Research on path Planning algorithm of Unmanned ship in narrow Water area

Liang SUN¹, a, Yue-lin ZHAO², b*, Jian ZHANG³, c

¹college of Navigation, Dalian Maritime University, Dalian, Liaoning, China
²college of Navigation, Dalian Maritime University, Dalian, Liaoning, China
³college of Navigation, Dalian Maritime University, Dalian, Liaoning, China

a e-mail: sl11075835@163.com
b Corresponding author email: yuelin_zhao@163.com
c e-mail: 17705426104@163.com

Abstract: Aiming at the problems of strong random search characteristics and high volatility of planning path in bi-directional Rapidly-exploring Random Tree (bi-RRT) algorithm, a new random tree growth strategy is proposed, the connection between random trees is improved to improve the path-finding speed of the algorithm, and the improved bi-RRT algorithm is used to design a path planning algorithm for unmanned ships in narrow waters. At the end of this paper, the unmanned ship path planning experiment in the island and reef area is carried out, and the experimental results show that the unmanned ship path planning algorithm based on the improved bi-RRT algorithm can quickly provide a safe path for the unmanned ship in narrow waters, and has advantages over the traditional bi-RRT algorithm in terms of time performance, planning path length and volatility, which can provide relevant reference for the research of unmanned ship intelligent navigation.

1. Introduction
At present, unmanned ship is an important direction of research and development in the shipping industry, and ship autonomous obstacle avoidance technology is an important pillar to ensure the safety of unmanned ship navigation. Ship path planning algorithm is the premise and basis for the realization of unmanned ship autonomous obstacle avoidance technology [1]. Therefore, it is necessary to study the unmanned ship path planning algorithm.

The research of unmanned ship path planning algorithm can be divided into different directions according to the type of objects designed to avoid, when the avoidance target is static obstacles such as islands and reefs, isolated water objects, this kind of research direction is called global path planning or static path planning algorithm research [1]. For example, probabilistic random search algorithms such as RRT are widely used in the research of unmanned ship global path planning [2,3]. The RRT algorithm does not need to build a map model and the path-finding speed is fast, but its random search characteristic also makes the path planning of the RRT algorithm fluctuating and the quality is low. Therefore, in order to overcome some of the shortcomings of the RRT algorithm, the bi-RRT algorithm with bi-directional search characteristics is proposed and used in the related research of unmanned ship path planning [4,5].

At present, there are still some shortcomings in the research of using bi-RRT algorithm to solve the
unmanned ship path planning problem, for example, there is no constraint on the random growth of bi-RRT algorithm, and the problem of useless search caused by random tree connection is not considered. In order to solve the above problems, based on the existing research, an unmanned ship path planning algorithm suitable for narrow waters is designed based on the improved bi-RRT algorithm\textsuperscript{[4,5]}. The improvement of the unmanned ship path planning algorithm is as follows: a random tree growth strategy is proposed to restrict the random search characteristics of the traditional bi-RRT and improve the stability of the planning path; in order to further improve the search efficiency of the bi-RRT algorithm, the connection between random trees is improved. In addition, the Dijkstra algorithm is introduced to re-optimize the preliminary planning path to improve the quality of the planning.

2. path. Introduction and analysis of bi-RRT algorithm

2.1. Introduction of bi-RRT algorithm

The bi-directional Rapidly-exploring Random Tree (bi-RRT) algorithm\textsuperscript{[6]} is improved by J.J.Kuffner and S.M.LaValle on the basis of the Rapidly-exploring Random Tree (RRT) algorithm. The improved idea is to generate the search random tree from the planning end point and the starting point respectively. When the two random trees approach a certain threshold, they will form an obstacle avoidance path. Therefore, compared with the RRT algorithm, the bi-RRT algorithm not only inherits the advantages of the RRT algorithm, such as no need to build a map model and fast search speed, but also overcomes the disadvantage of blind search of the RRT algorithm to a certain extent, and further improves the search speed of the algorithm\textsuperscript{[7]}. The schematic diagram of the bi-RRT algorithm is shown in figure 1.

2.2. analysis of bi-RRT algorithm

Bi-RRT algorithm is that two random trees grow in two directions, and any random tree has two growth directions, one is growing toward random points, and the other is growing toward the new growth point of another random tree. This bi-directional random tree growth strategy can not only make the two random trees close to each other, but also avoid the problem of pathfinding failure caused by the algorithm falling into a local trap. However, this growth strategy also has some problems, such as low search efficiency and high volatility of path planning.

Combined with figure (1), the traditional bi-RRT algorithm is to determine whether there is the possibility of being directly connected to another random tree in the area with a fixed step size after each generation of the new growth point \( p_{\text{new}} \) as the center of the circle. If it can be connected directly, the pathfinding algorithm ends successfully. Otherwise, another random tree continues to grow to seek the opportunity to connect the two random trees. However, when the newly growing node of a random tree \( p_{\text{new}} \) can be connected with another random tree at a distance other than a fixed step, or when there is a possibility of direct connection between the existing nodes of two random trees, the traditional random tree connection methods do not connect but choose to continue to grow.
3. Improvement of bi-RRT algorithm

3.1. Brief introduction of bi-RRT algorithm improvement ideas

Above, the defects of the traditional bi-RRT algorithm are analyzed, and this section will improve the shortcomings of the bi-RRT algorithm. The improvement idea of the bi-RRT algorithm is to determine the growth direction of the random tree by designing a new random tree growth strategy, to determine whether the path-finding is successful by using the random tree fast connection method after getting the new growth point, and to use the Dijkstra algorithm to optimize the path after the success of the path-finding. The specific improvement strategy will be listed in the subsequent summary, at the same time, in order to better clarify the design ideas of the improved algorithm, the algorithm flow of the improved algorithm is listed in figure (2).

![Improved bi-RRT algorithm flow chart](image)

Fig 2. Improved bi-RRT algorithm flow chart

3.2. Random tree growth strategy

First of all, the new random tree growth strategy should ensure that the algorithm can jump out of the local trap. Therefore, the concepts of small probability threshold \( q_{\text{prob}} \) and probability factor \( q \) are introduced. When \( q \) is less than \( q_{\text{prob}} \), a random point is generated in the current environment, and the random point is taken as the growth direction of the random tree. When \( q \) is greater than or equal to \( q_{\text{prob}} \), the dominant node \( p_{\text{best}} \) is selected from all the nodes in another random tree as the growth direction of the random tree to reduce the disordered search of the random tree. The above growth strategy is shown in formula (1). In addition, the number of random tree attempts \( k \) to grow is limited by setting the maximum number of attempts. When the number of attempts is greater than \( k \), it means that the growth of the random tree fails, and another random tree grows. The above strategies can limit the random search characteristics of the algorithm and improve the efficiency of the algorithm.

\[
\begin{align*}
p_{\text{rand}} &= \text{rand}(m, n) \quad \text{if } q < q_{\text{prob}} \\
p_{\text{rand}} &= p_{\text{best}} \quad \text{if } q \geq q_{\text{prob}}
\end{align*}
\]

(1)
In formula (1), $rand$ is a random function, $m$, $n$ is the dimension of the current adaptation space, and the dominant node $p_{\text{best}}$ filtered by another search tree. When determining the $p_{\text{best}}$, the distance factors between nodes, steering angle factors and collision factors are comprehensively considered. When it is assumed to be greater than, the filtering method described in the process of generating a new node with a random tree is as follows.

### 3.2.1. distance factor

In the selection of dominant nodes, distance heuristic information is added to make the two random trees close to the target point in the process of growth, so as to reduce the waste of step size caused by unnecessary search. The setting method of distance heuristic information is shown in formula (2).

$$
D = \frac{1}{d(p_{\text{goal}}, p_i)}
$$

In formula (2), $d$ represents the Euclidean distance function, $p_{\text{goal}}$ represents the root node of the random tree $T_2$, $p_i$ is the node in the random tree $T_1$, $D$ represents the distance cost value of the node to be selected.

### 3.2.2. steering angle factor

The purpose of limiting the turning angle of the growth point is to avoid a large angle turning point in the growth process of the search tree, which leads to the irrationality of the planning path. Moreover, considering the corner factor is also beneficial to improve the smoothness of the path and improve the speed of the algorithm. The idea of this paper is to select the node $p_{\text{nearest}}'$ of the current random tree $T_2$ which is closest to the root node of another random tree $T_1$, and take the two-point connection line of the nearest node $p_{\text{nearest}}'$ and the root node $p_{\text{init}}$ of $T_1$ as the baseline. The smaller the node $p_{\text{init}}$ deviates from the baseline, the smaller the corresponding steering angle. The corresponding principle is shown in figure 2. The calculation method of the steering angle limiting factor of the node to be selected is shown in formula (3).

$$
Y = \frac{|Ax + By + C|}{\sqrt{A^2 + B^2}}
$$

$$
\begin{align*}
A &= y_2 - y_1 \\
B &= x_2 - x_1 \\
C &= x_1 y_2 - x_2 y_1
\end{align*}
$$

In formula (3), $(x_1, y_1)$ is the coordinate of the selected node $p_{\text{init}}$ in the current adaptation space in the search tree $T_1$, and $Y$ represents the turning angle factor value of the selected node. In formula (4), $(x_1, y_1)$ denotes the coordinates of the root node $p_{\text{init}}$ of the random tree $T_1$ in the current space, and $(x_2, y_2)$ denotes the coordinates of the search tree $T_2$ to the nearest node $p_{\text{nearest}}'$ of $p_{\text{init}}$ in the current adaptation space.

### 3.2.3. Collision factor

As shown in figure (3), according to the traditional bi-RRT algorithm, the new growing point is $p_{\text{new}}'$, but there are obstacles in front of the $p_{\text{new}}'$, so the growth of random tree $T_i$ is likely to be useless. Because the traditional bi-RRT algorithm does not take into account the obstacles in the growth direction of the random tree, the search efficiency of the random tree in the narrow water area and a large number of irregular obstacles is very low. Therefore, in the selection of random points, the collision in the direction of the selected point should be detected in advance, and when there are large area obstacles in this direction, the search in this direction should be avoided. As shown in figure (3), the $T_i$ search tree should grow in the direction of $p_{\text{best}}$. The specific algorithm for taking collision factors into account in the screening of $p_{\text{best}}$ is shown in formula (5).
\[ C = \frac{\Delta t \times I}{d(p', p_{\text{nearst}}')} \]  

(5)

In formula (5), \( \Delta t \) is the discrete step, \( I \) is the number of times obstacles are detected, and \( C \) represents the collision cost of the node to be selected.

After considering the distance between the selected nodes, the turning angle factor and the collision factor, the node \( p_{\text{best}} \) with the lowest comprehensive factor \( S \) in \( T_i \) is selected as the direction of \( T_2 \) growing nodes. The specific method of calculating the comprehensive factor cost \( S \) is shown in formula (6).

\[ S = \min \sum_{i=1}^{d}(D_i + Y_i + C_{p_i}) \]  

(6)

### 3.3. Fast connection method of Random Tree

Combined with figure (4), and taking the new node \( p_{\text{new}} \) growing from random tree \( T_i \) as an example, the connection mechanism of this new random tree is illustrated. After the random tree \( T_i \) obtains the new growth point \( p_{\text{new}} \), we should judge whether there is the possibility of direct connection between the node in another random tree \( T_2 \) and the new growth point \( p_{\text{new}} \) of \( T_i \). In addition, considering the randomness of the growth direction of the growing point, we should also find the nearest target node \( p_{\text{nearst}}' \) from the random tree \( T_2 \) to the \( T_i \) root node \( p_{\text{init}} \) of the random tree, and judge whether the \( p_{\text{nearst}}' \) is directly connected to the random tree \( T_i \). If you can connect directly from both directions, choose the nearest direction to connect. As shown in figure (4), \( p_{\text{nearst}}' \) is finally selected to connect with \( p_{\text{best}} \). If neither of the two directions can be directly connected, the two random trees continue to grow alternately until they are connected to each other.
4. Simulation experiment and result analysis
In order to verify the good performance of the unmanned ship path planning algorithm in narrow waters based on the improved bi-RRT algorithm, the path planning simulation experiments were carried out in a certain area of Zhoushan with many islands and reefs, and the path planning process and experimental results were analyzed.

4.1. Analysis of path planning process
The map size of this path planning experiment is 6.5 * 12 square kilometers, and the map overview is shown in figure (5). In figure (5), the green triangle represents the starting point of the planning, the starting point coordinates is (600,35), and the red pentagram represents the end point of the path planning, and the end coordinates is (900,620).

This path planning simulation experiment will use the traditional bi-RRT algorithm and the unmanned ship path planning algorithm based on the improved bi-RRT algorithm (hereinafter referred to as the improved bi-RRT algorithm) for path planning experiments. In order to reflect the objectivity of the experiment, when using the two algorithms for the preliminary path planning experiment, the Dijkstra algorithm is used for path re-optimization. The probability threshold $p_{rob}$ of the improved bi-RRT algorithm is 0.38, the maximum number of attempts $k$ is 10000, and the fixed step size of the two algorithms is 10 (step size unit is 10 meters).

First of all, the path planning experiment is carried out by using the traditional bi-RRT algorithm. The experimental results are as follows: the path search time is 121.02375 seconds, and the path search process is shown in figure (5); the initially planned path length is 13573.06 meters (shown by the green solid line in figure 6), and the path length optimized by Dijkstra algorithm is 11728.5 meters (red solid line in figure 6).

![Fig 5. Path search process of traditional bi-RRT algorithm](image1)

![Fig. 6 traditional bi-RRT algorithm for path planning](image2)
Then, the path planning experiment is carried out by using the improved Bi-RRT algorithm, and the experimental results are as follows: the path search time is 73.34106 seconds, and the path search process is shown in figure (7); the initial planned path length is 11635.18 meters (as shown by the light blue line in figure 8), and the optimized path length by Dijkstra algorithm is 10507.04 meters (shown in figure 8).

![Figure 7. Improving the path search process of bi-RRT algorithm](image)

![Figure 8. Improved bi-RRT algorithm for path planning](image)

As shown in figure(6), because the random tree growth strategy of the traditional bi-RRT algorithm does not take into account the random constraint of the growing point, the algorithm will appear many useless nodes in the search process. Moreover, it can be found from figure (6) that even if the random trees are connected to each other, the traditional bi-RRT algorithm will continue to search, so the path search time of the traditional bi-RRT algorithm is longer. Looking back at figure(8), we can find that there are fewer useless nodes in the random tree generated by the improved bi-RRT algorithm, and the fast connection mechanism of the random tree designed by the improved bi-RRT algorithm effectively improves the efficiency of path search. In addition, the starting point of this path planning is placed in the trap area, and the success of the improved bi-RRT algorithm also shows that the algorithm has the ability to escape the trap area.

By comparing figure (6) and figure (8), it can be found that even though the two algorithms use the Dijkstra algorithm to re-optimize the path, the improved bi-RRT algorithm re-optimizes the path obtained by the preliminary planning through the Dijkstra algorithm, so that the final path is better than the traditional bi-RRT algorithm in terms of path smoothness and path length. Therefore, from the above experimental results, we can see that the improved Bi-RRT algorithm has less planning time, high planning path quality and good performance.
4.2. Stability Analysis of improved bi-RRT algorithm

Considering that both the traditional bi-RRT algorithm and the unmanned ship path planning algorithm based on the improved bi-RRT algorithm (hereinafter referred to as the improved bi-RRT algorithm) have certain random search characteristics, and in order to better prove that the improved bi-RRT has a better stable performance, the two algorithms are used for 50 independent experiments, and the experimental results are processed and listed in Table (1).

Table 1. Experimental results of algorithm

| Time index | Average value (Unit) | Variance (Unit) | coefficient of variation (Unit) |
|------------|---------------------|-----------------|-------------------------------|
| Traditional bi-RRT | 124.1974 (s) | 55.7154 (s) | 0.4486 (s) |
| Improved bi-RRT  | 75.4593 (s)  | 29.6824 (s)  | 0.39385 (s) |

| Distance index | Average value (Unit) | Variance (Unit) | coefficient of variation (Unit) |
|----------------|---------------------|-----------------|-------------------------------|
| Traditional bi-RRT | 11216.0144 (m) | 614.2002 (m) | 0.054761 (m) |
| Improved bi-RRT  | 10777.2596 (m) | 157.3162 (m) | 0.014597 (m) |

Combined with Table (1), from the average point of view, the time index of the improved bi-RRT algorithm is about 40% lower than that of the traditional Bi-RRT algorithm. In addition, from the standard deviation of the results of the two algorithms, we can see that the planning time and planning distance given by the improved bi-RRT algorithm are relatively stable. The value coefficient of variation of the improved bi-RRT algorithm is much smaller than that of the traditional Bi-RRT algorithm, which shows that the improved bi-RRT algorithm has good stability.

5. Conclusion

In this paper, the traditional bi-RRT algorithm is improved by redefining the random tree growth strategy, improving the random tree connection mode and introducing the Dijkstra algorithm, and an unmanned ship path planning algorithm suitable for narrow waters is proposed based on the improved bi-RRT algorithm. The comparative experimental results of the two algorithms at the end of this paper show that the unmanned ship path planning algorithm has a great improvement in time performance and path quality compared with the traditional bi-RRT algorithm, and the unmanned ship path planning algorithm also has great advantages in terms of algorithm stability. Therefore, the algorithm proposed in this paper can provide relevant reference for the autonomous navigation of unmanned ship in narrow waters.

References

[1] Zhang D, Zhao YX, Cui YF, et al. A Visualization Analysis and Development Trend of Intelligent[J]. Journal of Transport Information and Safety. 2021,39(01):7-16.
[2] Li YF, Shao GM, Wu ZQ, et al. Research on Navigation Planning of Unmanned Surface Vessel in Complex Water Area Environment[J]. China Shipbuilding, 2020561 (S1): 70-79.
[3] Zhuang JY, Zhang L, Sun HB, et al. Improved rapidly-exploring random tree algorithm application in unmanned surface vehicle local path planning [J]. Journal of Harbin Institute of Technology, 2015 Magi 47 (1): 112-117.
[4] Xiang JL, Wang HD, OUYANG ZL, etc. Algorithm of Local Path Planning for Unmanned Surface Vehicles Based on Improved Bi-RRT[J]. China Shipbuilding, 2020,561 (1): 157-166.
[5] OUYANG ZL, Wang HD, Wang JY, et al. Automatic collision avoidance algorithm for unmanned surface vessel based on improved Bi-RRT algorithm [J]. Chinese Journal of Ship Research, 2019714 (6): 8-14.
[6] Kuffner J J, LaValle S M, Authors A. RRT-connect: An efficient approach to single-query path
planning[J]. 2000.

[7] Huang YF, Hu LK, Xue WC. Improved RRT-Connect mobile robot path planning algorithm [J]. Computer Engineering, 2021Vol 1-9.