \\ V = k_3 \cdot \frac{B\left(t_j\right) - B\left(t_{j-1}\right)}{\Delta t}, \Delta t = t_j - t_{j-1} \end{cases} \tag{26}$$

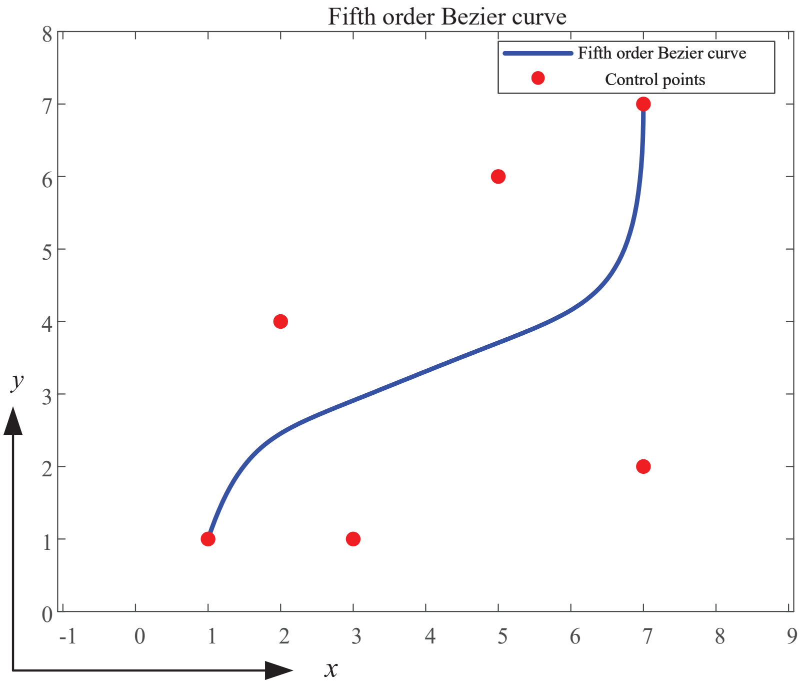


Figure 7. A fifth-order Bezier curve with a total of six control points.

Formula (22) represents the direction of the control curve B to the center of the encirclement at time t, grad is a unit vector, and G represents the spatial range of the obstacle; formula (23) updates the control curve B to drive our UUVs to a safe distance from the obstacle, where k_1 is a parameter for setting the safe distance between the control curve B and the obstacle. In the case of avoiding collision or approaching obstacles, it provides a buffer zone to ensure that our UUVs will not intersect or get too close to the obstacle during motion control. Formula (24) represents the changes in the movement velocity v and angular velocity w of our respective UUVs in the process of driving from the starting position to their respective encirclement points, where k_2 and k_3 are the proportional coefficients for adjusting the movement velocity and angular velocity, both of which are taken as 2 in this paper.

5. Simulation Experiments

In this section, we perform a series of simulation experiments to verify the effectiveness of the proposed method. First, we introduce the execution details of the simulation experiment and then report the task allocation results of the improved Grey Wolf Optimizer for UUVs. Finally, we present the saturation attack results based on Bezier curve control.

5.1. Implementation Details

In the simulation experiment, Windows 10 is used as the operating system and MATLAB 2023a is used as the simulation tool. The hardware platform is Intel(R) Core(TM) i5−1135G7 processor with a main frequency of 2.40GHz and a memory of 16GB. The simulation experiment takes the USVs as the simulation object and selects $x \in [0, 500], y \in [0, 800]$, where x and y represent the length and width of the sea area respectively. 7 enemy USVs and 8 of our USVs lined up on both sides of the coast are randomly set in the sea area, and 3 reefs are set (inaccessible areas). In the simulation test, each enemy USV has no special characteristics and can be detected and attacked by any of our USVs. To make the detection result more accurate, it is assumed that $S_{\min}(W) = 3$, $S_{\max}(W) = 5$, that is, each enemy area requires at least 3 or at most 5 of our USVs to carry out a saturation attack on it. The weights of the various coefficients of gain and loss are: k_d is 0.3, k_{val} is 0.3, k_o is 0.4, k_{dis} is 0.6, and k_s is 0.4. The specific parameters are presented in Table 1.

Table 1. Specific parameter settings for saturation attack scenarios.

Parameter	Numerical Value
Sea surface area	500× 800
Number of enemy UUVs	7
Number of our UU value of the (a) GVs	8
Enemy UUVs Positions value of the (a) G	A(275,250), B(240,285), C(205,250), D(240,215), E(625,200), F(587,221), G(587 ,178)
Enemy lethality	A:2.5, B:2.5, C:2.5, D:2.5, E:2, F:2, G:2
Value of the enemy	A:2, B:2, C:2, D:2, E:1.5, F:1.5, G:1.5
Our attack capability	U_1: 1, U_2: 1, U_3: 2, U_4: 2, U_5: 3, U_6: 3, U_7: 3.5, U_8: 3.5
Our UUVs position	U_1: (100,450), U_2: (100,60), U_3: (300,450), U_4:(100,60) U_5: (500,450), U_6:(500,50), U_7:(700,450), U_8: (720,50)

In addition, Figure 8 shows a schematic diagram of the distribution of the enemy and our UUVs at sea. The enemy area is divided into the α area with four high−value UUVs and the β area with three low−value UUVs. Additionally, our UUVs are distributed in the surrounding sea areas to achieve the purpose of a saturation attack.

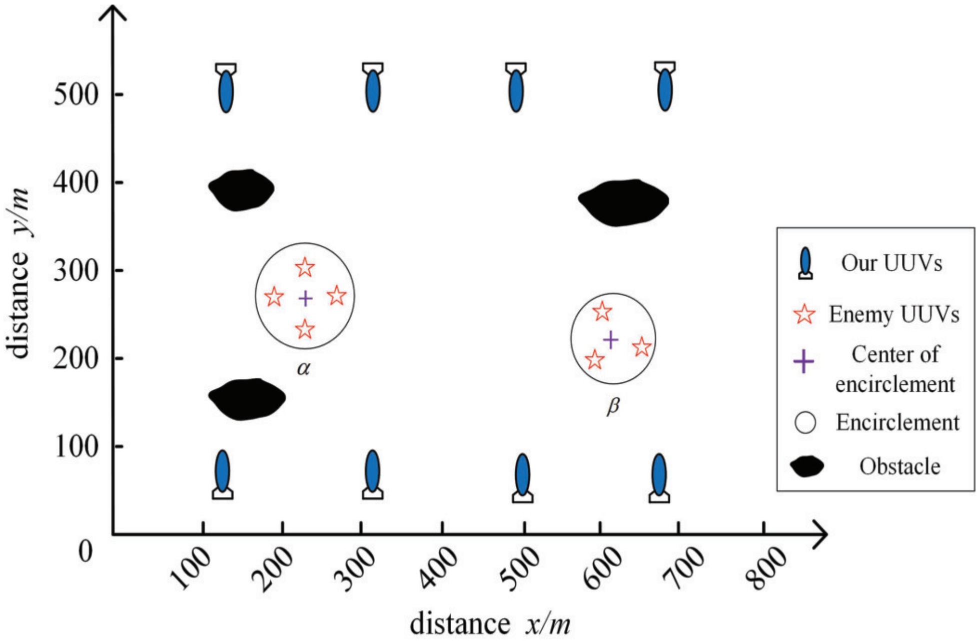


Figure 8. Schematic diagram of the distribution of the enemy and our UUVs at sea.

5.2. Task Assignment Result

We use Genetic Algorithm (GA), GWO and IGWO algorithms for the simulation experiments. In the optimization algorithm, the fitness curve directly reflects the dynamic changes of the algorithm during the search process. The fitness curves of the three algorithms used in this paper are shown in Figure 9, so as to more intuitively compare their optimization performance. The ultimate goal of optimization is to achieve task allocation at the minimum cost. Therefore, this paper directly uses the objective function as the fitness value to clearly measure the "pros and cons" of each solution. As shown in Figure 9, the optimal fitness of the three algorithms is 759.9, 661.63 and 608.3, respectively. Considering that the genetic algorithm and the grey wolf optimizer are a kind of random optimization algorithm, we conducted 20 simulation experiments and averaged the experimental results to ensure the fairness of the comparison. Table 2 shows the results of 20 simulations and the average optimal fitness value. Note that the initial population size is 100 and the number of iterations of each simulation experiment is 1000. *T* represents the number of simulations, and fitness represents the optimal fitness value. It can be concluded that the IGWO algorithm we proposed converges faster and has stronger adaptability. This improvement can significantly improve our resource utilization and improve the efficiency and accuracy of task completion in UUV task allocation.

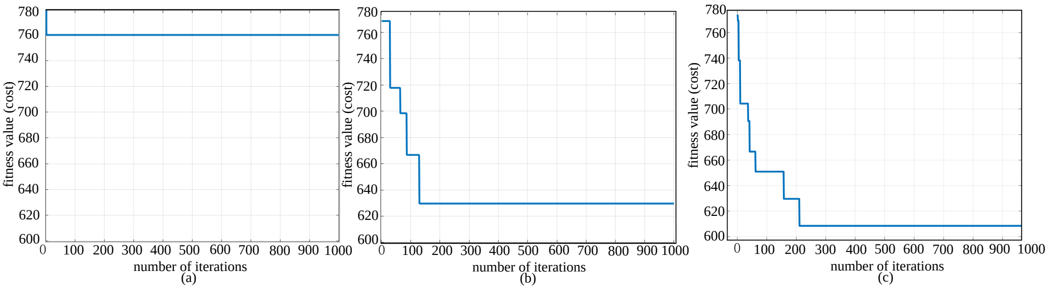


Figure 9. Fitness value of the (**a**) GA algorithm, (**b**) GWO algorithm and (**c**) IGWO algorithm.

Table 2. Results of 20 simulations and the average optimal fitness value

					GA algorithm						
T	1	2	3	4	5	6	7	8	9	10	20 times average
fitness	760.0	760.0	760.0	760.0	760.0	760.0	760.0	760.0	760.0	760.0	
T	11	12	13	14	15	16	17	18	19	20	760.0
fitness	760.0	760.0	760.0	760.0	760.0	760.0	760.0	760.0	760.0	760.0	
					GWO algorithm						
T	1	2	3	4	5	6	7	8	9	10	20 times average
fitness	679.3	629.7	659.5	646.0	716.9	613.2	608.3	675.8	650.9	726.2	
T	11	12	13	14	15	16	17	18	19	20	661.63
fitness	656.1	660.9	679.3	616.7	700.4	676.0	664.1	635.6	686.7	651.0	
					IGWO algorithm						
T	1	2	3	4	5	6	7	8	9	10	20 times average
fitness	654.6	629.5	646.0	701.2	650.9	661.7	608.1	697.2	629.1	608.3	
T	11	12	13	14	15	16	17	18	19	20	640.34
fitness	608.4	645.0	654.7	628.7	602.3	641.8	626.2	645.1	654.7	613.3	

Through the data of 20 experiments with three algorithms, it can be seen that the GA is the worst, so we mainly compare GWO and IGWO here. In addition, from the perspective of mathematical statistics, we also conducted statistical analysis on the performance of the GWO and IGWO algorithms by using the Wilcoxon test and Cohen's d effect size analysis to comprehensively evaluate the differences between the two algorithms. Among them, the Wilcoxon test is a non−parametric test method used to compare whether there is a significant difference in the median of two independent samples; Cohen's d effect size analysis is a standardized effect size indicator used to measure the difference between two groups of data. The experimental results are shown in Table 3. We can conclude that through the Wilcoxon test, due to $p = 0.0186 < 0.05$, we reject the null hypothesis and believe that there is a significant difference in the median fitness value between the GWO and IGWO algorithms; through the Cohen's d effect size analysis d = 0.7133 > 0.7, which is a high effect size, further verifying the significant advantage of the IGWO algorithm in performance. In summary, the statistical analysis results show that the IGWO algorithm is significantly better than the GWO algorithm in global search capability and convergence performance. This result provides strong statistical support for the superiority of the IGWO algorithm.

Table 3. Performance statistical analysis results of GWO and IGWO algorithms.

Inspection Methods	Statistics	$p-$Value	Effect Size (Cohen's d)	In Conclusion
Wilcoxon test	U = 287.5	0.0186	–	There is a significant difference in the median
Cohen's d effect size analysis	–	–	0.7133	High effect size, IGWO has a significant advantage

Finally, the allocation results of our UUV attack targets are given in Table 4. According to the prediction of the IGWO algorithm, our UUVs numbered 1, 3, 5, 2, and 4 are assigned to the α area, and the rest were assigned to the β area. Considering the enemy's killing capability, more of our UUVs are allocated to high−value target areas, and fewer UUVs are allocated to low-value areas, so the allocation results meet the expected goals.

Table 4. The final allocation result (dList).

Enemy Area	Our Assigned UUVs
α area	U_1, U_3, U_5, U_2, U_4
β area	U_6, U_7, U_8

5.3. Dynamic Path Planning Results

In the dynamic control simulation experiment of the optimal matching algorithm and Bezier curve, this paper uses a fifth−order Bezier curve and six control points to generate the path, ensuring the high−precision tracking of the UUVs. As illustrated in Figure 10, enemy UUVs of different values gather in a certain area on the complex sea surface. Our UUVs are distributed around the area to carry out a saturation attack on the enemy. After the task is assigned, our unmanned vehicles set off to rush towards the enemy. For example, at t = 91 s, our UUVs effectively avoid obstacles on the sea and head towards the enemy area. At t = 20 s, our UUVs adjust their angles and speeds to form a trend and behavior of being evenly distributed in the encirclement. At the last moment, our UUVs arrive at the enemy encirclement evenly and simultaneously, successfully completing the saturation attack. In addition, Figure 11 presents the distance between each of our UUVs and the target area over time. It can be seen that the distance between each UUV and the target gradually decreases, and when it reaches the last moment (t = 100 s), each UUV can reach the target area at the same time to complete the saturation attack.

Figure 12 shows the trend of the movement velocity and angular velocity of each UUV. In the initial stage of the path, within 0−2 s, as shown in Figure 10a, the movement velocity of the UUV = 0, but the angular velocity is 0. At this time, each unmanned vehicle mainly adjusts its own heading angle to keep consistent with the target route. This dynamic heading angle adjustment lays the directional foundation for the smooth tracking of the subsequent path. Then, UUV1−UUV8 gradually adjust their speeds to quickly attack the enemy. Figure 10b–d shows that our unmanned vehicles corrects the path deviation in real-time according to the control algorithm. Taking UUV4 as an example, during the driving process it continuously adjusts the heading angle to avoid islands and reefs so that the unmanned vehicle moves smoothly along the Bezier curve, significantly improving the path tracking accuracy. Figure 10 displays the position distribution of the unmanned vehicle in four states (0 s, 54 s, 91 s, 100 s) from the beginning to the attack process, verifying the combination of path smoothness and precision control, and ensuring efficient navigation in complex marine environments. The experiments fully demonstrate that the strategy proposed in this paper can achieve efficient saturation strikes against unmanned vehicle clusters in complex environments.

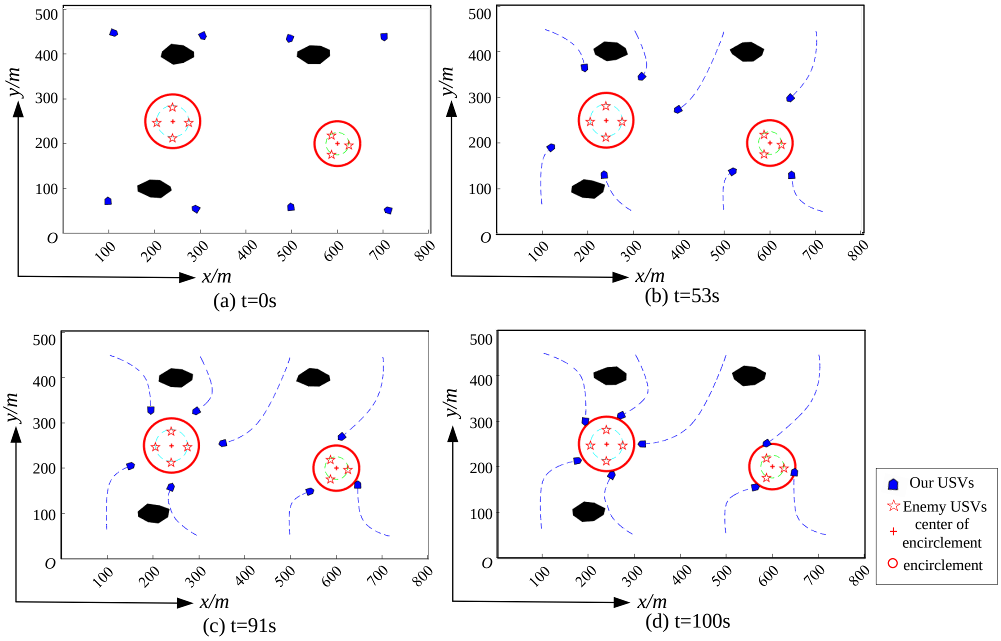


Figure 10. Schematic diagram of the dynamic saturation attack process of our UUVs.

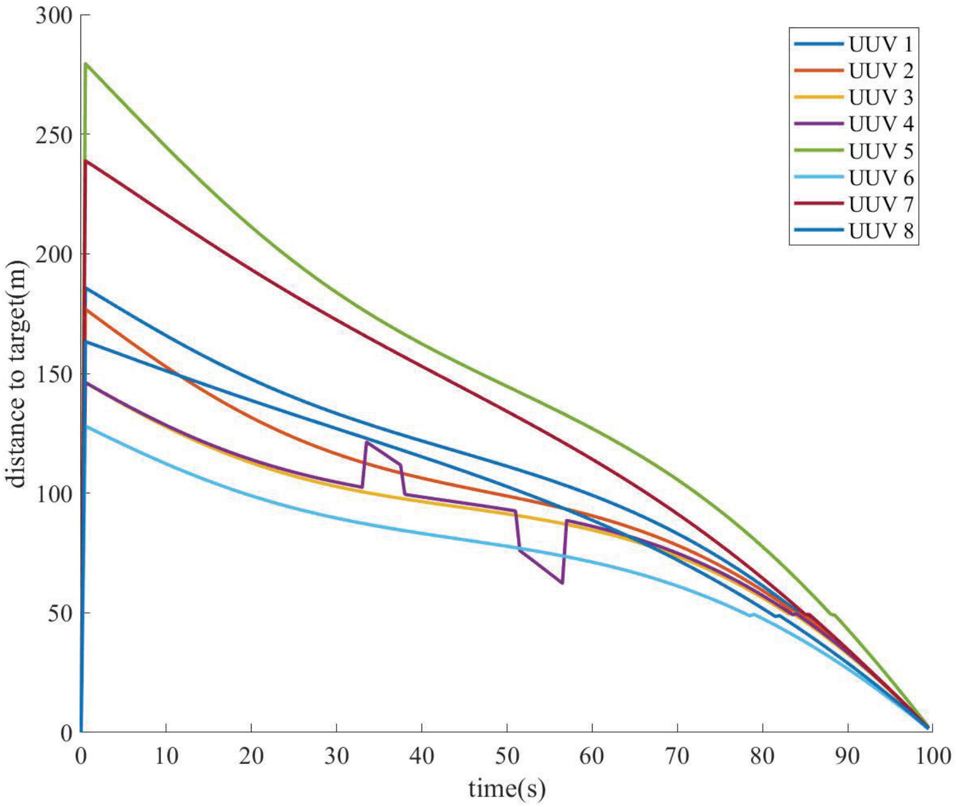
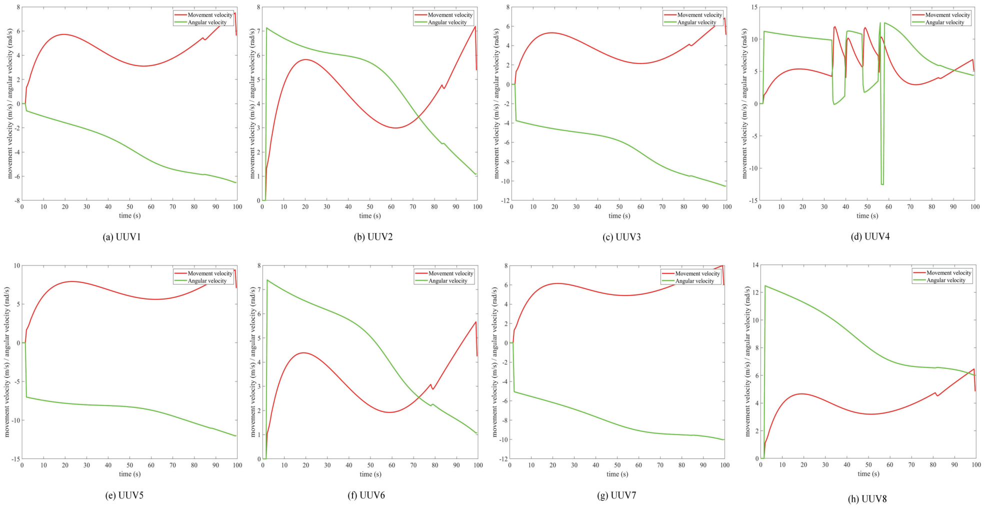


Figure 11. Our UUVs arrive at the encirclement points at the same time.



(a) UUV1 (b) UUV2 (c) UUV3 (d) UUV4

(e) UUV5 (f) UUV6 (g) UUV7 (h) UUV8

Figure 12. Schematic diagram of the dynamic changes in velocity and angle of different UUVs.

6. Conclusions

In this article, we present a novel saturation attack strategy aimed at addressing task allocation and saturation attack issues for UUVs in complex maritime environments. First, the enemy target area is divided according to the density of enemy UUVs, and task allocation is conducted based on the value of the enemy regions and the strike capabilities of friendly UUVs. An improved Grey Wolf Optimizer (GWO) is employed for task allocation, incorporating a Logistic chaotic map to enhance population diversity and differential evolution to optimize local searches, thus improving task allocation efficiency. The saturation attack strategy leverages a stepwise optimal assignment algorithm and dynamic path control via Bezier curves to ensure the UUVs form an encirclement and execute precise attacks, mitigating issues related to path jitter.

We then perform a series of simulation experiments to demonstrate the performance of the proposed strategy. Experimental results demonstrate that (1) The improved GWO outperforms the original algorithm in terms of fitness value, convergence speed, and the ability to escape local optima. These findings indicate that integrating chaotic initialization and differential evolution into GWO significantly enhances its global optimization capabilities and convergence performance, thereby improving the efficiency of task allocation. (2) By integrating the stepwise optimal matching allocation algorithm with dynamic path control using Bezier curves, our UUVs can all reach their designated encirclement points at the 100 second mark, demonstrating high saturation attack efficiency.

This study also has some limitations. (1) The study assumes that the distribution of obstacles and enemy UUVs in the marine environment is known and static. However, in actual combat environments, the marine environment may change dynamically, and the positions of obstacles and enemy UUVs may change over time. (2) The proposed task allocation and saturation attack strategies show high efficiency in simulation experiments, but in actual applications, UUVs may need to respond to emergencies in real—time, such as the sudden appearance or disappearance of enemy targets.

Future research could focus on the following aspects: (1) rapid response to emergent enemy targets by integrating real—time target recognition and tracking technologies to optimize task allocation strategies; (2) enhancing environmental awareness and autonomous decision—making capabilities by incorporating dynamic ocean environmental changes and deep reinforcement learning, thereby improving UUV adaptability and decision—making in complex combat environments. The saturation attack strategy proposed in this article is expected to play an important role in future maritime operations and provide strong technical support.

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

Funding: This work was funded by the National Natural Science Foundation of China (Project No. 52401362).

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

DURC Statement: Current research is limited to the technical exchange in the field of underwater unmanned systems, which is beneficial to further development and advancement of underwater unmanned system technology and does not pose a threat to public health or national security. The authors acknowledge the dual−use potential of the research involving saturation strike and confirm that all necessary precautions have been taken to prevent potential misuse. As an ethical responsibility, authors strictly adhere to relevant national and international laws about DURC. Authors advocate for responsible deployment, ethical considerations, regulatory compliance, and transparent reporting to mitigate misuse risks and foster beneficial outcomes.

Acknowledgments: All of the authors would like to thank the National Natural Science Foundation of China for their support. All individuals mentioned in this section have consented to be acknowledged.

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

Abbreviations

The following abbreviations are used in this manuscript:

UUV	unmanned underwater vehicle
GWO	Grey Wolf Optimizer
IGWO	Improved Grey Wolf Optimizer

References

1. Lin, M.; Zhang, Z.; Pang, Y.; Lin, H.; Ji, Q. Underactuated USV path following mechanism based on the cascade method. *Sci. Rep.* **2022**, *12*, 1461. https://doi.org/10.1038/s41598-022-05456-9.
2. Gu, G.; Lou, J.; Wan, H. A multi-strategy improved rime optimization algorithm for three-dimensional USV path planning and global optimization. *Sci. Rep.* **2024**, *14*, 12603. https://doi.org/10.1038/s41598-024-63188-4.
3. Liao, Y.; Chen, C.; Du, T.; Sun, J.; Xin, Y.; Zhai, Z.; Wang, B.; Li, Y.; Pang, S. Research on disturbance rejection motion control method of USV for UUV recovery. *J. Field Robot.* **2023**, *40*, 574–594. https://doi.org/10.1002/rob.22148.
4. Shi, B.; Guo, J.; Wang, C.; Su, Y.; Di, Y.; AbouOmar, M.S. Research on the visual image-based complexity perception method of autonomous navigation scenes for unmanned surface vehicles. *Sci. Rep.* **2022**, *12*, 10370. https://doi.org/10.1038/s41598-022-14355-y.
5. Liang, X.; Zhang, Y.; Yang, G. Platoon control design for unmanned surface vehicles subject to input delay. *Sci. Rep.* **2021**, *11*, 1481. https://doi.org/10.1038/s41598-020-80348-4.
6. Wu, C.; Zhu, G.; Liu, Y.; Li, F. Self-triggered adaptive neural control for USVs with sensor measurement sensitivity under deception attacks. *J. Field Robot.* **2025**, *42*, 153–168. https://doi.org/10.1002/rob.22400.
7. Chen, C.; Liang, X.; Zhang, Z.; Zheng, K.; Liu, D.; Yu, C.; Li, W. Cooperative target allocation for air-sea heterogeneous unmanned vehicles against saturation attacks. *J. Frankl. Inst.* **2024**, *361*, 1386–1402. https://doi.org/10.1016/j.jfranklin.2023.12.058.
8. Zhuang, J.; Long, L.; Zhang, L.; Zhang, Y.; Li, X. Research on task allocation for multi-type task of unmanned surface vehicles. *Ocean. Eng.* **2024**, *308*, 118321. https://doi.org/10.1016/j.oceaneng.2024.118321.
9. Xue, K.; Huang, Z.; Wang, P.; Xu, Z. An exact algorithm for task allocation of multiple unmanned surface vehicles with minimum task time. *J. Mar. Sci. Eng.* **2021**, *9*, 907. https://doi.org/10.3390/jmse9080907.
10. Bi, W.; Zhang, M.; Chen, H.; Zhang, A. Cooperative task allocation method for air-sea heterogeneous unmanned system with an application to ocean environment information monitoring. *Ocean. Eng.* **2024**, *309*, 118496. https://doi.org/10.1016/j.oceaneng.2024.118496.

11. Zhang, J.; Ren, J.; Cui, Y.; Fu, D.; Cong, J. Multi-USV Task Planning Method Based on Improved Deep Reinforcement Learning. *IEEE Internet Things J.* **2024**, *11*, 18549–18567. https://doi.org/10.1109/JIOT.2024.3363044.
12. Xia, G.; Sun, X.; Xia, X. Multiple task assignment and path planning of a multiple unmanned surface vehicles system based on improved self-organizing mapping and improved genetic algorithm. *J. Mar. Sci. Eng.* **2021**, *9*, 556. https://doi.org/10.3390/jmse9060556.
13. Wu, H.; Wu, K.; Yang, X.; Wen, H.; Li, H. *Analysis on the Operation Mode of UAV Swarm in Urban Combat*; Springer: Singapore, 2022; pp. 2781–2788. https://doi.org/10.1007/978-981-99-0479-2_257.
14. Marzoughi, A.; Savkin, A.V. Autonomous navigation of a team of unmanned surface vehicles for intercepting intruders on a region boundary. *Sensors* **2021**, *21*, 297. https://doi.org/10.3390/s21010297.
15. Sun, Z.; Sun, H.; Li, P.; Zou, J. Cooperative strategy for pursuit-evasion problem in the presence of static and dynamic obstacles. *Ocean. Eng.* **2023**, *279*, 114476. https://doi.org/10.1016/j.oceaneng.2023.114476.
16. Qian, F.; Su, K.; Liang, X.; Zhang, K. Task assignment for UAV swarm saturation attack: A deep reinforcement learning approach. *Electronics* **2023**, *12*, 1292. https://doi.org/10.3390/electronics12061292.
17. Wen, L.; Zhao, Y.; Zhen, Z.; Yan, C.; Xue, Y.; Zhao, Y. *Cooperative Fencing Control with UAV Swarm Based on Saturation Attack Mission*; Springer: Singapore, 2022; pp. 4125–4133. https://doi.org/10.1007/978-981-19-6613-2_402.
18. Zhang, T.; Zuo, Y.; Zhang, D.; Ding, F.; Huang, J.; Lin, Y. *Research on Autonomous Maneuvering Strategies of UAV Swarm for Saturation Attack*; IEEE: Piscataway, NJ, USA 2022; pp. 1605–1615. https://doi.org/10.1109/ICUS55513.2022.9986769.
19. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007.
20. Sarda, E.I.; Dhanak, M.R. A USV-Based Automated Launch and Recovery System for AUVs. *IEEE J. Ocean. Eng.* **2017**, *42*, 37–55. https://doi.org/10.1109/JOE.2016.2554679.
21. Liu, Y.; As'arry, A.; Hassan, M.K.; Hairuddin, A.A.; Mohamad, H. Review of the grey wolf optimization algorithm: Variants and applications. *Neural Comput. Appl.* **2024**, *36*, 2713–2735. https://doi.org/10.1007/s00521-023-09202-8.
22. Ibrahim, R.A.; Abd Elaziz, M.; Lu, S. Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization. *Expert Syst. Appl.* **2018**, *108*, 1–27. https://doi.org/10.1016/j.eswa.2018.04.028.
23. Shapero, S.A.; Hughes, H.; Tuuk, P. Adaptive semi-greedy search for multidimensional track assignment. In Proceedings of the 2016 19th International Conference on Information Fusion (FUSION), Heidelberg, Germany, 5–8 July 2016; pp. 409–415.
24. Eze, M.; Okunbor, C.; Aleburu, D.; Adekola, O.; Ramon, I.; Richard-Nnabu, N.; Avwokuruaye, O.; Olayiwola, A.; Yoro, R.; Solomon, E. Graphics Evolutionary Computations in Higher Order Parametric Bezier Curves. *Comput. Syst. Sci. Eng.* **2022**, *41*, 020835. https://doi.org/10.32604/csse.2022.020835.
25. Niu, T.; Wang, L.; Xu, Y.; Wang, J.; Wang, S. Quintic Bézier curve and numerical optimal solution based path planning approach in seismic exploration. *Control. Eng. Pract.* **2024**, *145*, 105855. https://doi.org/https://doi.org/10.1016/j.conengprac.2024.105855.
26. Nitulescu, M.; Ivanescu, M. *Method for Geometric Modeling of Obstacles in Mobile Robot Operating Scene*; IEEE: Piscataway, NJ, USA, 2024; pp. 515–520. https://doi.org/10.1109/ICSTCC62912.2024.10744687.

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.