Multimachine Collaborative Path Planning Method Based on A* Mechanism Connection Depth Neural Network Model

YIPENG ZHENG
College of Physics and Engineering, Chengdu Normal University, Chengdu, Sichuan 611130, China
e-mail: 239255064@qq.com

Abstract The naval battlefield is one of the main positions for future conflicts between major powers. The powerful naval battlefield target search capability is the last barrier to carry out maritime training and operations. At the same time, because of its complex and changeable environment and important strategic position, it has become the most difficult battlefield in joint search and rescue, the most central part. In order to reduce the time of target search in maritime battlefield, a real-time path planning method in the maritime battlefield based on deep reinforcement learning is proposed. The method proposed in this paper has advantages for path planning in multi-machine collaborative and can meet the requirements of real-time performance. First, a mathematical planning model for target search in the naval battlefield is constructed and mapped into a reinforcement learning model; then, based on the Rainbow deep reinforcement learning algorithm, the state vector, neural network structure and algorithm framework for target search path planning in the naval battlefield are designed with process. Finally, the feasibility and effectiveness of the proposed method are verified by experiments. The sequential exchange strategy and sequential insertion strategy of the A* algorithm are added to improve the optimal path obtained by the algorithm in each iteration. The improved algorithm is applied to the target path planning problem with time window. Through experiments on the Solomon data set, the optimal solution of the path before and after the improved algorithm is compared, and it is proved that the improved algorithm has better performance. In this way, the cooperative behavior of multi-agents can be realized, and the stability of reinforcement learning can be improved. The simulation results show that the length of the path searched by the proposed algorithm and the inflection point of the path are shortened by 19% and 58% respectively compared with the benchmark algorithm, which verifies the superiority of the proposed algorithm.

Index Terms Bezier curve, naval battlefield, target search, path planning.

I. INTRODUCTION
In recent years, unmanned systems have received extensive attention, and have made great progress in the fields of unmanned vehicles, unmanned aerial vehicles, unmanned subsurfaces, and unmanned ships [1]. Surface unmanned boat is a new type of intelligent unmanned surface platform, which can complete complex engineering and scientific research in complex water environment. Path planning is a mobile robot under the environmental constraints of obstacles, according to a given starting point and destination Apply algorithmic criteria to find an optimal or near-optimal, safe, unobstructed path [2]. Path planning mainly includes two parts: environment modeling and path planning algorithm.

Agricultural machinery automatic navigation technology is one of the key technologies in the implementation of precision agriculture, which can realize efficient and precise operation of agricultural machinery [3]. Nazarahari M Nazarahari M et al, uses a weight sum method to deal with the objectives; Davoodi M et al provided Pareto-optimal fronts. They have contributed to multi-objective path planning methods [4]–[5]. Multi-machine cooperative operation path planning is one of the key issues in the field of collaborative navigation [6]–[8]. Efficient, reasonable, safe and reliable path planning can improve the execution efficiency of the entire system and reduce the execution cost at the same
time [9]. Although domestic and foreign scholars have made many achievements in multi-machine collaborative path planning, and research methods are constantly innovating, there are still certain limitations [10]. Most of the research results are applied in the fields of multi-UAV collaboration, multi-robot collaboration and autonomous vehicles, and less research in the field of agricultural machinery collaborative navigation combined with actual agricultural application scenarios [11]. In addition, in actual farmland operations, path planning is an optimization problem under multiple constraints [12]. In addition to ensuring that the path cost is as small as possible, safe and collision-free, it must also comply with agricultural machinery kinematics and dynamics constraints [13]. Therefore, when using various algorithms to solve the path planning problem, it needs to be improved in combination with the actual operation scenario.

The targets to be rescued in the naval battlefield have the characteristics of short life cycle, wide search area, low probability of sighting, and complex drifting trajectory [14]. It is required that the search method must be able to respond quickly and support real-time planning [15]. Traditional solver methods, such as precise optimization algorithms, heuristic algorithms, and meta-heuristic algorithms, can only solve definite problems, but cannot deal with real-time changes in the search situation [16]. Reinforcement learning is a dynamic programming method that constantly interacts with the environment and adjusts its own strategy to cope with changes in the environment. Reinforcement learning continuously interacts with the environment to update the value in different states to realize the learning of the optimal policy [17]. However, when the state space increases, it faces the problem of exponential growth in storage requirements [18]. In recent years, the use of deep reinforcement learning algorithms to solve combinatorial optimization problems has become a hot topic. Deep reinforcement learning combines the characteristics of deep learning and reinforcement learning, which not only solves the “data hunger” problem of deep learning, but also solves the inability of reinforcement learning to store large the dilemma of scaled state spaces [19]. Current deep reinforcement learning methods are generally divided into two categories: value learning and policy learning [20].

Path planning algorithms mainly include local obstacle avoidance algorithms, geometric model-based algorithms and intelligent search algorithms [21]. Local obstacle avoidance algorithms include artificial potential field method and dynamic window method [22]. The artificial potential field method pushes the robot to avoid obstacles and move towards the end point by simulating the repulsion field generated by the obstacle and the gravitational field generated by the end point. The dynamic window method samples multiple sets of data in the velocity space, simulates the trajectory of the mobile robot in the next time period at these velocities, and evaluates the trajectory, and selects the velocity corresponding to the optimal trajectory to drive the robot. Algorithms based on geometric models mainly include Dijkstra algorithm and A* algorithm [23]. Dijkstra algorithm calculates the shortest path length from the starting point to all other points. It is a breadth-first search and has high space and time complexity. The A* algorithm is a heuristic algorithm and a depth-first algorithm. Intelligent search algorithms include ant colony algorithm [24], genetic algorithm [25], particle swarm algorithm, artificial bee colony calculation, etc., which use the rapid convergence characteristics of swarm intelligence to find the optimal solution. Path planning is an NP-hard problem, so many scholars use heuristic swarm intelligence algorithms to solve it [26]. The ant colony algorithm simulates the natural ants to find the shortest foraging path. Each ant will leave pheromone on the path when foraging, and the shorter the path, the higher the pheromone concentration. Ants prefer to walk along paths with higher pheromone concentrations and continue to leave new pheromones on the path [27]. Through the above-mentioned positive feedback effect, the final ant colony route converges to the shortest path with the highest pheromone concentration. Ant colony algorithm has the advantages of good robustness, strong global search ability, and convenient expression of environmental constraints, and is widely used in path planning problems. The traditional ant colony algorithm has the disadvantages of many iterations and low search efficiency. In order to solve these problems, Lee proposes a path planning method based on heterogeneous ant colonies, which improves the transition probability function and pheromone update rules of ant colonies, and can directly find the number of nodes without subsequent smoothing processing [28]. Fewer optimal paths; Zhang et al. proposed an intelligent ant colony path planning algorithm, which introduced a temporary weight matrix by eliminating the constraints of the table in the ant colony algorithm, avoiding the repeated selection of paths with small weights, and improving the improved the efficiency of the algorithm; LEE et al. changed the way the ant colony algorithm generated the initial population, shortened the time to find the optimal solution, and accelerated the convergence speed of the algorithm; Qi et al. [29] divided the ant colony algorithm into two stages to carry out the path search, effectively shortening the completion time of path planning. Wang et al. used the method of parameter migration to optimize the parameters of the ant colony algorithm, and used the prior knowledge to effectively solve the parameter selection problem in the path planning of the ant colony algorithm [30]. Zhang Qiang built a negative feedback channel in the ant colony algorithm, so that the update rate of the global pheromone and the local pheromone can be adaptively adjusted with the change of the convergence times, so that the convergence speed and the global search accuracy are coordinated and unified. Ant Colony Optimization (ACO) is an intelligent optimization algorithm that optimizes practical problems by simulating the foraging behavior of ants in nature [31]. It was first proposed by Italian scholar Marco. At the beginning of the search, the behavior of the ants is not guided by pheromone. Each ant randomly chooses a path
and releases pheromone along the way. Over time, the ants individually found food and left pheromones on their way. After multiple round trips, the information density left on the short path is higher in unit time. The ants will tend to choose the path with high concentration. In the subsequent search, more and more ants choose the short path, and finally get a solution of the shortest path. According to the algorithm idea, it can be known that since the ant colony algorithm does not need an initial solution, it can automatically adjust the optimal route in different environments, which reflects its positive feedback mechanism, strong adaptability, robustness and self-efficacy in combination with other algorithms [32]. It has the characteristics of strong adaptability [33].

In order to solve the problem of bias due to overestimation in the Q-learning algorithm, Guan and Zheng proposed a double Q network (double DQN) model, using two Q networks to learn action selection and action state value estimation respectively [34], to avoid overestimation of action value. In many deep reinforcement learning tasks based on visual perception, the magnitude of the time-value function is action independent. Based on this, Li et al. proposed a competitive Q-network structure (dueling DQN), which divides the estimation of the action state value into two branches, one is the state value function, the other is the advantage function, and the two branches are performed at the output layer [35]. Combining and summing to get the state value of each action can enable the agent to learn a more realistic value in the state environment without the influence of the action. Inspired by the path planning of mobile robots, this paper conducts an in-depth study of the multi-objective collaborative path planning method in the maritime battlefield [36].

This paper is oriented to the path planning problem of target search in the naval battlefield, considering the rapid response and real-time planning requirements of the problem, and researches the path planning method for target search in the naval battlefield based on deep reinforcement learning, aiming to achieve real-time, efficient and accurate search planning. Provide advanced method reference and algorithm support for improving target search path planning in my country’s naval battlefield, and smooth the global path based on the Bezier curve to seek a smooth and collision-free global optimal path. Path planning can use any direction and any distance grid as the next target grid method to improve search efficiency and reduce turning frequency.

II. METHOD AND PRINCIPLE

Multi-machine collaborative path planning needs to establish a mapping relationship between multiple operation plots. Under the premise of satisfying the actual operation constraints, with the goal of minimizing the scheduling cost and loss, a safe and collision-free optimal route is generated. Establish a scheduling model with short paths, high efficiency, safety and reliability [30].

A. MULTI-MACHINE COLLABORATIVE OPERATION PATH PLANNING

The operating environment is complex and dynamic, and there may be dynamic and static obstacles in the process of multi-machine cooperative operation. In order to solve the multi-machine cooperative operation path planning in the complex offshore operating environment, it is necessary to carry out local path planning under the premise of global path planning. The path adjustment ensures that the global planning is optimal when there is no conflict. When a conflict occurs, local planning is performed. After the conflict is resolved, it continues to travel with the global optimal. Therefore, this study divides the multi-machine cooperative operation path planning into two parts: global path planning and local dynamic obstacle avoidance. The overall framework is shown in Figure 1.

In the process of multi-machine cooperative operation path planning, first, an environment map model is established according to the initial environment information, and a global path planning is carried out based on the task planning scheme, and an optimal or better global path from the current position to the target plot is planned; Then, in the process, sensors are used to detect the surrounding environment information in real time to obtain dynamic obstacles or other target information; finally, according to the dynamic environment information, collision detection is performed and a decision-making plan is generated, and the global path is adjusted in real time. The task plot is the ultimate goal, avoiding the occurrence of local extreme points and ignoring the overall operation planning. At the same time, it detects the behavior of agricultural machinery and guides the agricultural machinery to safely travel to the designated position, so that it has a stronger ability to adapt to the environment [31].

B. TARGET SEARCH MODEL IN NAVAL BATTLEFIELD

First, a maintenance model of maritime search map is constructed, the search environment is formally described, and the search probability update mechanism is modeled. The objective function is modeled based on search theory and search cost. By constructing the above mathematical model, we quantified the degree of task progress and the degree of goal satisfaction in the search process [37]. This paper considers...
the advantages of UAV’s rapid response capability, wide-area search capability, long-distance communication capability, etc., and assumes that it is the main equipment for target search in the naval battlefield.

1) FORMAL DESCRIPTION OF MARITIME SEARCH MAPS

The task area E is divided into \( L \times L \) grids, as shown in Figure 2. Each grid is independent, and the coordinates of the center point of each grid are used as the position coordinates of the grid. Assuming that the initial prior information is known, each grid \((m, n)\) is assigned a certain initial value of target inclusion probability, that is, the probability that the target exists in the grid) \[38\], \((m, n)\) is in the range of \(m \in \{1, 2, 3 \ldots L\}, n \in \{1, 2, 3 \ldots L\}\).

This paper assumes that 100% of the area to be searched contains crash targets, so the inclusion probability of the entire area to be searched is 1. Assuming that the initial prior information is known and normalized, that is, it satisfies:

\[
\sum_{m=1, n=1}^{L} p_{mn} = 1 \quad (1)
\]

2) TARGET SEARCH PLANNING MODEL

The conventional search method for targets in the naval battlefield only determines the area to be searched, and then uses fixed search patterns such as parallel lines to plan the search path, resulting in a low search success rate \(POS\) (probability of success). For this reason, it is necessary to plan a search path in the area \(E\) to be searched based on the target existence probability model. Therefore, the planning model of target search in the naval battlefield is: within the limited range of the UAV, plan the search path scheme \(x\) of the UAV to maximize the probability of target discovery \(POS\) \[33\], as shown below:

\[
\max \ POS(x) = \sum_{m=1, n=1}^{L} p'_{mn}
\]

\[s.t.\ length(x) < L\] \( (2)\)

Among them, \(x(m, n)\) is whether the UAV searches the sub-area \(x(m, n)\), if it is, it is 1, otherwise it is 0; \(length(x) < L\) means that the length of the UAV search path is less than its range \(L\).

This paper adopts the method of centralized training and distributed execution. Since the input of the evaluator decision network during the training process is the environment state and the joint action of all agents, the evaluation value function of its output already contains the guidance information for multi-agent collaboration. In the distributed execution process, each agent-executor decision-making network does not need to communicate. When the number of training rounds is large enough, all coordination can be achieved through training without the need to establish a separate mechanism. However, in the future algorithm improvement, an agent communication mechanism can be added to further improve the synergy.

C. INFLECTION POINT BASED RASTER MAP

The grid map cuts the map into grids of the same size and connected to each other \[34\], and each grid corresponds to the corresponding location information. Three typical grid search methods are shown in Figure 3. This proposed method that can use any direction and any distance grid as the next target grid, which greatly improves the search efficiency and reduces the turning frequency. As shown in Figure 3, in a barrier-free environment, four search methods are used to search for the trajectory of the robot. Considering the simplicity of the map setting, the conventional ant colony algorithm \[8\] is used for path search. The key parameters of the ant colony algorithm are: The number of iterations \(K = 10\), the number of ants \(M = 20\), the pheromone initiation factor is 2; the heuristic function factor is 3, and the pheromone evaporation coefficient is 0.3. As can be seen from Figure 2, the 4-neighborhood and 4-direction search method can only go straight and turn right angles. The shortest path length from the starting point to the end point is 14, and the number of steps is 14; The length of the shortest path to the end point is 9.1, and the number of steps is 7;
the shortest path from the start point to the end point in the 24-neighborhood 16-direction search method is 8.7, and the number of steps is 5; the search method in this paper can take a shortcut to the end point, and the path length is 8.6, the number of steps is 1.

The increase in the number of neighborhoods means that the single-step search space increases. In order to reduce the calculation amount of the single-step search, this paper proposes a grid inflection point [35]. Considering that the shortest path must bypass the obstacle grid, only the neighborhood grid of the obstacle needs to be considered in the search process, that is, the inflection point must be the neighborhood grid of the obstacle. Use the sliding window method to find the inflection point, set the 22 sliding window to traverse the entire map, when there is only one obstacle grid in the window, set the other three grids as the inflection point; If there are 2 obstacle grids, and the positions are in the opposite corners, set the other pair of diagonal 2 grids as the inflection points. Several typical inflection points are shown in Figure 4: where black represents the obstacle grid, white represents the passable grid, and red represents the inflection point. When searching for a path, only the inflection point needs to be searched. By setting the inflection point, the workload of single-step search can be greatly reduced. Several typical grid inflection points are shown in Figure 5. The black is the obstacle, the white is the passable area, and the red is the inflection point.

**D. ANT COLONY ALGORITHM**

Ant colony algorithm was first proposed by Italian scholar Marco. Dorigo in the 1990s. Later, many scholars improved the ant colony algorithm and applied it to more combinatorial optimization problems, all of which showed relatively good results to varying degrees [36]. Assuming that there are m ants, each ant needs to traverse each city, and \( \tau_{ij}(t) \) is specified to represent the pheromone value between city i and city j at time t. The ants choose the next city to move according to the state transition probability, that is, \( P_{kij} \) represents the state transition probability of the k-th \( (k = 1, 2, \ldots, m) \) ant transferring from city i to city j, which is a combination of multiple parameters. get, the formula is as follows [37].

\[
P_{kij} = \begin{cases} \frac{[\tau_{is}(t)]^\alpha \times [\eta_{is}(t)]^\beta}{\sum_{s=allow_k} [\tau_{is}(t)]^\alpha \times [\eta_{is}(t)]^\beta} & s \in allow_k, j \notin j_{k-1} \\ 0, & s \notin allow_k \
\end{cases}
\]

\( \eta_{ij}(t) \) is a heuristic function defined according to its own self-adaptation, where:

\[
\eta_{ij} = \frac{1}{d_{ij}}
\]

The distance between cities i and j is represented by \( d_{ij} \), and \( d_{ij} \) is inversely proportional to \( \eta_{ij}(t) \). The expected magnitude of ants moving from city i to the next city j. After all the ants complete the task of visiting all the cities in one iteration, in order to avoid the accumulation of pheromone too much and cover the unexplored path inspiration information, the pheromone should be updated in time:

\[
\tau_{ij}(t + 1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta \tau_{ij}
\]

Among them, \( \rho \) represents the pheromone volatilization coefficient, which will directly affect the global search ability and convergence speed of the entire ant colony algorithm. \( (1 - \rho) \) is the residual factor of pheromone, which reflects the strength of the mutual influence between ants. The value range of \( \rho : \{0 < \rho < 1\} \), the initial \( \tau_{ij}(0) = 0 \). \( \tau_{ij} \) is the pheromone increment in this iteration, which can be expressed as:

\[
\Delta \tau_{ij} = \frac{Q}{L_k}
\]

\( Q \) is a constant, representing the fixed value of the pheromone carried by the ant itself, and \( L_k \) represents the path traveled by ant k in this iteration.
III. ALGORITHM IMPROVEMENTS AND MODELS

The A* algorithm is a heuristic search algorithm based on the Dijkstra algorithm, which can solve the shortest path in a static environment [37]. The A* algorithm is a search algorithm in the state space.

A. ALGORITHMIC PATH OPTIMIZATION

It first evaluates each search position to get the best position, and then searches from this position to the target, which can omit a large number of worthless search paths and improve the efficiency of the search [38], efficient. In the heuristic search, the evaluation of the location is very important, and different evaluations can have different effects. The evaluation function is:

\[ f(n) = g(n) + h(n) \]  

(7)

where \( f(n) \) —— the estimated cost from the initial node to the target node via node \( g(n) \) —— the actual cost from the initial node to the node \( h(n) \) —— from the node \( n \) to the target node

Estimated Cost of the Best Path The determination of the evaluation function is closely related to the actual situation, which will directly affect the results of the A* algorithm. In order to ensure the shortest path is found, the key lies in the selection of the evaluation function \( f(n) \) [39]. Considering the characteristics of the evaluation function of the A* algorithm, this study adopts the improved A* algorithm for global path planning. The evaluation function of the improved A* algorithm is:

\[ f(n) = g(n) + w(n)h(n) \]  

(8)

In the formula, \( w(n) \) —— the weight of the estimated cost \( h(n) \) The improved A* algorithm increases the weight \( w(n) \), and \( w(n) \) can affect the evaluation value. By setting \( w(n) \), the influence of \( h(n) \) on the A* algorithm in the path search process can be changed. The larger \( w(n) \) is, the closer it is to the BFS algorithm, and the smaller \( w(n) \) is, the closer it is to Dijkstra. algorithm. As shown in Figure 2, the specific implementation steps of the global path planning based on the improved A* algorithm are as follows: ① Parameter initialization. ② Establish a farmland grid map based on the initial environmental information. ③ Add the starting position of the agricultural machinery to the OpenList, and iterate in a loop to find the optimal path from the starting point of the agricultural machinery to the target operation point. ④ Expand child nodes: If there is an obstacle at the node to be expanded, ignore this point; if the node to be expanded is not in the OpenList, add it to the OpenList; For each expansion, put the points that meet the requirements obtained from the expansion into the OpenList as the candidate points, and continue to iterate until there are no points that can be expanded or the target operation point is found. ⑤ Path backtracking: After finding the target operation point, look for its parent node, and keep backtracking until the starting position, all parent nodes found also contain the planned path information of the agricultural machinery from the starting position to the target plot.

According to the index value of the parent node, you can view the position information of the parent node, and calculate the index value of the expected child node based on this. For example, the position information of the parent node is R, which means that the parent node is extended from the node on the right. If you want to extend a straight line, it is expected that the extended child node is the node to the left of the parent node. If the child node that needs to be expanded is the point where the straight line is expected to be expanded, skip the optimization; if the child node expected to be expanded is in the OpenList, compare the total path cost of the child node expected to be expanded and the child node that needs to be expanded. Corner optimization; if it is expected that the total cost of the extended child node is greater than the total cost of the original path to expand the child node, then the optimization is skipped; if the expected extended child node is not in the OpenList, indicating that the point
is an obstacle or beyond the boundary, then the optimization is skipped.

**B. ALGORITHM FRAMEWORK AND PROCESS OF DEEP REINFORCEMENT LEARNING ALGORITHM FOR TARGET SEARCH PATH PLANNING IN NAVAL BATTLEFIELD BASED ON RAINBOW**

Combined with the Rainbow algorithm and the basic idea of DQN, a deep reinforcement learning algorithm framework for target search path planning in the naval battlefield is designed, as shown in Figure 5. The specific process is as follows: Step 1: Build two neural networks with the same structure: prediction network \( \theta_p^{N2D} \) and target network \( \theta_T^{N2D} \) and initialize the parameters.

Step 2: According to the current observation state \( s_t \), leverage prediction network \( \theta_p^{N2D} \) to predict the value distribution of the UAV search action space, and use Equation (9) to calculate each the expected value of human-machine search actions. Step 3: According to the greedy strategy, select an action \( a_t \) from the drone search space.

Step 4: Generate a new state \( s_{t+1} \).

Step 5: If the round does not end, obtain the temporary reward \( r_t \) output by the environment, and if the round ends, obtain the temporary reward \( r_t \) output by the environment and the round reward \( R_t \).

Step 6: Update the current state to \( s_{t+1} \) and enter the prediction network

Step 7: When the round ends, recalculate the reward values for all actions in the round:

\[
r'_t = r_t + \lambda^{(T-t)} R_t
\]

where \( r_t \) is the final reward value of actions taken in each state, \( \lambda^{(T-t)} \) is a discount factor for rewards, which acts to allocate the round reward \( R_t \) more to later actions and less to earlier actions.

Step 8: Put \( [s_t, a_t, s_{t+1}, r_t] \) stored in memory. The above steps are the process of interaction between the agent and the environment. After a certain number of interactions, the agent trains the neural network according to the trajectory data stored in the memory, as shown in the following steps.

Step 9: Sample data from the memory bank using a priority experience replay strategy. A weight is assigned to each memory in the memory bank on the basis that those situations that cause the predicted value to deviate significantly from the target value should be the focus and training. Therefore, first calculate the predicted value of the prediction network and the target value of the target network:

\[
Q(s_t, a^t; \theta_p^{N2D}) = \left( \sum_{j=1}^{N} z_j \cdot p_j(s_t, a^t; \theta_p^{N2D}) \right)
\]

\[
Q(s_t, a^t; \theta_T^{N2D}) = r'_t + \gamma \cdot \max_{a} \left( \sum_{j=1}^{N} z_j \cdot p_j(s_{t+1}, a; \theta_T^{N2D}) \right)
\]

Based on this, calculate the degree of deviation between the predicted value and the target value:

\[
\delta = |Q(s_t, a^t; \theta_T^{N2D}) - Q(s_t, a^t; \theta_T^{N2D})|
\]

Then, calculate the probability of each record being selected, which is positively related to the absolute value of the deviation degree, satisfying:

\[
p_i \propto \delta_i + \varepsilon, \sum p_i = 1
\]

where \( \varepsilon \) is a minimum value and the probability of avoidance is 0. Then, probabilistically sample from the memory bank.

Step 10: Put the sampled data \( (s_t, a^t; s_{t+1}, r_t) \) into the prediction network separately \( \theta_T^{N2D} \) and target network \( \theta_T^{N2D} \). The predicted value distribution output of the prediction network.

**C. BIDIRECTIONAL A* ALGORITHM**

Because the marine environment is relatively wide, the distance between the selected target point and the starting point is relatively far, and the number of grids in the established grid map is also relatively large. Therefore, the time required for the one-way A* algorithm to find the path will become very long, and the space required by the storage node will become very large. In order to solve the problem of A* algorithm in large-scale grid number search, this paper makes certain improvements to the traditional A* algorithm, and performs path search at the target point and starting point at the same time [39]. The detailed steps of the bidirectional A* algorithm proposed in this paper are as follows.

Step 1 establishes two different OPEN lists, one is used to store the points that have not been checked in the search process from the target point to the starting point, and the other is used to store the points that have not been checked during the search process from the starting point to the target point. At the same time, two different CLOSE lists are concatenated as empty lists, one is used to store the points that have been traversed in the search process from the target point to the starting point, and the other is used to store the points that have been traversed during the search process from the starting point to the target point. passing point;

Step 2: Add the initial node S to the OPEN list that stores the points that have not been checked in the search process from the starting point to the target point, and add the destination node G to the OPEN list that stores the points that have not been checked during the search process from the target point to the starting point, in the OPEN list of past points;

Step 3: Search from the target point and the starting point to their respective directions at the same time, and the search method is consistent with the A* search;

Step 4 When the node of a certain OPEN list is in another CLOSE list, the final path [40] has been obtained at this time, the search is completed, and the search result is returned.
D. DYNAMIC WINDOW METHOD

If you want to drive without errors in a relatively wide and complex sea area, you need to plan and avoid the obstacles that have been marked in the map, and you need to monitor the driving environment in real time to avoid dynamic obstacles, and the local path planning is just right. This difficult problem can be dealt with, so the local path planning of USV (Unmanned Surface vessel) is a direction worth exploring. In this paper, DWA (Dynamic Window Algorithm) is used to complete the above-mentioned problems, and a relatively correct USV local path planning result can be obtained. Combined with the evaluation function in the improved A* algorithm mentioned above, the evaluation function is carried out correspondingly considering the level
of sea conditions. Improvements are made to determine the corresponding weighting coefficients to deal with a series of emergencies when the USV is traveling in the marine environment. Figure 8 is a schematic diagram of the DWA, the USV is represented by a black ship, the obstacles are represented by a gray rectangle, and the curve in the figure represents the navigation track of the USV.

As shown in Figure 8, the dotted line represents that the predicted trajectory will collide with the obstacle, so these unsafe planning results will be directly excluded during the path planning, and then the trajectory of the solid line will not collide with the obstacle. In the safety trajectory of the obstacle, the final path planning result is obtained according to the evaluation function of each trajectory.

IV. EXPERIMENTS AND EXAMPLES

The real environment of tracking device sailing on the sea surface is intricate and complex, which is reflected in static obstacles such as islands and large underwater stones, and dynamic obstacles such as large marine creatures that will affect the tracking device.

A. THE ESTABLISHMENT OF HYBRID PATH PLANNING SYSTEM

Both need to be considered when planning USV paths. First, the advantage of the A* algorithm is that it can accurately avoid static obstacles, but for dynamic obstacles, this algorithm cannot accurately plan a complete path. Second, the advantage of the DWA algorithm is that it can accurately avoid dynamic obstacles and some nearby static obstacles, but for the global static obstacles, this algorithm cannot accurately plan a complete path, the problem of getting stuck in local optimum mentioned in Section 3 may occur. In order to solve the above two problems, this paper proposes a fusion algorithm based on the combination of A* algorithm and DWA algorithm. This hybrid algorithm not only solves the problem of avoiding dynamic and static obstacles, but also improves the smoothness of the path while ensuring the optimal path, making it more suitable for the actual navigation of the USV, and has stronger practicability. Figure 11 shows the design framework of the hybrid path planning system proposed in this paper, which is mainly divided into a global path planning part and a local path planning part. The two are integrated with each other for path planning. The main steps are as follows. Step 1: Determine the distribution of dynamic obstacles in the global environment through the grid map obtained by the grid processing method of this paper through the actual ocean map, and use the bidirectional A* algorithm to obtain the global path of the USV; Step 2 Use external sensing sensors to sense in real time The data to update the local map, and combined with the global path in step 1 and the obtained local map to determine the local target point, use the DWA algorithm to avoid the dynamic obstacles to generate the local path planning path, in this way, the local optimal point is continuously updated to make The USV finally reaches the global target point, and finally gets the best planned path. In order to obtain a longer safe distance, the evaluation function is optimized in the DWA algorithm, and a coefficient related to the real-time condition level of the sea surface is added. When the real-time conditions of sea navigation are constantly changing, the relevant coefficients are adjusted in real time, so that the USV can sail on the sea environment relatively safely.

For the Multi-machine collaborative path planning method proposed in this paper, what should be done is to fully integrate the advantages of the A* algorithm and the DWA algorithm to ensure that the shortcomings of the two algorithms can be solved. 9 shows the implementation steps of the hybrid path planning algorithm proposed in this paper, which continuously integrates and updates the local optimal point and the global path, and finally obtains a safe planning path.

B. COMPARISON OF BASIC ANT COLONY ALGORITHM AND IMPROVED ANT COLONY ALGORITHM

In this experiment, the basic ant colony algorithm and the improved ant colony algorithm are compared, and examples of C1 type, R1 type and RC1 type are selected, and 4 examples are selected for each type, namely: C101, C102, C103, C104, R101, R102, R103, R104, RC101, RC102, RC103, RC104, a total of 12 instances [41]. Each example is tested 10 times, and the obtained optimal path lengths are averaged for comparison. The results are shown in Table 1.

From the comparison of the mean value of the longest path length in Table 1, it can be seen that the quality of the solution
of the basic A* algorithm has been significantly improved after the improvement, and then the mean result is shown in Figure 9.

In Figure 10, it is more intuitive to see that the optimal path obtained by the improved A* connection reinforcement learning algorithm is obviously better than the optimal path obtained by the basic ant colony algorithm. In this study, the optimal path solution obtained by the improved A* connection reinforcement learning algorithm and the basic A* algorithm on the c101 example was selected, and the path diagram of the optimal solution was drawn according to the horizontal and vertical coordinates in the example.

The path length, running time and number of corners before and after the corner optimization of the global path are shown in Table 3. After corner optimization of the global path, the number of corners is reduced from 9 to 6 when the path length is unchanged and the running time is similar. This effectively reduces the number of turns; similarly, after using the Bezier curve to smooth the path, the peaks at the corners are optimized to ensure the smooth running of the agricultural machinery; at the same time, the optimized running time is 0.832s, which preliminarily meets the smoothness of the agricultural machinery operation, and real-time requirements.

| Category       | Path length | Running time/s | Number of corners/pcs |
|----------------|-------------|----------------|-----------------------|
| Corners are not optimized corner optimization | 102 | 0.831 | 7 |

| TABLE 2. Path length, running time and number of corners before and after global path corner optimization. |
| Study name | Basic A* Algorithm | Improved A* - Reinforcement learning Algorithm |
|------------|--------------------|-----------------------------------------------|
| C101       | 1245.3             | 877.83                                        |
| C 102      | 1329.82            | 922.28                                        |
| C 103      | 1342.44            | 1034.55                                       |
| C 104      | 1182.36            | 956.19                                        |
| R101       | 2384.89            | 1952.6                                        |
| R 102      | 2108.81            | 1714.44                                       |
| R 103      | 1786.79            | 1440.46                                       |
| R 104      | 1367.16            | 1178.27                                       |
| RC101      | 2375.24            | 1977.53                                       |
| RC 102     | 2108.81            | 1816.12                                       |
| RC 103     | 1963.44            | 1598.55                                       |
| RC 104     | 1440.22            | 1320.32                                       |

FIGURE 10. Comparison of the mean values of optimal path lengths.

C. EXPERIMENTS ON THE NUMBER OF EFFECTIVE INFLECTION POINTS AND ONE-STEP SEARCH SPACE STATISTICS

In order to verify the scientificity of the method proposed in this paper using the shortest minimum distance as the heuristic value, statistical experiments of the shortest minimum distance and the Euclidean distance are carried out. Randomly generate 100 grid maps with a scale of 3030 and an obstacle ratio of 0.2 to 0.4, and randomly select the starting point and ending point from them (each map can take multiple sets of starting points and ending points), and find the shortest minimum path and Euclidean path between the starting point and the ending point. At the same time, the ant colony algorithm in this paper is used to calculate the shortest path from the starting point to the end point. The key parameters of the ant colony algorithm are: the number of iterations \( K = 50 \), the number of ants \( M = 50 \), the pheromone initiation factor \( = 3 \); the heuristic function factor \( = 6 \), the pheromone Evaporation coefficient \( = 0.3 \). The ratio of the shortest minimum distance and the Euclidean distance to the shortest distance obtained by the method in this paper is calculated. The path distance statistics are shown in Table 2, where the R value represents the ratio of the shortest minimum distance (Euclidean distance) to the shortest distance obtained by the ant colony algorithm in this paper.

It can be seen from Table 3 that even if the number of layers of the inflection point reaches 12, the ratio of the shortest...
minimum distance to the shortest distance is still less than 1.08, indicating that the shortest and minimum distance can largely reflect the distance from the start point to the end point; and the ratio of the Euclidean distance to the minimum distance As the number of layers increases, the ratio decreases to around 0.35, indicating that the gap between the two gradually increases. It can be seen from this that it is more accurate and reliable to use the shortest minimum distance as the heuristic value of the search node.

When the ants choose the next node, the long-term vision range can help the ants to plan the path better, but it also causes the single-step search space to be huge, which is not conducive to fast convergence. In order to solve this problem, based on the relationship between the shortest path and the obstacle grid, this paper proposes an inflection point grid, and further deletes the invalid inflection points through the branching operation; It reduces the number of grids that ants need to search for a single step. In order to intuitively understand the role of the branching operation in reducing the number of effective inflection points and reducing the single-step search space, the simulation calculation example is shown in Figure 11. Figure 11(a) is the original grid map, white indicates passable, black indicates obstacles, the upper left corner is the starting point, and the lower right corner is the end point. Figure 11(b) is the inflection point map, and the red grid is the inflection point. Figure 11(c) is the layered map generated by the branching operation. In the figure, the purple grid is the point of the L0 layer, that is, the end point, and the green grid is the point of the L1 layer, that is, all the points that are directly connected to the points of the L0 layer. Similarly, the dark blue is the point of the L3 layer, the light blue is the point of the L4 layer, the starting point is located in the L4 layer, and the red point in Figure 11(c) is the invalid inflection point that is eliminated. It can be seen that the number of effective inflection points is reduced by the branching operation. In the shortest path, the point to be reached in the next step must be the point in the layer where the current point is located or the point in the layer closer to the end point, rather than the point in the layer farther from the end point, which can significantly reduce the single step In the search space, taking the starting point as an example, the current single-step search only needs to search for the points of the L4 layer (light blue) and the points of the L3 layer (dark blue), and does not need to search for other valid inflection points. At the same time, there is a limit on whether the nodes can be directly reached, and a part of the inflection points to be searched can be further deleted, which can significantly reduce the pressure of single-step search.

**D. COMPARATIVE EXPERIMENT**

Table 4 shows the performance comparison of three rule-based methods in simulation scenarios of different difficulty in daytime 1, night time 2, foggy day 3 and rainy day 4.

| Environment | Literature [19] method | Literature [25] method | Our method |
|-------------|------------------------|------------------------|------------|
|             | Success rate | Sailing time step | Success rate | Sailing time step | Success rate | Sailing time step |
| Difficulty 1 | 82%         | 201.80               | 70%         | 89.8              | 90%         | 200.78             |
| Difficulty 2 | 75%         | 185.05               | 65%         | 73.05             | 81%         | 184.03             |
| Difficulty 3 | 55%         | 153.06               | 50%         | 41.06             | 71%         | 152.04             |
| Difficulty 4 | 48%         | 145.57               | 46%         | 33.57             | 54%         | 144.55             |
It can be seen from the simulation results that our method shows significant advantages in terms of security; in terms of efficiency, our method is significantly higher than the other two algorithms. However, as the difficulty of the simulation environment gradually increases, the success rates of the three algorithms all decline to a certain extent, and they are more sensitive to environmental conditions. However, the method proposed that Multi-machine collaborative path planning method still has advantages as a whole.

In order to compare the convergence of the improved method and the original method, 10 tests were performed, and every 100 rounds was used as a round to test the planning ability of the current model. The average success rate change curve is shown in Figure 12.

**E. LOSS FUNCTION AND REWARD FUNCTION CURVE**

The loss function curve is an important basis for evaluating whether the algorithm converges, and the reward function curve is the basis for evaluating the training effect of the algorithm. Therefore, plot the loss function and reward function curve of the training process, as shown in Figure 13 and Figure 14, respectively. For the convenience of display, Figure 13 only shows the convergence curve of 2000 times of training. It can be seen that the Rainbow algorithm can quickly converge in different voyage situations. Figure 14 shows the average reward value per 100 training sessions. It can be seen that the Rainbow algorithm can steadily improve the training effect during the 105 training sessions.

**V. CONCLUSION**

Aiming at the path planning problem in the grid graph environment, an ant colony path planning method based on the effective inflection point and the shortest minimum path is proposed. In the grid map, an infinite neighborhood search method is adopted, so that the robot can take a shortcut to any grid point that can be directly connected. At the same time, in order to reduce the search volume, the concept of effective inflection point is proposed, which greatly reduces the search space. At the same time, the shortest and smallest distance from the starting point to the end point is used to guide the pheromone update, which improves the iterative performance of the ant colony algorithm. It is found that the running time of the former algorithm is shortened by 19%, and the variables generated by the algorithm occupy 34% less memory. The innovations are as follows:

1. Considering factors such as path cost, path smoothing and collision detection, the problem of multi-machine cooperative operation path planning in maritime battlefield environment is analyzed, and path planning is divided into global path planning and local dynamic obstacle avoidance.

2. A global path planning scheme based on improved A* algorithm coupled with reinforcement learning is proposed. Considering the influence of the weight \( w(n) \) on the estimated cost \( h(n) \), the generated job path is optimized for corners, and the Bezier curve is used. It is path smoothed.

3. According to the randomly generated marine scene obstacle environment map, experiments are carried out on the global path planning algorithm respectively. The results show that by selecting the appropriate weight \( w(n) \), the path search efficiency is significantly improved; after the corner optimization of the global path, the number of turns is effectively reduced when the path length and running time are similar; using Bezier curve After the path is smoothed, the peaks at the corners are optimized to ensure smooth path planning, which initially meets the requirements of real-time and smoothness.

In future research, we should consider the situation that multiple search devices are searching at the same time, and the path planning method based on multi-objective collaborative search will further improve the success rate and efficiency of target search in the naval battlefield.

**REFERENCES**

[1] L. Fei, L. Zhaoquan, N. Chen, S. Weiye, and Z. Haodong, “Research on 3D path planning based on improved ant colony algorithm,” Sensors Microwzy, vol. 41, no. 1, pp. 45–49, 2022, doi: 10.13873/J.1000-9787(2022)01-0045-05.
