

Two-Layer Path Planner for AUVs Based on the Improved AAF-RRT Algorithm

Le Hong^{1,2,3}, Changhui Song^{2,3}, Ping Yang^{2,3} and Weicheng Cui^{2,3}

Received: 06 July 2021 / Accepted: 02 December 2021

© Harbin Engineering University and Springer-Verlag GmbH Germany, part of Springer Nature 2022

Abstract

As autonomous underwater vehicles (AUVs) merely adopt the inductive obstacle avoidance mechanism to avoid collisions with underwater obstacles, path planners for underwater robots should consider the poor search efficiency and inadequate collision-avoidance ability. To overcome these problems, a specific two-player path planner based on an improved algorithm is designed. First, by combining the artificial attractive field (AAF) of artificial potential field (APF) approach with the random rapidly exploring tree (RRT) algorithm, an improved AAF-RRT algorithm with a changing attractive force proportional to the Euler distance between the point to be extended and the goal point is proposed. Second, a two-layer path planner is designed with path smoothing, which combines global planning and local planning. Finally, as verified by the simulations, the improved AAF-RRT algorithm has the strongest searching ability and the ability to cross the narrow passage among the studied three algorithms, which are the basic RRT algorithm, the common AAF-RRT algorithm, and the improved AAF-RRT algorithm. Moreover, the two-layer path planner can plan a global and optimal path for AUVs if a sudden obstacle is added to the simulation environment.

Keywords Autonomous underwater vehicles (AUVs); Path planner; Random rapidly exploring tree (RRT); Artificial attractive field (AAF); Path smoothing

1 Introduction

Path planning is one of the core technologies in autonomous underwater vehicles (AUVs), and it is also a key link in realizing autonomous navigation and operation

(Sun et al. 2020). Path planning in underwater robots refers to finding a collision-free path from the initial state to the target state according to certain evaluation criteria in the water environment containing obstacles. In addition, to a certain extent, path planning technology marks the intelligence level of underwater robots (Gasparetto et al. 2015; Sariff and Buniyamin 2006). However, compared with other types of mobile robots, the working environment of underwater robots is a large-scale three-dimensional (3D) space. The complexity of the underwater environment increases the difficulty of path planning. At the same time, the robot's position, energy consumption, self-attitude, and motion constraints should be considered (Gu et al. 2005). Therefore, it is very important to choose a reasonable and suitable path planning method for underwater robots.

Path planning methods for underwater robots can be divided into two categories: traditional approaches and intelligent approaches. Traditional approaches include path planning methods based on the road map construction method (Lozano-Pérez 1990), element decomposition method (Li 2012; Tanakitkorn et al. 2014), A* algorithm, D* algorithm, and artificial potential field (APF) method. The A* algorithm, D* algorithm, and APF method are the most widely used path planning approaches at present.

Article Highlights

- An improved AAF-RRT path planning algorithm for AUVs is proposed.
- The algorithm makes the attractive factor proportional to the distance between the point to be extended and the goal point.
- The algorithm is verified to have a better performance than the common AAF-RRT algorithm and basic RRT algorithm in terms of the search ability and ability to pass through narrow passages.

✉ Weicheng Cui
cuiweicheng@westlake.edu.cn

¹ Zhejiang University-Westlake University Joint Training, Zhejiang University, Hangzhou 310024, China

² Key Laboratory of Coastal Environment and Resources of Zhejiang Province (KLaCER), School of Engineering, Westlake University, Hangzhou 310024, China

³ Institute of Advanced Technology, Westlake Institute for Advanced Study, Hangzhou 310024, China

The A* algorithm is suitable for static path planning (Hart et al. 1972). D* algorithm is a dynamic A* algorithm, which is suitable for solving the dynamic path planning problem by detecting the change in the previous node or nearby nodes of the shortest path (Carsten et al. 2006). The APF method is a virtual method proposed by Khatib etc. (Khatib 2003; Volpe and Khosla 1990). The intelligent path planning approaches include swarm intelligence-based methods and the method based on machine learning (Sun et al. 2020). Path planning methods for underwater robots based on swarm intelligence mainly include genetic algorithm (Xu and Li 2010; Xu and Yao 2008), ant colony algorithm (Liu et al. 2016; Wang et al. 2008), and firefly algorithm (Dong 2013; Liu et al. 2015). Algorithms based on swarm intelligence are relatively mature in global path planning, but they do not have a high real-time performance. Concerning the application of machine learning to the path planning for underwater robots, Zhu et al. focused on the situation of sudden obstacles, using environmental changes to cause changes in neuron excitation and activity output values, and thereby outputting collision-free path points (Zhu et al. 2015). To sum it up, traditional path planning methods are simple to describe and easy to implement. However, they generally lack flexibility and the ability to be applied in 3D complex underwater environments. Intelligent methods make up for many shortcomings of traditional methods. They can solve global and local path planning problems. However, based on the basic environment model, they can obtain high real-time planning.

Considering the above-listed drawbacks of the two common approaches, in this paper, the RRT algorithm is adopted as the basic algorithm to study for the path planning of underwater robots. The RRT algorithm is chosen for three main reasons. First, the RRT algorithm is based on a tree structure and random sampling. Therefore, rather than an arbitrary geometric curve between two points, the route from the point to be extended and the new point could be generated by the kinematics and dynamics simulations of robots. Thus, it is suitable for solving dynamic path planning problems. Second, based on random sampling, the RRT algorithm does not need to search all points in the space and model the obstacle space accurately. Lastly, RRT algorithms have advantages in multidimensional spaces. The higher the dimensionality is, the more the node states are. For traditional path planning methods, these algorithms can easily cause dimensionality disasters. However, RRT algorithms are based on random sampling. In a high-dimensional space, they just need more dimensions for sampling, rather than sampling every state in the space.

The search randomness of RRT algorithms makes them have a strong exploration ability. However, it will lead to blindness while planning a certain path. At present, to solve this problem, scholars have proposed some strategies. In the early stage, a single RRT-tree was mainly used for searching. To improve the search speed, scholars pro-

posed biased RRT (Lavalle and Kuffner 2000) and bidirectional RRT (Jr and Lavalle 2000) search algorithms. Therefore, improved RRT algorithms generally include two categories, namely, heuristic algorithms and bidirectional search approaches, which improve the convergence speed to a certain extent. To consider kinematic constraints of the intelligent vehicles, Liu, et al. proposed a target-biasing strategy, which reduces the nodes of trees and improves the search efficiency and running time of an algorithm (Liu et al. 2017). Wang proposed a smooth splicing algorithm (Wang 2014). By optimizing the connection line between random points, the path is generated under the constraint of the motion system. However, the above-mentioned improved RRT algorithms can only plan a relatively optimal path at the cost of search efficiency, which would increase the complexity and searching time of path planning. Therefore, how to plan an optimized path without influencing the running efficiency of RRT algorithms is the problem. Recently, combined algorithms have performed well in solving this problem. By combining membrane computing, genetic algorithm, and the APF method, hybrid approaches successfully reduce time complexity (Orozco-Rosas et al. 2019b) and speed up the computation (Orozco-Rosas et al. 2019a). Inspired by the advantages of hybrid approaches, APF, a goal-oriented algorithm, may be a good choice for improving the performance of RRT algorithms.

Inspired by the two problems, the contributions of this study lie in the following aspects. First, the RRT algorithm is combined with the concepts of AAF in the APF method to guide random trees to grow in the direction of the target point. Based on the common AAF-RRT algorithm with a constant attractive force, an improved AAF-RRT algorithm with a proportional attractive force is proposed to enhance the goal orientation of the algorithm and reduce the computation resource. Second, other researchers mainly focus on studying obstacle avoidance on the global path, which has been planned for AUVs (Cai et al. 2020). To simultaneously plan an optimal and global path, in this study, a two-layer path planner with path smoothing is designed, which combines the global planning based on the proposed AAF-RRT algorithm and the approaches of the local planning. Third, based on the simulation results, the proposed improved AAF-RRT algorithm is verified to perform better than the common AAF-RRT algorithm in the three aspects, namely, search ability, collision-avoidance ability, and the ability to pass through narrow passages. Accordingly, based on the proposed AAF-RRT, the two-layer path planner can successfully plan the global and optimal path for AUVs when adding sudden obstacles to a simulated map.

2 Preliminaries for the RRT and APF

In this section, the concepts, characteristics, and plan-

ning process of the RRT algorithm and APF approach will be illustrated in detail.

2.1 RRT algorithm

Uninterruptedly searching the state space of robots via random sampling is the basic principle of RRT algorithms. It can be generally divided into two steps. First, a random tree is rapidly established. Then, an effective path from the initial point to the goal point is queried based on the random tree. The corresponding pseudocode of RRT is shown in Algorithm 1.

Algorithm 1 RRT algorithm

Input: the initial point x_{init} ; the number of samples K ; the step size t between the two points
Output: the RRT-Tree T

```

1 Initialization;23456789
2 while  $k \leq K$  do
3    $x_{rand} = \text{Random\_State}()$ ;
4    $x_{near} = \text{Nearest\_Neighbor}(x_{rand}, x_{near})$ ;
5    $u = \text{New\_State}(x_{rand}, x_{near})$ ;
6    $x_{new} = \text{New\_State}(x_{rand}, x_{near})$ ;
7    $T.add\_vertex(x_{new})$ ;
8    $T.add\_vertex(x_{near}, x_{new}, u)$ ;
9 return  $T$ ;
```

The detailed steps of algorithm 1 can be illustrated in the following steps:

Step 1 Initialize the whole statespace. Define the initial point x_{init} , the number of samples K , and the step size t between the two points.

Step 2 Use the function $\text{Random_State}()$ to randomly generate a point x_{rand} .

Step 3 Use the function $\text{Nearest_Neighbor}()$ to find the point x_{near} closest to the point x_{rand} .

Step 4 Get the extension vector using the function $\text{Selet_Input}()$ and get the new point x_{new} with the step size t and extension vector u .

Step 5 Determine whether there is an obstacle from x_{near} to x_{new} . If it does not exist, then add the point x_{new} to the tree vertex and connect x_{new} with x_{near} .

Figure 1 vividly shows the growth of a random tree in a simulated environment. The blue triangle and red triangle represent the start point and goal point, respectively. Green circles denote points in the tree, and the blank circle denotes the sampling point. Located in the center of the figure, the two white irregular frustums of a pyramid represent the two obstacles in the environment. The expanding process of RRT can be illustrated as follows. After sampling point 5 in the map, the RRT algorithm would figure out the nearest point 3 in the tree. Then, the new point 4 would be added in the tree connected with point 3 at a prescribed step size.

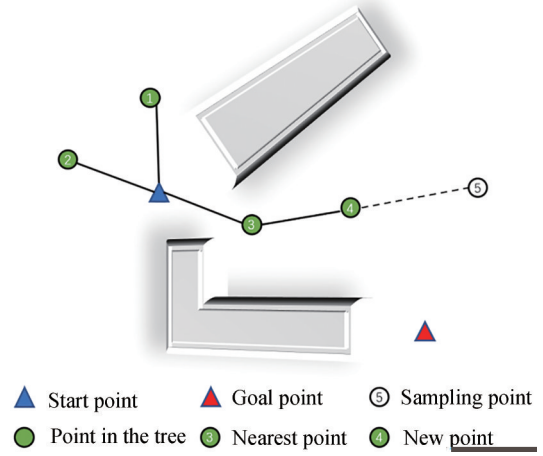


Figure 1 Growth process of random trees in RRT

2.2 APF approach

The APF method is a classical path planning algorithm for robots. The APF regards the target position and obstacles as objects that have attractive and repulsive forces on robots, respectively. Therefore, the APF contains two kinds of force fields: AAF formed by the target position and artificial repulsive field (ARF) formed by various obstacles. The robot would move along the synthetic force of the attractive force and repulsive forces. This synthetic force is given as follows:

$$F(x) = F_{att}(x) + F_{rep}(x) \quad (1)$$

where x denotes the coordinate position of the intelligent vehicle in the environment, $F_{att}(x)$ is the attractive force that guides the vehicle to move to the goal position, and $F_{rep}(x)$ is the repulsive force that makes the vehicle avoid obstacles. $F(x)$ is the synthetic force of $F_{att}(x)$ and $F_{rep}(x)$.

Figure 2 shows the force analysis of an intelligent vehicle in the APF algorithm. The blue oval denotes an intelligent vehicle in a simulated environment. The red triangle and white frustum of a pyramid still represent the goal point and obstacle in the environment, respectively. According to the previous illustration of the APF, F_1 and F_2 are the repulsive forces that two obstacles give to the intelligent vehicle. F_0 is the attractive force given from the goal position to the position of the intelligent vehicle. After the synthesis of the three forces, F_3 is the final force that the vehicle receives. The vehicle would move under the guidance of F_3 .

3 Improved AAF-RRT algorithm

As the randomness of RRT causes its search blindness, to improve the search efficiency, we add the concepts of

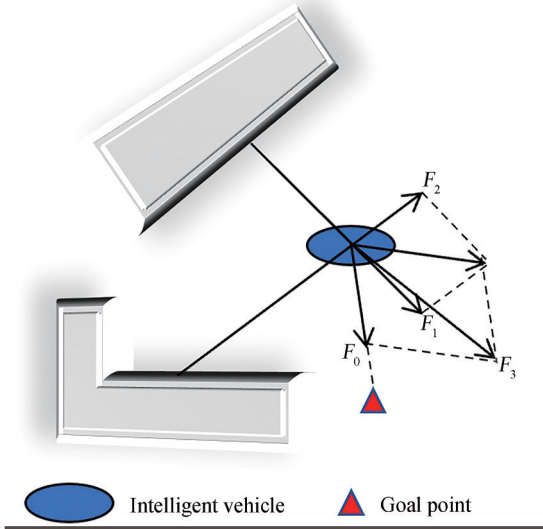


Figure 2 Force analysis of an intelligent vehicle in the APF

the AAF in APF to the extension of random trees in the RRT algorithms. Then, the goal point in the search environment would have an attractive field for robots. Considering that there exist the judging criteria for the collisions among points and obstacles in RRT, the concept of ARF in the APF approach is not necessary to be added to RRT algorithms. After this combination, RRT becomes goal-oriented, and this improved RRT algorithm is named AAF-RRT algorithm in this paper, which is a kind of RRT algorithm integrated with AAF.

In the concept of AAF, the potential function carries the information of targets, which should be delayed to RRT. Then, the random tree can move in the direction of the potential field from the targets when exploring the space. In addition, the excellent exploration ability of RRT can make up for the shortcoming that the potential field method easily falls into a local minimum. After all, we combine RRT and AAF by adding the idea of target attraction into the extension of the RRT random tree. The detailed illustration of the extension in AAF-RRT is as follows: First, after randomly generating the n^{th} sample point $p_{\text{rand}}(n)$, the corresponding n^{th} nearest point, $p_{\text{near}}(n)$, is obtained. Then, for the n^{th} nearest point $p_{\text{near}}(n)$, $\mathbf{R}(n)$ represents the extension vector that the random point $p_{\text{rand}}(n)$ gives to $p_{\text{near}}(n)$. The attractive vector $\mathbf{A}(n)$ is given from the goal point to $p_{\text{near}}(n)$. $\mathbf{R}(n)$ and $\mathbf{A}(n)$ simultaneously affect the extension of the random tree. Therefore, the random tree can grow toward the target point with a certain and rational trend.

According to the basic RRT algorithm, the extension vector of the nearest point $p_{\text{near}}(n)$ can be expressed in the following mathematical formula:

$$\mathbf{R}(n) = \varepsilon \cdot \frac{p_{\text{rand}}(n) - p_{\text{near}}(n)}{\|p_{\text{rand}}(n) - p_{\text{near}}(n)\|} \quad (2)$$

where $\mathbf{R}(n)$ denotes the extension vector that the n^{th} random point $p_{\text{rand}}(n)$ gives to the n^{th} nearest point $p_{\text{near}}(n)$. ε represents the step size in the RRT algorithm. Moreover, the fraction $\frac{p_{\text{rand}}(n) - p_{\text{near}}(n)}{\|p_{\text{rand}}(n) - p_{\text{near}}(n)\|}$ denotes the unit vector of the line connecting $p_{\text{rand}}(n)$ and $p_{\text{near}}(n)$, which later would become the unit vector of $\mathbf{R}(n)$.

Referring to Eq. (2), the artificial attractive vector $\mathbf{A}(n)$ of the AAF-RRT algorithm constructed in this work can be elaborated by Eq. (3).

$$\mathbf{A}(n) = \rho(n) \cdot \varepsilon \cdot \frac{p_{\text{goal}} - p_{\text{near}}(n)}{\|p_{\text{goal}} - p_{\text{near}}(n)\|} \quad (3)$$

where $\rho(n)$ is the attractive factor, which can change the target bias of the random tree. $\mathbf{A}(n)$ is the attractive vector given to the n^{th} nearest point $p_{\text{near}}(n)$ from the goal point p_{goal} . ε still represents the step size of the algorithm. $\frac{p_{\text{goal}} - p_{\text{near}}(n)}{\|p_{\text{goal}} - p_{\text{near}}(n)\|}$ denotes the unit vector of the line connecting p_{goal} and $p_{\text{near}}(n)$. Therefore, the final guided extension vector $\mathbf{F}(n)$ is as follows:

$$\mathbf{F}(n) = \mathbf{A}(n) + \mathbf{R}(n) \quad (4)$$

Then, the calculation equation for the new extended point $p_{\text{new}}(n)$ can be obtained by Eq. (5).

$$p_{\text{new}}(n) = p_{\text{near}}(n) + \mathbf{F}(n) = p_{\text{near}}(n) + \varepsilon \cdot \left[\frac{p_{\text{rand}}(n) - p_{\text{near}}(n)}{\|p_{\text{rand}}(n) - p_{\text{near}}(n)\|} + \rho(n) \cdot \frac{p_{\text{goal}} - p_{\text{near}}(n)}{\|p_{\text{goal}} - p_{\text{near}}(n)\|} \right] \quad (5)$$

Eq. (5) shows that the calculation for $p_{\text{new}}(n)$ includes two parts, i.e., the extension vector from the sample point $p_{\text{rand}}(n)$ and the attractive vector from the goal point p_{goal} . Then, based on Algorithm 1 of the basic RRT algorithms, the pseudocode of the AAF-RRT algorithms is shown in Algorithm 2.

Compared with Algorithm 1, \mathbf{v} and the function

Algorithm 2 AAF-RRT algorithm

Input: the initial point x_{init} ; the initial point x_{init} ; the number of samples K ; the step size t between the two points

Output: the AAF-RRT-Tree \mathbf{T}

```

1 Initialization; 23456789
2 while  $k \leq K$  do
3    $x_{\text{rand}} = \text{Random\_State}()$ ;
4    $x_{\text{near}} = \text{Nearest\_Neighbor}(x_{\text{rand}}, x_{\text{near}})$ ;
5    $\mathbf{u} = \text{Select\_Input}(x_{\text{rand}}, x_{\text{near}})$ ;
6    $\mathbf{v} = \text{Attractive\_Input}(x_{\text{goal}}, x_{\text{near}})$ ;
7    $x_{\text{new}} = \text{New\_State}(x_{\text{rand}}, x_{\text{near}})$ ;
8    $\mathbf{T.add\_vertex}(x_{\text{new}})$ ;
9    $\mathbf{T.add\_vertex}(x_{\text{near}}, x_{\text{new}}, \mathbf{u}, \mathbf{v})$ ;
10 return  $\mathbf{T}$ ;
```

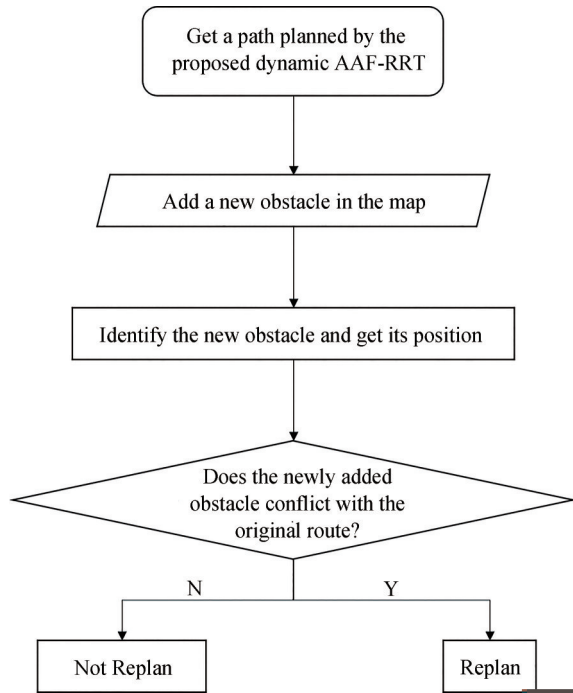


Figure 4 Diagram of the two-layer path planner

rithm. The other steps in the diagram all belong to the local planning. The detailed illustration of the local planning is as follows: After a sudden obstacle is added to the environmental model, the planner would identify its position and determine whether it conflicts with the previous path. Then, according to the determination, the planner would replan the path or not.

4.2 Path smoothing

Owing to the randomness of the RRT algorithm, the algorithm has a strong exploration ability. However, the randomness will lead to the blindness of path planning. Accordingly, the planned path would have more redundant points, which increases the corners and the length of the path. Therefore, many optimization methods have been used to determine sub-optimal and optimal paths. The most common and simplest optimization method is pruning, which aims to cut out unnecessary nodes. As shown in Figure 5, if there is no collision between point 1 and point 3, then the line from point 1 to point 2 and the line from point 2 to point 3 can be directly replaced with the line from point 1 to point 3.

After path smoothing is conducted by pruning, redundant points could be correspondingly cut, and the planned path would be shorter according to the triangle inequality.

5 Simulation results

Simulations are conducted in two parts: The first part is

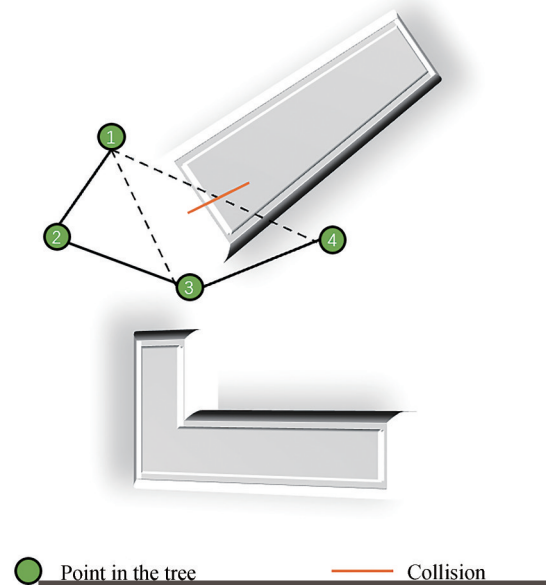


Figure 5 Diagram of the two-layer path planner

the simulations for comparing the improved AAF-RRT algorithm with the common AAF-RRT algorithm and basic RRT algorithm. The simulation includes the selection of the environmental mazes, outputting the figures on the views of the random trees, processing time and path length obtained by the algorithms, and conducting the rank-sum tests on the data samples. The second simulations are executed for the two-player path planner. A new environmental maze is chosen for the simulation. Then, the figures on the planned path after the path smoothing and path replanning after the emergence of the sudden obstacles are outputted.

5.1 Simulations for the algorithms

Several mazes are used for studying RRT algorithms. From a comprehensive view, three different mazes with different layouts of obstacles are chosen as the simulation environments, as shown in Figure 6. Based on the simulation results under the three simulation environments, the searching ability of the three algorithms, i.e., basic RRT algorithm, common AAF-RRT algorithm, and improved AAF-RRT algorithm, can be deeply compared and analyzed.

In the figure, the black pattern represents the obstacle that intelligent agents cannot pass, whereas the white areas are where the agents can pass. The coordinate system of the maze is established as follows: The upper long side of the maze is the x -axis. The right side is the positive direction of the x -axis. The left side of the maze is the y -axis, the positive direction of which is the downward direction. The green hollow circles in the mazes are the initial points, and the red hollow circles are the goal points. The corresponding coordinate positions of the two types of points are unified to be (10, 10) and (290, 290), respective-

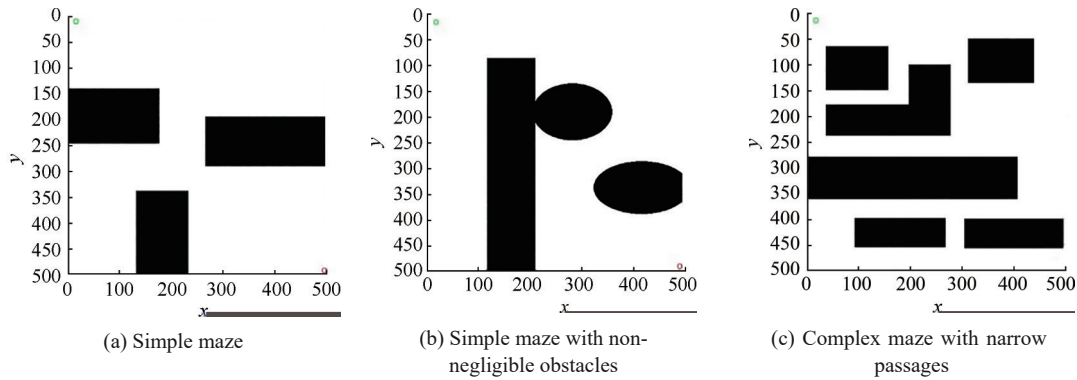


Figure 6 Three different mazes with three different layouts

ly. Figure 6(a) shows a simple maze with three square obstacles. There is no obvious obstacle blocking the agents from the start point to the goal point. This simple maze is used for the preliminary exploration of the search speed, search performance, and collision-avoidance ability of the three algorithms. For further exploration of the collision-avoidance ability, another simple maze, shown in Figure 6(b), contains the non-negligible obstacles on the line from the start point to the goal point. Considering the poor ability of the basic RRT algorithm to pass through the narrow passage in the environment, a complex maze with the narrow passage is selected. Whether AAF-RRT algorithms can improve this kind of ability would be answered and confirmed.

Before the simulation of the three algorithms under the three different mazes, we unified several standards for the subsequent simulations. First, the coordinate positions of the start point and goal point are set as (10, 10) and (290, 290), respectively. The maximum iterations of the simulations are 10 000. Two indexes are chosen to reflect the searching ability and collision-avoidance ability of the three planning algorithms, which are the processing time (s) reflecting the search speed and the path length reflecting the search performance. The overview of the random tree and final planned path would be outputted under the three algorithms, which could directly reflect the search randomness of the sampling-based planning algorithms. Lastly, according to Eqs. (5) and (6), we assume that k_1 and k_2 in the subsequent simulations are equal to 0.02 and 0.0001, respectively.

5.1.1 Simple maze

For the preliminary exploration, simulations are conducted in the simple maze. Again, the black areas represent the obstacles in the maze, and white areas represent the passable areas in the maze. The simulation results are shown in Figure 7.

The left sides of Figure 7 show the simulation results of the basic RRT. The middle and right sides show the results of the AAF-RRT algorithms with constant attractive and proportional attractive forces, respectively. Figure 7(a)

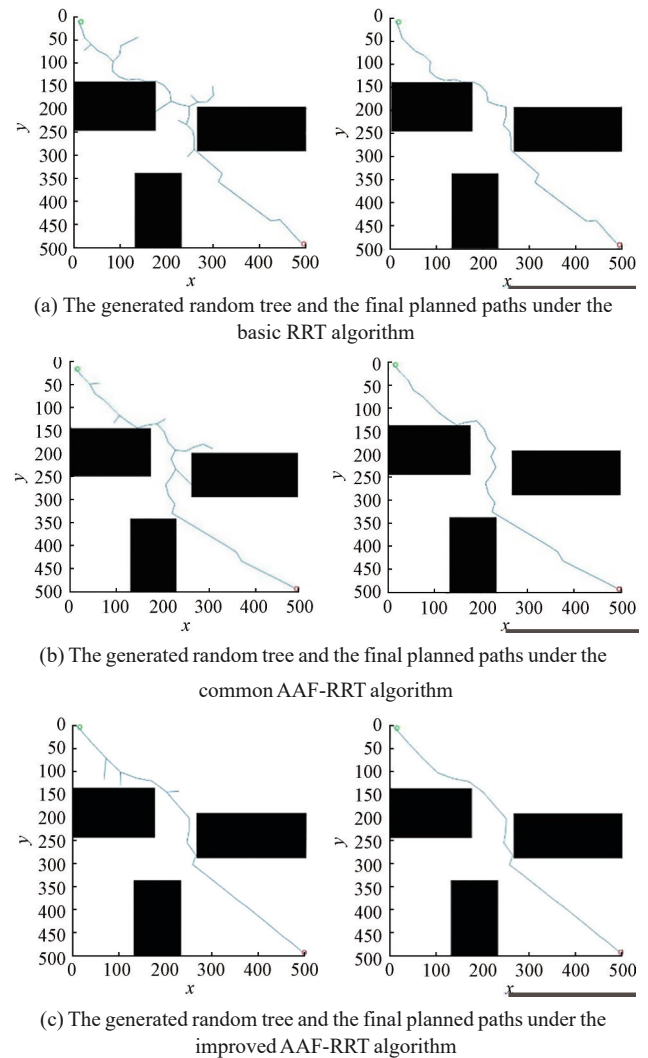


Figure 7 Whole view of the generated random trees and final planned paths in the simple maze under the three algorithms

shows the whole view of the three generated random trees, where the nodes in the tree become less from left to right. This result indicates that AAF-RRT algorithms can reduce the search randomness of the basic RRT and improve the

running efficiency of the algorithms. Conversely, the improved AAF-RRT algorithm with proportional attractive forces behaves better in this kind of utility compared with the common AAF-RRT algorithm. In terms of the final planned paths shown in Figure 7(b), there exists no obvious difference among the three planned paths. However, due to the improvement in the randomness of the RRT algorithms, the planned paths are smooth in the AAF-RRT algorithms.

After conducting the same simulation ten times, the data samples of the processing time and path length under the three algorithms are outputted, which are shown in Figures 8 and 9, respectively. Based on the data shown in the figures, the samples are all non-normal. Therefore, the rank-sum test should be used to preliminarily determine whether the gap between two data samples is evident. With the significance level set to 0.05, the rank-sum test results on the processing time and path length among the three samples are also shown in Figures 8 and 9, where “*” indicates that the overall difference between the two data samples is significant.

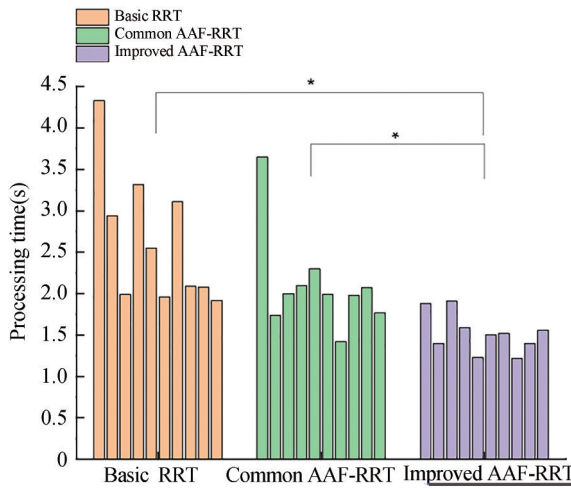


Figure 8 Processing time and rank-sum test results of the three algorithms on the simple maze

Two conclusions can be derived from the simulation results. First, in terms of the processing time, the gap between the common AAF-RRT and basic RRT is not significant, whereas the gap between the improved AAF-RRT and basic RRT and the gap between the improved AAF-RRT and common AAF-RRT are all significant. For the path length, only the gap between the improved AAF-RRT and basic RRT is significant. Then, a further conclusion can be obtained. The improved AAF-RRT behaves the best among the three algorithms in terms of processing time. Common AAF-RRT cannot behave better than the basic RRT in the processing time. Moreover, AAF-RRT behaves better than the basic RRT in the path length, whereas common AAF-RRT still does not have a significant difference

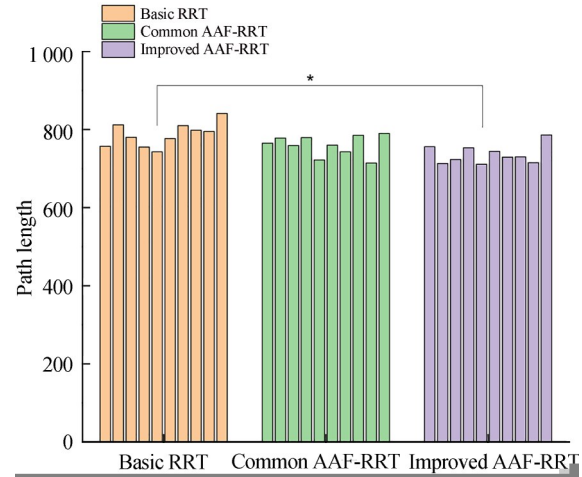


Figure 9 Path length and rank-sum test results of the three algorithms on the simple maze

with the basic RRT.

To sum it up, through the preliminary exploration of the simple maze, the simulation results confirm that the two AAF-RRT algorithms can improve the search randomness of the RRT algorithm. They can also improve the running efficiency and search performance of the basic RRT algorithm. Furthermore, the improved AAF-RRT behaves the best in terms of the running efficiency and search performance among the three algorithms.

5.1.2 Simple maze with obstacles on the line

Based on the former simulations, the simulation environment is upgraded from a simple maze to a simple maze with obvious obstacles on the line. This line refers to the virtual line from the initial point to the goal point in the environment. The aim of this simulation is mainly to determine whether AAF-RRT algorithms could improve the collision-avoidance ability of basic RRT algorithms. Similar to the former simulations, figures on the whole view of the random trees and the final planned paths are successfully outputted, as shown in Figure 10.

Compared with those in Figure 7, the meanings of the patterns stay unchanged in Figure 10. The numbers of nodes in the random trees still turn less from top to bottom in Figure 10. Moreover, the trees tend to be smooth under the AAF-RRT algorithms. Therefore, with the upgrading of the maze, the AAF-RRT algorithms still behave better than the basic RRT algorithm. In terms of the emergence of the obvious obstacles on the line from the initial point to the goal point, the simulation results also show that the AAF-RRT algorithms have a better collision-avoidance ability without the loss of running efficiency. Running efficiency is reflected by the nodes in the random tree.

The corresponding simulation data of the three algorithms are shown in Figures 11 and 12, where the rank-sum test results are also shown. The differences among the

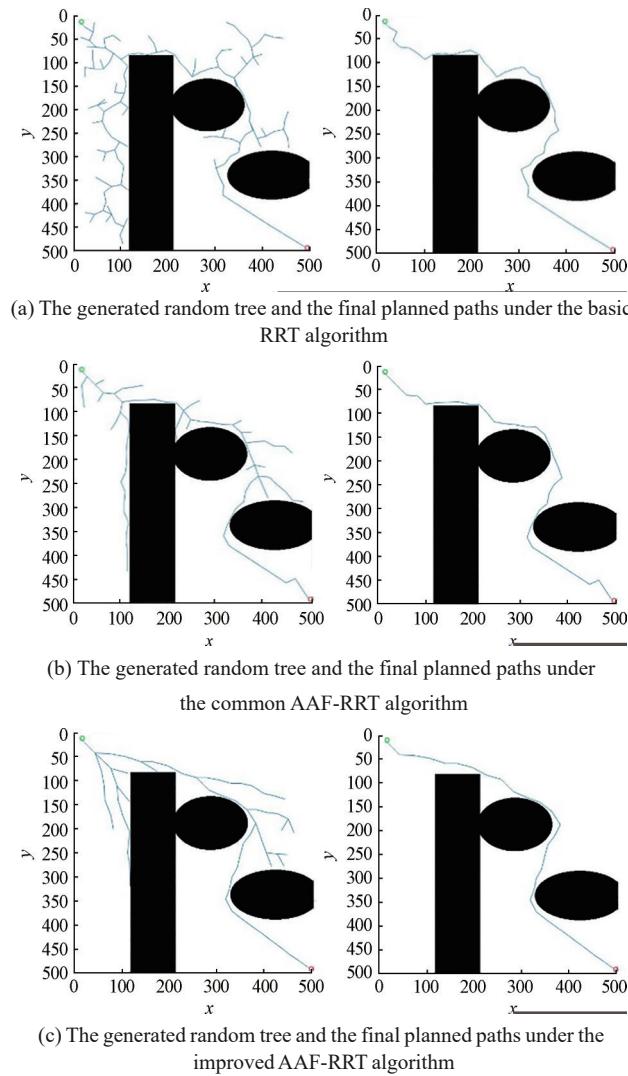


Figure 10 Whole view of the generated random trees and final planned paths in the simple maze with obstacles on the line under the three algorithms

three algorithms are all significant. Therefore, based on the overlook of the data samples, the order of the planning performance from good to bad is as follows: improved AAF-RRT algorithm→common AAF-RRT algorithm→basic RRT algorithm.

Based on the simulation results in the second maze, the AAF-RRT algorithms can improve the collision-avoidance ability of the RRT algorithms without the loss of the running efficiency and searching ability. At the same time, based on the common AAF-RRT algorithm, the improved AAF-RRT algorithm can further improve the search speed and reduce the path length.

5.1.3 Complex maze with narrow passages

The former two simulation results under the two simple mazes have verified that the AAF-RRT algorithms improve the search randomness and searching ability of the

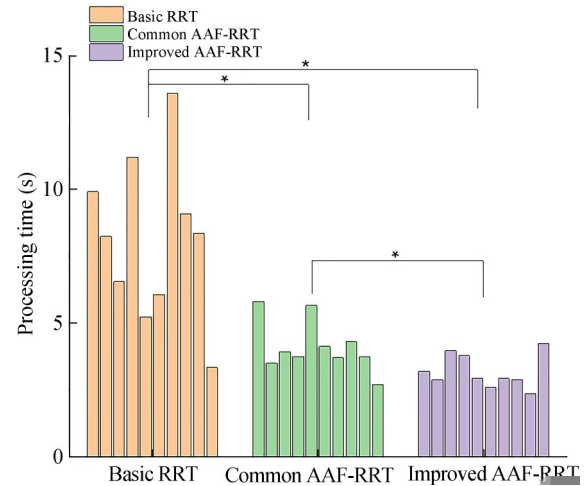


Figure 11 Processing time and rank-sum test results of the three algorithms on the simple maze with obstacles on the line

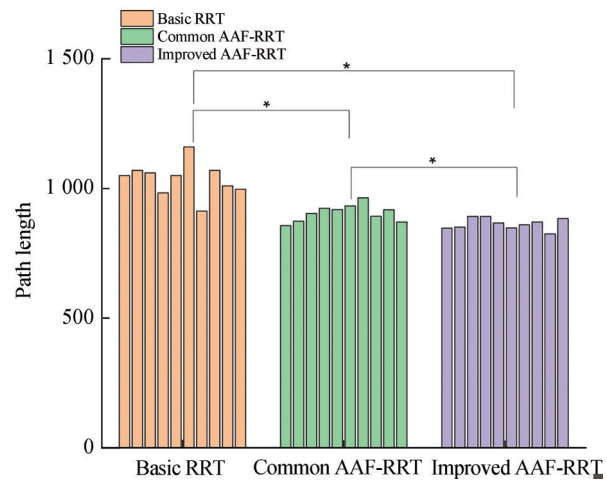


Figure 12 Path length and rank-sum test results of the three algorithms on the simple maze with obstacles on the line

basic RRT algorithms. Considering the poor ability of the RRT algorithms in passing through the narrow passage, a complex maze with a narrow passage is chosen as the simulation environment. This maze contains a narrow passage around the target point. It also includes obstacles on the line from the initial point to the target point. The simulation results of the random trees and planned paths are shown in Figure 11.

Figure 13(a) shows that the random tree under the basic RRT algorithm is very complicated and almost covers the entire maze. During the conducted simulation process under the basic RRT algorithm, the random tree extends in an extremely slow manner when passing the narrow passages. However, from the other two whole views of the random trees under the two AAF-RRT algorithms, the nodes of the random trees are significantly reduced, especially in the improved AAF-RRT algorithm with a proportional attrac-

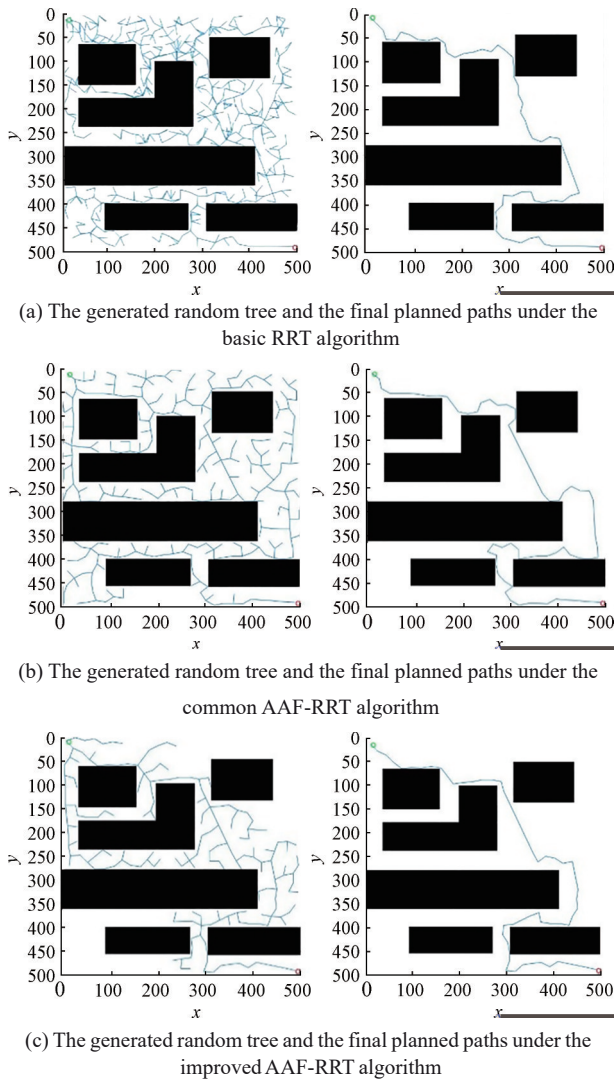


Figure 13 Whole view of the generated random trees and final path in the complex maze with the narrow passage under the three algorithms

tive force. Comparing the two random trees under the two AAF-RRT algorithms, the tendency of the random trees toward the goal point is more obvious under the improved AAF-RRT algorithm. This comparison indicates that the improved AAF-RRT algorithm behaves better in terms of the ability to pass the narrow passage than the common AAF-RRT algorithm.

The simulation results under the three algorithms are shown in Figures 14 and 15. Based on the positions of the “*,” in the searching speed, the order of the performance from good to bad is as follows: improved AAF-RRT algorithm → common AAF-RRT algorithm → basic RRT algorithm. However, in the path length tests, only the difference between the improved AAF-RRT algorithm and the basic RRT algorithm is significant.

Based on the above analysis of the simulation results,

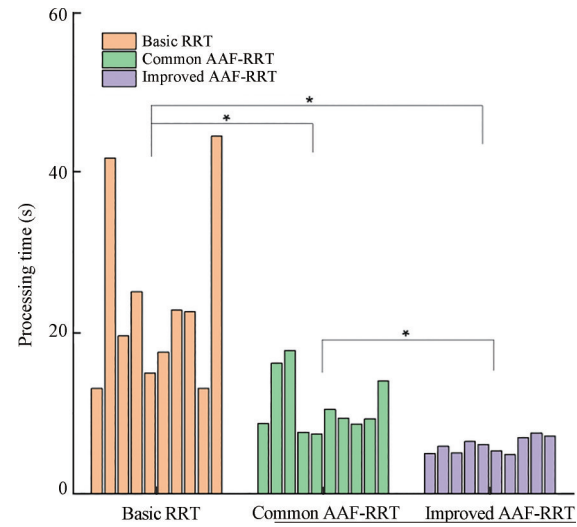


Figure 14 Processing time and rank-sum test results of the three algorithms on the simple maze with obstacles on the line

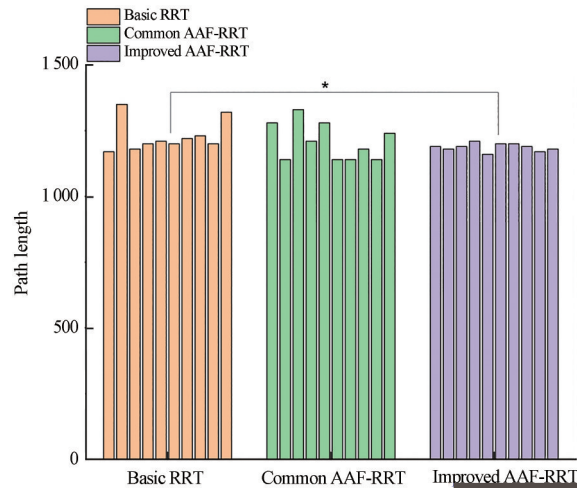


Figure 15 Processing time and rank-sum test results of the three algorithms on the simple maze with obstacles on the line

compared with the common AAF-RRT algorithm, the improved AAF-RRT algorithm can effectively improve the ability to pass the narrow passage of the RRT algorithms better. This kind of improvement has not been found and verified by other researchers.

5.1.4 Local minima trap

One of the challenges in the APF approach is the local minima trap, which may happen when the two APFs, i.e., AAF and ARF, cancel each other out. For example, when an obstacle is located between the robot and target position, a local minima trap may happen. Although the gradient of these local minima points is zero, they are not the target point set in the environment. Moreover, this kind of local minima trap exists in other path planning methods, such as genetic algorithms (Gong and Peng 2002), clustering algorithms (Dash et al 2003; Xu and Franti 2004), and

deep learning methods (Kawaguchi 2016).

In our proposed AAF-RRT algorithms, a local minima trap also exists in some situations. The existence of the local minima trap in APF approaches directly caused the emergence of that in the AAF-RRT algorithms. The emergence of the local minima trap also indicates that the combination of APF and RRT has advantages and disadvantages. Thus, we should highlight its strengths and avoid its weakness as much as possible. Many ways can be done to solve the local minima trap. One method is to introduce a high-level planner, which would make the robot not only use the detection information from the sensors at any time but also maintain the ability of global planning (Dong 2009). Some remedial solutions allow the robot to fall into the local optimum and then use other solutions to fix the problem. For example, a disturbance or backtracking could be added to the local minima trap to jump out of the local minima points (Morris 1993).

Considering that the objective of this study is to explore the search utilization of the improved AAF-RRT algorithm, an easy and simple solution is chosen to escape the local minima trap for the AAF-RRT algorithms. The solution is to manually reduce the relevant attractive factors until the path can get into the goal point. In the two AAF-RRT algorithms proposed in this paper, the relevant attractive factors are k_1 and k_2 in Eqs. (5) and (6). During the previous numerical simulations under the three RRT algorithms, when adopting the k_1 and k_2 of the simulations in the former two simple mazes, the local minima trap emerges in the complex maze under the two AAF-RRT algorithms with $k_1 = 0.2$ and $k_2 = 0.002$, respectively. By using the proposed solution of reducing the two attractive factors, the local minima trap can be effectively escaped when $k_1 = 0.02$ and $k_2 = 0.0001$. The four overviews of the random tree of the emergence and escape of the local minima trap in the mentioned situation are shown in Figures 16 and 17. Figure 16 shows the situation under the common AAF-RRT algorithm, whereas Figure 17 shows the situation under the improved AAF-RRT algorithm. As shown in the two figures, although the attractive factors are manually adjusted and reduced, the proposed solution to escape the local minima trap for the AAF-RRT algorithms is useful and feasible.

5.2 Simulations for the two-layer path planner

A new simulation environment is configured for the two-layer path planner, which is shown in Figure 16. As shown in Figure 18, the blue and green squares denote the initial node and goal node, respectively. The coordinate position of the initial node is set to (2, 2), whereas that of the goal node is set to (49, 24). The black frame in the figure represents the boundary of the environment. The gray patterns denote the obstacles in the environment,

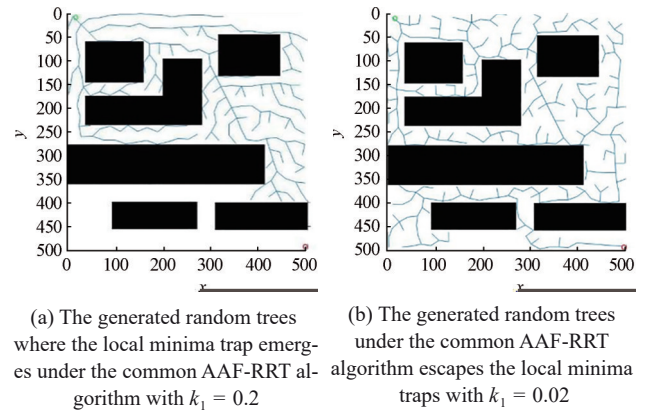


Figure 16 Whole view of the generated random trees where the local minima trap emerges be overcome under the common AAF-RRT algorithm

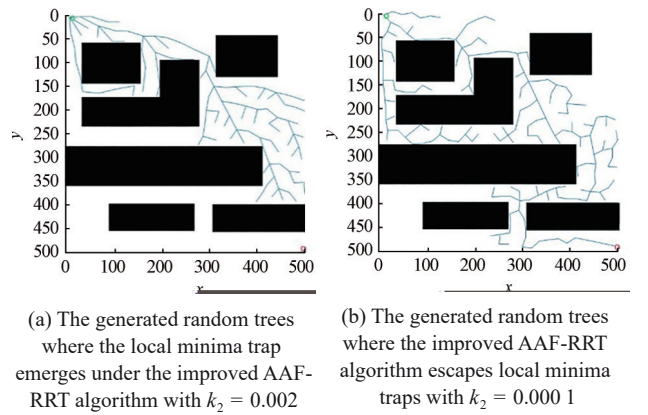


Figure 17 Whole view of the generated random trees where the local minima trap emerges and be overcome under the improved AAF-RRT algorithm

which consist of four circular obstacles and four rectangular obstacles.

5.2.1 Two-layer path planner combined with path smoothing

The improved AAF-RRT is chosen to be the planning algorithm in the two-layer path planner. The simulation process of the two-layer path planner consists of two parts according to the description in Section 4.1. First, the global path would be outputted by the planning algorithm. Second, considering the probability of the sudden obstacles in the environment, the path would be replanned according to the positions of the sudden obstacles.

For the first part, the improved AAF-RRT algorithm would be used to generate a global path. To further shorten the path length, pruning is conducted after generating a path. Then, the planned paths with no pruning and pruning are both outputted, which are shown in Figure 19. The red line denotes the path simulated with no pruning, whereas the blue line represents the path simulated with pruning.

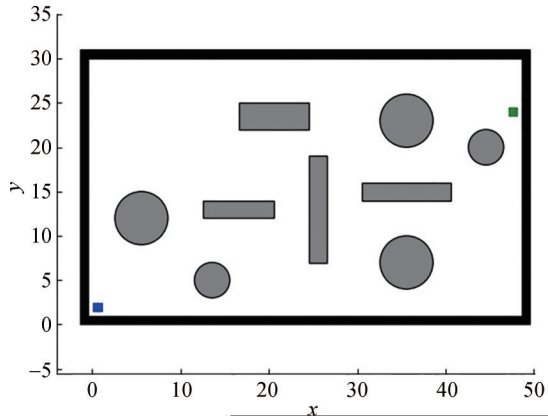


Figure 18 Environmental configuration of the simulations for the two-player path planner

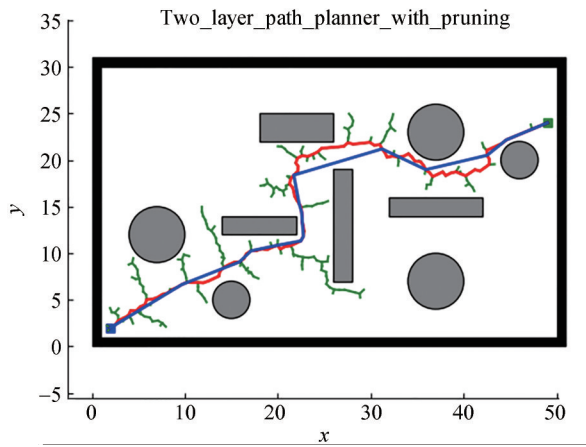


Figure 19 Planned paths without pruning and with pruning

Then, the corresponding path length is reduced from 66.95 to 59.36 by adding pruning.

5.2.2 Two-layer planner added with sudden obstacles

Based on the planned path simulated in Figure 19, we add the sudden obstacles thrice one by one. The fish pattern represents the sudden obstacle. The geometric center of the first obstacle is located in (10, 7), which conflicts with the previously planned path. Therefore, a new path is planned after adding the obstacle, as shown in Figure 20 (a). The meanings of the red and blue lines are unchanged. The second obstacle is added at (15, 22). The obstacle would not conflict with the previously planned path. Therefore, there is no need for the path to be replanned. The previously planned path with pruning would be retained in the red line, as shown in Figure 20(b). Lastly, the third sudden obstacle emerges at (23, 16), conflicting with the previously planned path. Then, the replanned paths are outputted based on the newly added three obstacles, which are also shown in Figure 20(c).

To sum it up, based on the simulation results, the two-layer path planner can not only search a global path with the help of the improved AAF-RRT algorithm and the

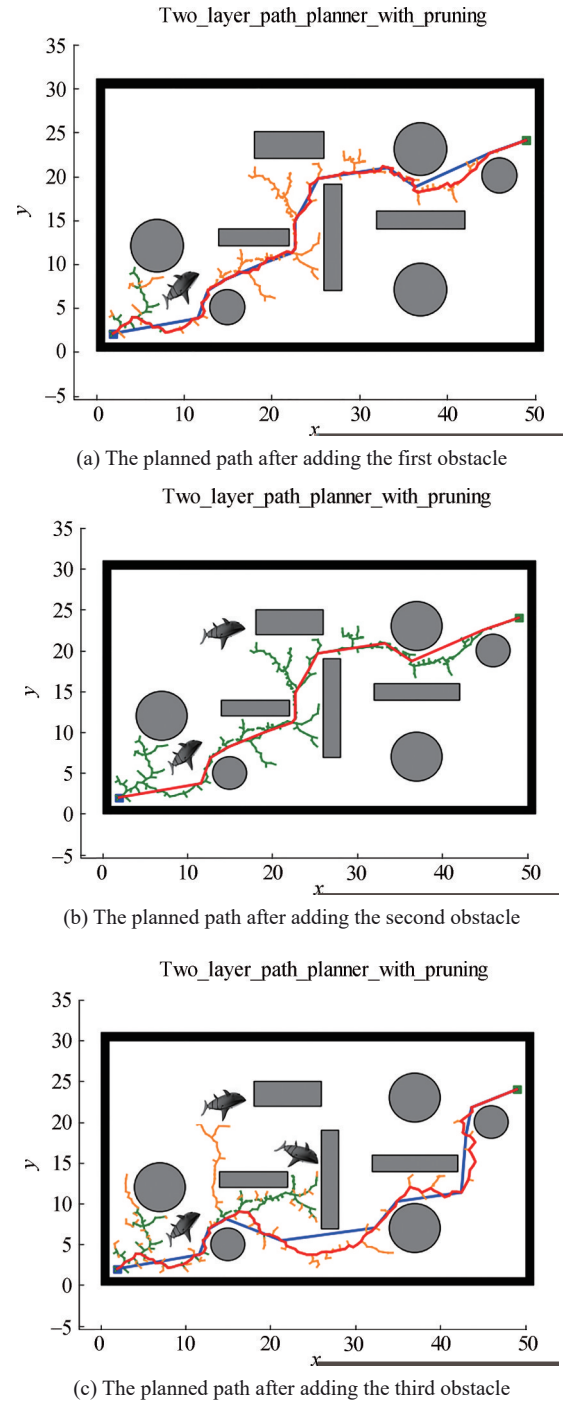


Figure 20 Planned paths after the sudden obstacles thrice one by one pruning process but also effectively replan the path when the sudden obstacle emerges in the environment.

6 Conclusions

In order to improve the search efficiency and collision-avoidance ability of AUVs, this study proposes a specific two-layer path planner based on an improved algorithm.

By integrating the AAF concept with the RRT algorithm, the proposed AAF-RRT algorithm is verified to perform better than the basic RRT algorithm and common AAF-RRT algorithm. After adding several sudden obstacles to the simulation environment, it is confirmed that the two-layer path planner based on the proposed algorithm can quickly react and effectively avoid obstacles. The two verification results provide a basis for the subsequent practical application of the RRT algorithm in AUVs. They also make stable path planning in harsh underwater environments no longer a luxury.

Thus far, the improved AAF-RRT algorithm and two-layer path planner have only been applied to a two-dimensional environment. In the future, to further explore the feasibility of the algorithm and path planner, a 3D simulation will be conducted. Moreover, it would be interesting to explore whether the expansion direction branches in RRT can refer to the true growth direction of branches in nature.

Founding Information Supported by Zhejiang Key R&D Program 558 No. 2021C03157, the “Construction of a Leading Innovation Team” project by the Hangzhou Munic-559 ipal government, the Startup funding of New-joined PI of Westlake University with Grant No. 560 (041030150118) and the funding support from the Westlake University and Bright Dream Joint In-561 stitute for Intelligent Robotics.

References

- Cai W, Wu Y, Zhang M (2020) Three-dimensional obstacle avoidance for autonomous underwater robot. *IEEE Sensors Letters* 4(11): 1-4. DOI: 10.1109/LESENS.2020.3034309
- Carsten J, Ferguson D, Stentz A (2006) 3D Field D: Improved path planning and replanning in three dimensions. *IEEE/RSJ International Conference on Intelligent Robots & Systems*
- Dash AK, Chen IM, Song HY, Yang G (2003) Singularity-free path planning of parallel manipulators using clustering algorithm and line geometry. *IEEE International Conference on Robotics & Automation*
- Dong HK (2009) Escaping route method for a trap situation in local path planning. *International Journal of Control Automation & Systems* 7(3): 495-500. DOI: 10.1007/s12555-009-0320-7
- Dong J (2013) Research on firefly algorithm and its application in path planning of underwater vehicle. Master thesis, Harbin Engineering University (in Chinese)
- Gasparetto A, Boscaroli P, Lanzutti A, Vidoni R (2015) Path planning and trajectory planning algorithms: A general overview. Springer International Publishing
- Gong JF, Peng SX (2002) Approach to Robot path planning based on numerical potential field and genetic algorithm. *Journal of Tianjin University (Science and Technology)*, (in Chinese) DOI: 10.1007/s11769-002-0038-4
- Gu GC, Fu Y, Liu HB. (2005). Path planning of AUV based on genetic simulated annealing algorithm. *Journal of Harbin Engineering University*. (in Chinese) DOI: 10.3969/j. issn. 1006-7043. 2005.01.018
- Hart PE, Nilsson N, Raphael B (1972) A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science & Cybernetics* 4(2): 28-29. DOI: 10.1145/1056777.1056779
- Jr J, Lavelle SM (2000) RRT-connect: an efficient approach to single-query path planning. *Proceedings of the 2000 IEEE International Conference on Robotics and Automation, ICRA 2000, San Francisco, USA*
- Kawaguchi K (2016) Deep learning without poor local minima.
- Khatib O (2003) Real-time obstacle avoidance for manipulators and mobile robots. *IEEE International Conference on Robotics and Automation*
- Lavelle SM, Kuffner JJ (2000) Rapidly-exploring random trees: progress and prospects. *Algorithmic & Computational Robotics New Directions*. DOI: 10.1201/9781439864135-43
- Li YS (2012) Research on the environment modeling method of the AUV path planning system. Master thesis, Harbin Engineering University (in Chinese)
- Liu C, Han J, An K (2017) Dynamic path planning based on an improved RRT algorithm for roboCup robot. *Robot* 39(1): 8-15. (in Chinese) DOI: 10.13973/j.cnki.robot.2017.0008
- Liu C, Zhao Y, Gao F, Liu L (2015) Three-dimensional path planning method for autonomous underwater vehicle based on modified firefly algorithm. *Mathematical Problems in Engineering* 2015 (Pt.20): 1-10. DOI: 10.1155/2015/561394
- Liu G, Liu P, Mu W, Wang S (2016) A path optimization algorithm for AUV using an improved ant colony algorithm with optimal energy consumption. *Hsi-An Chiao Tung Ta Hsueh/Journal of Xi'an Jiaotong University*, 50(10): 93-98. (in Chinese) DOI: 10.7652/xjtub201610014
- Lozano-Pérez T (1990) Spatial planning: A configuration space approach. *Autonomous Robot Vehicles* 108-120. DOI: 10.1109/TC.1983.1676196
- Morris P (1993) The Breakout method for escaping from local minima. *Proceedings of the 11th National Conference on Artificial Intelligence*, Washington, DC, USA
- Orozco-Rosas U, Montiel O, Sepúlveda R (2019a). Mobile robot path planning using membrane evolutionary artificial potential field. *Applied Soft Computing*. DOI: 10.1016/j.asoc.2019.01.036
- Orozco-Rosas U, Picos K, Ross O (2019b) Hybrid path planning algorithm based on membrane pseudo-bacterial potential field for autonomous mobile robots. *IEEE Access* 7(1): 156787-156803. DOI: 10.1109/ACCESS.2019.2949835
- Sariff N, Buniyamin N (2006) An overview of autonomous mobile robot path planning algorithms. *Conference on Research & Development*
- Song JZ, Dai B, Shan EZ, He HG (2010) An improved RRT path planning algorithm. *Acta Electronica Sinica* 38: 225-228. (in Chinese) DOI: CNKI:SUN:DZXU.0.2010-S1-040
- Sun Y, Wang L, Wu J, Ran X (2020) A general overview of path planning methods for autonomous underwater vehicle. *Ship Science and Technology*, 42(4): 1-7. (in Chinese) DOI: CNKI: SUN:JCKX.0.2020-07-002
- Tanakitkorn K, Wilson PA, Turnock SR, Phillips AB (2014) Grid-based GA path planning with improved cost function for an over-actuated hover-capable AUV. *AUV 2014*
- Volpe R, Khosla P (1990) Manipulator control with superquadric artificial potential functions: theory and experiments. *Systems Man & Cybernetics IEEE Transactions* 20(6): 1423-1436. DOI: 10.1109/21.61211
- Wang H, Wu X, Shi X (2008) Global planning path method of AUV based on ant colony optimization Algorithm. *Shipbuilding of China*,

- 49(2): 88-93.(in Chinese) DOI: 10.1080/02699930701273823
- Wang LL, Sui ZZ, Pu ZQ, Liu Z, Yi JQ (2020) An Improved RRT algorithm for multi-robot formation path planning . *Acta Electronica Sinica* 48: 60-67. (in Chinese) DOI: 10.3969/j.issn.0372-2112.2020.11.007
- Wang Q (2014) Research on rapidly-exploring random trees based global path planning and its application. Master thesis, National University of Defense Technology. (in Chinese)
- Xu H, Li Y (2010) An immune genetic algorithm for AUV local path planning. *International Society of Offshore and Polar Engineers*
- Xu M, Franti P (2004) A heuristic K-means clustering algorithm by kernel PCA. 2004 International Conference on Image Processing, ICIP' 04.
- Xu YR, Yao YZ (2008) Research on AUV global path planning considering ocean current. *Shipbuilding of China* 49(4): 109-114. (in Chinese) DOI: 10.3969/j.issn.1000-4882.2008.04.014
- Zhu DQ, Sun B, Li L (2015) Algorithm for AUV's 3-D path planning and safe obstacle avoidance based on biological inspired model. *Control & Decision* 30(5): 798-806. (in Chinese) DOI: 10.13195/j.kzyjc.2014.0339