electronics

Article

The Optimization and Application Research of the RRT-APF-Based Path Planning Algorithm

Bolin Zhang [†] and Changyong Li [*,†]

College of Mechanical Engineering, Xinjiang University, Xinjiang 830017, China; 107552101375@stu.xju.edu.cn
[*] Correspondence: lcy@xju.edu.cn; Tel.: +86-159-9944-9306
[†] These authors contributed equally to this work.

Abstract: To address the shortcomings of the original rapidly-exploring random tree (RRT) algorithm, such as long and non-smooth paths, slow convergence to the goal region, and limited adaptability in dynamic environments, this paper proposes a hybrid path planning method combining the artificial potential field (APF) approach with the RRT algorithm. This integrated approach leverages the strengths of both methods to achieve efficient, collision-free path planning in both two-dimensional and three-dimensional environments. The algorithm overcomes the local minima problem inherent in APF while maintaining the RRT's efficiency in high-dimensional spaces and complex environments. A dynamic adjustment strategy is introduced to adapt to specific application scenarios and varying environmental complexity. Additionally, Bezier curve fitting is applied to smooth the resulting paths. Simulations conducted in various environments demonstrate the effectiveness of the proposed method, highlighting its efficiency and robustness in generating collision-free paths. Compared to the original RRT algorithm, the proposed method reduces path length by 13.4% to 24.9% and decreases search time by 9.8% to 56.5%, improving both path quality and planning efficiency.

Keywords: path planning; improve the RRT algorithm; artificial potential field method

Citation: Zhang, B.; Li, C. The Optimization and Application Research of the RRT-APF-Based Path Planning Algorithm. *Electronics* **2024**, *13*, 4963. https://doi.org/10.3390/electronics13244963

Academic Editor: Mauro Tropea

Received: 31 October 2024
Revised: 10 December 2024
Accepted: 14 December 2024
Published: 17 December 2024

1. Introduction

With the continuous development of intelligent technology and robot applications, the requirements for path planning algorithms have become more and more stringent. In complex working environments, robots need to avoid multiple obstacles and find an efficient and safe path, which poses a high challenge to the performance of planning algorithms [1,2]. Traditional path planning methods, such as those based on graph search, intelligent bionics, and sampling [3], have their own advantages. Among them, the rapid exploration random tree (RRT) algorithm is widely used due to its superior performance in high-dimensional spaces [4]. However, the RRT algorithm suffers from the shortcomings of long paths, redundant nodes, slow convergence, etc., and it is especially difficult to generate high-quality paths in complex environments [5]. To overcome these problems, recent studies have proposed various improvement algorithms such as RRT-Connect [6], RRT* [7], Informed RRT* [8], etc. However, most of these improvements have trade-offs between computational efficiency and path quality, and some of the algorithms still fall short of performance in complex obstacle environments. For this reason, this paper proposes an improved RRT algorithm based on artificial potential field guidance and dynamic step size. The method improves the directionality of tree expansion through the guidance of gravitational and repulsive fields, reduces the generation of invalid nodes by combining with a dynamic spatial sampling strategy, and introduces pruning and smoothing after path generation to improve the compactness and feasibility of the paths, so as to realize a more efficient path planning in a variety of environments.

Gu et al. [7] proposed the PI-DP-RRT algorithm, which integrates AIS data to guide the sampling region and applies an enhanced Douglas–Peucker (DP) algorithm to improve convergence speed and path smoothness, though its adaptability in dynamic environments remains limited. Chai et al. [9] introduced the RJ-RRT algorithm, which enhances search efficiency in narrow passages through a greedy sampling space reduction strategy and subtree fusion but performs less effectively in dense obstacle environments compared to traditional RRT. Gao et al. [10] developed the BCRRT* algorithm, which leverages bidirectional path planning and caching to enhance real-time performance in underwater environments, though it demands significant storage resources. Wu et al. [11] combined RRT and PRM, utilizing adaptive step size and fan-shaped goal orientation strategies to improve search efficiency in complex environments, albeit with increased computational costs. Yang et al. [12] proposed the ACO-RRT algorithm, which uses pheromone-guided tree expansion based on ant colony optimization (ACO) to accelerate convergence and improve path quality, though it incurs high iterative computation costs and is prone to local optima. Wang et al. [13] introduced the bidirectional–unidirectional RRT extension algorithm, which avoids the two-point boundary value problem (BVP) to simplify planning in complex scenarios but relies heavily on the effectiveness of bidirectional search. Zhao et al. [14] proposed the AODA-PF-RRT* algorithm, which dynamically adjusts obstacle density and incorporates artificial potential fields (APF) to improve path smoothness in complex environments, though its sensitivity to environmental changes may increase computational overhead. Finally, Guo et al. [15] developed the HPO-RRT* algorithm, which employs time-constrained RRT* for path planning under dynamic threats, achieving efficient path optimization but requiring frequent re-planning, leading to additional computational costs.

In summary, established studies have improved the path planning performance of the RRT algorithm in different aspects, but there is still room for improvement in terms of path smoothing and computational cost, sampling strategy robustness, and exploration efficiency in complex spaces. To address these issues, this paper proposes an improved RRT algorithm. The improvement of the algorithm is as follows:

1. Determine the step size of the current iteration according to the distance between the current iteration point and the nearest obstacle in the environment in order to reduce the number of iterations and improve the efficiency of the algorithm;
2. Combining the RRT and APF algorithms and adjusting the parameters of the potential field function according to the characteristics of the environment to make up for some of the shortcomings of the RRT algorithm in terms of obstacle avoidance capability, path quality, and convergence speed, so as to improve the effectiveness of path planning;
3. Use iterative endpoint fitting algorithm combined with angle threshold detection to remove nodes in the original path, this method can reduce redundant nodes and smooth the original path.

2. Principle and Analysis of Traditional RRT Algorithm

Rapidly exploring random tree (RRT) is a classical path planning algorithm commonly used for robot path planning problems in high-dimensional unstructured environments [11]. The algorithm continuously expands the tree structure through random sampling and gradually explores the whole space from the starting point until a feasible path from the starting point to the goal point is found. The core idea of RRT is to cover the feasible region of the space quickly, and thus it performs well in dealing with complex environments. The basic steps of the RRT algorithm are as follows: firstly, the root node of the tree is initialized at the starting point; subsequently, a randomly generated sample point is generated in the space. Next, the nearest node from the tree to that sample point is found and expanded by one step along that direction to generate a new node. Then, collision detection is performed to ensure that the path does not intersect an obstacle. If there are no collisions, the new node is added to the tree. This process is repeated until a node in the tree is close to the goal point or reaches a preset number of iterations, thus generating a feasible path from the starting point to the goal point.

Although the RRT algorithm has a better exploration ability in complex environments, it also has some obvious drawbacks. Firstly, the generated paths are usually long and not smooth, requiring subsequent smoothing. Second, as the spatial dimension increases, the sampling efficiency decreases significantly, leading to an increase in computation time. In addition, due to the random sampling strategy, the algorithm explores the target region less efficiently and converges slowly.

3. Improved RRT Algorithm Principle and Analysis

In order to address the shortcomings of the original RRT algorithm, such as long and unsmooth paths, slow convergence in the target region, and poor adaptability to the dynamic environment, this paper proposes an improved RRT algorithm and introduces optimization methods, such as dynamic step-size adjustment, dynamic weighting strategy, and path smoothing processing into the algorithm flow. Through these improvements, the algorithm improves the quality and planning efficiency of the paths on the basis of ensuring the feasibility of the paths. Specifically, this improved RRT algorithm dynamically calculates the step length in each iteration according to the environment, so that the path is more accurate when approaching the target or obstacle; at the same time, the potential field function of the artificial potential field method is introduced to influence the generation of nodes, and the algorithm's orientation and obstacle avoidance ability are enhanced by adjusting the weights of the attractive force and the repulsive force. In addition, after generating the path, a smoothing process is used to remove redundant points and optimize the curves so that the path is more suitable for the actual robot control system. The flowchart, shown in Figure 1, demonstrates the specific steps and execution logic of the improved algorithm.

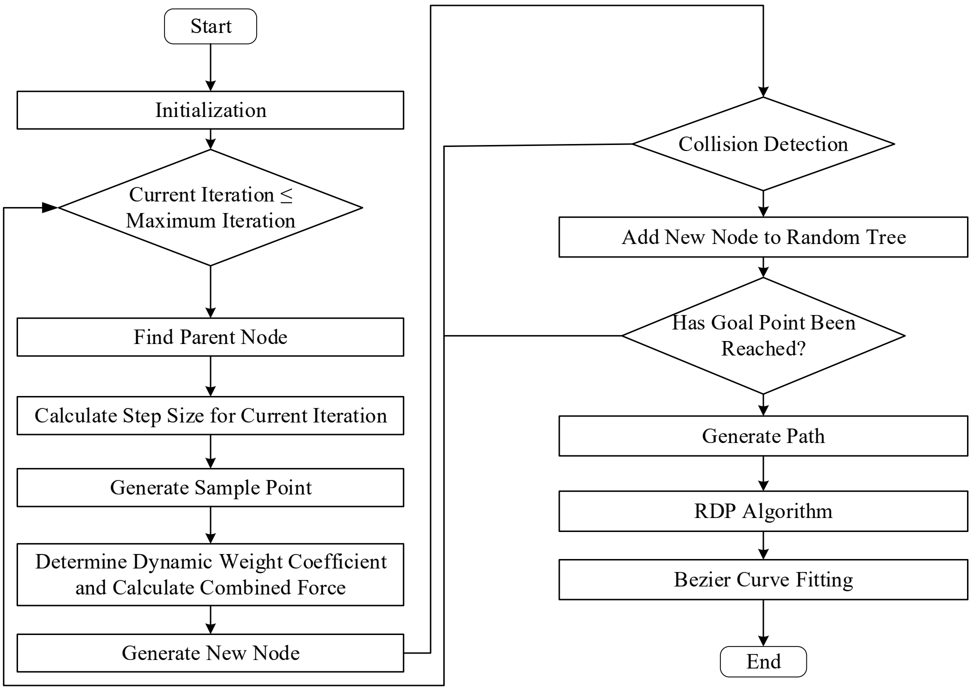

Start
Initialization
Current Iteration ≤ Maximum Iteration
Find Parent Node
Calculate Step Size for Current Iteration
Generate Sample Point
Determine Dynamic Weight Coefficient and Calculate Combined Force
Generate New Node
Collision Detection
Add New Node to Random Tree
Has Goal Point Been Reached?
Generate Path
RDP Algorithm
Bezier Curve Fitting
End

Figure 1. Flowchart of improved RRT algorithm.

1. Initialize the coordinates of the start point X_{start} and the goal point X_{goal}, the obstacle parameters, the initial *step* length $step_0$, the initial potential field weights kp_0 and $krep_0$, the distance threshold ε, the tolerance t, the maximum iteration number $iter_{\text{MAX}}$ and other information.
2. Determine whether the current iteration exceeds the maximum number of iterations, if it does exceed, the path planning fails; if it does not exceed, the number of iterations

plus one, and then calculate the step length of this iteration according to the position of the node of the previous generation, and generate a new sampling point. The rule of random point generation is that if the distance of the parent node to the target point is less than the distance of the target point to the nearest obstacle, then the target point will be set as the new node. Otherwise, a random sample point is a generated point; after the random point generation, the weight coefficients of the potential field function are calculated, and then the synthetic total potential field force is calculated. Finally, the new node new_{point} is generated according to the random point position and potential field force.

3. Connect the new node new_{point} with the parent node $parent_{\text{point}}$, and determine whether the collision with the obstacle and its influence range occurs, if the collision occurs, the new node is discarded; if no collision occurs, the new node is added to the random tree.
4. After generating the new node, judge whether the distance between the new node and the target point is less than the set tolerance t. If it is less than or equal to the set tolerance t, terminate the algorithm and return to the collision-free path, and the path planning is completed; otherwise, return to step 2.
5. The generated path is sequentially node deletion and curve fitting, a smooth curve is obtained, and the algorithm ends.

The node expansion schematic is shown in Figure 2, the position X_{new} of the new node is determined by the combined result of the direction of the combined force and the random point guidance, and its expansion formula can be expressed as:

$$X_{\text{new}} = X_{\text{parent}} + \lambda \cdot \frac{F_{\text{total}}}{\|F_{\text{total}}\|} \cdot step + (1-\lambda) \cdot \frac{X_{\text{rand}} - X_{\text{parent}}}{\|X_{\text{rand}} - X_{\text{parent}}\|} \cdot step \tag{1}$$

In the formula, λ is the weight, and in the text, λ is set to 0.5. This can balance the goal orientation and exploratory nature of the algorithm, avoid getting stuck in local optima, and increase the flexibility of path generation.

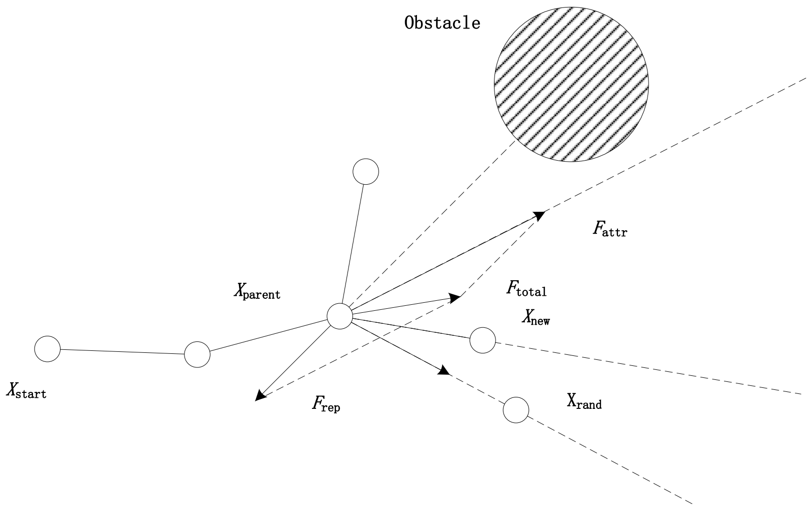


Figure 2. Schematic diagram of node expansion.

3.1. Dynamic Step Size

In the original RRT algorithm, a fixed-length step is extended from an existing node of the tree (e.g., the starting point) to a randomly sampled target location. The core steps of the expansion strategy are as follows:

$$q_{\text{new}} = q_{\text{near}} + step \cdot \frac{q_{\text{rand}} - q_{\text{near}}}{\|q_{\text{rand}} - q_{\text{near}}\|} \tag{2}$$

In original RRT, the expansion of nodes is affected by the step size, and in path planning algorithms, the choice of step size directly affects the search efficiency of paths and obstacle avoidance performance. The fixed step size is an important characteristic of the original RRT algorithm, which maintains a constant size during each path expansion. This strategy simplifies the computational process of path generation to a certain extent and also has certain limitations. For this reason, the method of dynamic adjustment of the initial step size *initial_step* is introduced to cope with different environmental complexity and improve the adaptability of the algorithm, and the step size adjustment formula is as follows:

$$step = \begin{cases} \frac{initial\_step}{3+N_{\text{obstacles}}}, & D(q) < Q^{*} \\ initial\_step, & D(q) \geq Q^{*} \end{cases} \tag{3}$$

The initial step length *initial_step* takes the value of 40 mm, $N_{\text{obstacles}}$ is the number of obstacles near the current node, the range is taken as a sphere with a radius of 100 mm, $D(q)$ is the distance from the current position to the nearest obstacle, and $Q*$ is the obstacle's action distance threshold. The formula adopts a segmentation strategy to dynamically adjust the step length according to the distance between the current node and the obstacle; when $D(q) < Q*$, the node is close to the obstacle, decreasing the step length, and the more obstacles in the vicinity, the smaller the step length is; when $D(q) \geq Q*$, the node is farther away from the obstacle, keeping the initial step length.

Relying on this method to obtain the step length ensures the accuracy of obstacle avoidance; keeping the step length in the area with few obstacles improves the search efficiency; decreasing the step length also makes the path smoother in complex environments; when the number of obstacles in the environment increases, the step length shrinks more, avoiding the collision and improving the obstacle avoidance ability.

The dynamic step size adjusts the expansion step based on the proximity to obstacles. When the node is near an obstacle, the step size is reduced to ensure better collision avoidance. In areas with fewer obstacles, the step size remains larger, improving search efficiency. This dynamic adjustment improves obstacle avoidance accuracy and path smoothness, especially in complex environments with high obstacle density.

3.2. Dynamic Weighting Strategy

The artificial potential field method (APF) is a classical path planning algorithm whose basic idea is to construct virtual gravitational and repulsive fields to transform the robot motion problem into a motion problem under the action of a force field. The robot is driven by the gravitational force in the force field to move toward the target point, and at the same time, it is driven by the repulsive force to avoid the obstacles. In the traditional APF, the weights of the gravitational and repulsive forces are set statically, and the combined force field $U(q)$ is formulated as the sum of the gravitational field $U_{\text{att}}(q)$ and the repulsive force field $U_{\text{rep}}(q)$:

$$U(q) = U_{\text{att}}(q) + U_{\text{rep}}(q) \tag{4}$$

The common gravitational field formula is:

$$U_{\text{att}}(q) = \begin{cases} \frac{1}{2}\,\xi\, d^{2}(q, q_{\text{goal}}), & d(q, q_{\text{goal}}) \leq d^{*}_{\text{goal}} \\ d^{*}_{\text{goal}}\,\xi\, d(q, q_{\text{goal}}) - \frac{1}{2}\,\xi\,(d^{*}_{\text{goal}})^{2}, & d(q, q_{\text{goal}}) > d^{*}_{\text{goal}} \end{cases} \tag{5}$$

The common repulsive field function is:

$$U_{\text{rep}}(q) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{D(q)} - \frac{1}{Q^{*}}\right)^{2}, & D(q) \leq Q^{*} \\ 0, & D(q) > Q^{*} \end{cases} \tag{6}$$

In the above equation, ξ and η are the gravitational gain constants and repulsive gain constants, respectively, and changing these two constants can help to overcome the local minima or oscillations that may be caused by the static potential field, so the gravitational field gain coefficients, $\xi(q)$, and the repulsive field gain coefficients, $\eta(q)$, are introduced and are adjusted according to the current node-to-target-point distance as well as the node-to-surrounding obstacle distance. The modified gravitational field gain coefficient $\xi(q)$ and repulsive field gain coefficient $\eta(q)$ are as follows:

$$\xi(q) = \begin{cases} \xi_{\max}, & d(q, q_{\text{goal}}) \leq d_{\min} \\ \xi_{\min} + (\xi_{\max} - \xi_{\min}) \cdot [1 - \tanh(\beta \cdot d(q, q_{\text{goal}}) - d_{\min})], & d_{\min} < d(q, q_{\text{goal}}) \leq d^*_{\text{goal}} \\ \xi_{\min}, & d(q, q_{\text{goal}}) > d^*_{\text{goal}} \end{cases} \tag{7}$$

$$\eta(q) = \begin{cases} \eta_{\max}, & D(q) \leq D_{\min} \\ \eta_{\min} + (\eta_{\max} - \eta_{\min}) \cdot \exp\left(-\alpha \cdot (D(q) - D_{\min})^2\right) \cdot \frac{\sum_{i=1}^{n} \frac{1}{D_i(q)}}{n}, & D_{\min} < D(q) \leq D_{\max} \\ \eta_{\min}, & D(q) > D_{\max} \end{cases} \tag{8}$$

The two formulas first specify the maximum and minimum gain ($\xi_{\max}, \xi_{\min}, \eta_{\max}, \eta_{\min}$), which are taken as 20, 150, 2, and 1, respectively, in this paper; the gain coefficient is changed by judging the distance $d(q, q_goal)$ between the current node and the target point, as well as the critical distance d_min from the target point, and β is the rate of controlling the gravitational gain change with distance, which is taken as 1.5 in this paper; the tanh function is used in the formula of the gain coefficient of the gravitational field for the smooth transition, which ensures that the gravitational gain increases as the node approaches the target point increases gradually, thus avoiding sudden drastic gravitational changes and improving the smoothness of path planning; while the exponential decay function is used for the repulsive gain to ensure that the node avoids obstacles quickly when it is close to the obstacle and the repulsive force decays smoothly when it is far away from the obstacle. In addition, the average distance inverse $\frac{\sum_{i=1}^{n} \frac{1}{D_{\text{i}}(q)}}{n}$ is introduced to be able to take into account the effects of multiple obstacles, making the repulsive field smarter in complex environments.

The dynamic weighting strategy adjusts the gravitational and repulsive force coefficients in the APF based on the node's distance to the target and obstacles. This dynamic adjustment helps avoid local minima and enhances obstacle avoidance, making the algorithm more robust in complex environments.

3.3. Path Smoothing

The algorithm plans a collision-free path, but the path is not smooth due to the presence of turning points and redundant points. In order to make the path smooth, it is necessary to remove redundant points and points with too many turns and keep the points that make the path smoother. Therefore, the Ramer–Douglas–Peucker (RDP) algorithm is used first to remove the redundant nodes, and then the points with too many turns are removed by the angle threshold detection method, and then the simplified path is fitted with a Bézier curve to obtain a smooth path.

The Ramer–Douglas–Peucker (RDP) algorithm, also known as the iterative endpoint fitting algorithm, is an algorithm that reduces the number of points in a curve consisting of a set of connected points, preserving the basic shape of the curve and reducing the number of connected points. The algorithm first sets a distance threshold, then selects and connects the start and end points to find the point furthest from the line; if the distance is greater than a predetermined threshold, the point is retained and the path is recursively simplified; if the distance is less than the threshold, all intermediate points are removed. The process is repeated until all points in the path meet the simplification conditions, and the simplified path is finally generated while retaining its main features. The principle is shown in Figure 3 below:

The first row shown in Figure 3 consists of an initial path composed of initial connection points with a distance threshold ε set.

Take the first and last two points as the start and end points, and find the connection point P with the farthest distance from the straight line formed by the two points, take the distance from point P to the straight line as d, judge the size of the distance threshold ε and the straight line d over the first and last two points, if $\varepsilon < d$, then the connection point P is retained; on the contrary, delete all the points between the starting point and the endpoint and take the new path for the start point to the endpoint, as shown in the second row of the figure.

If the connection point P is retained, recursively apply the RDP algorithm to the two curves from the start point to P and P to the endpoint, respectively, until there is no connection point between the new first and last points, as shown in rows three to five of the figure.

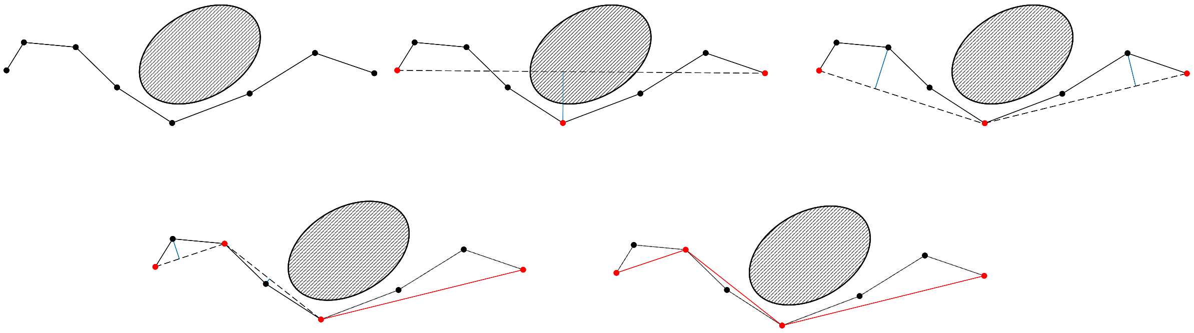


Figure 3. RDP algorithm steps.

4. Simulation Experiment Design

In order to verify the effectiveness of the improved RRT-APF algorithm, MATLAB software version # R2019a was used to carry out simulation experiments in a two-dimensional plane and three-dimensional space, respectively, to compare the environment of simple obstacles with that of complex obstacles, and to analyze the degree of improvement of the indexes such as the path length and the number of nodes under different environments. In the two-dimensional plane experiment, a two-dimensional map of 1000 mm × 1000 mm was constructed first, and the coordinates of the starting point were set to (0,0), and the coordinates of the target point were set to (999,999). Two rectangular obstacles and two circular obstacles were set up in the map of the simple environment, and five rectangular obstacles and five circular obstacles were set up in the complex environment and in the three-dimensional space experiment. In the experiment, a three-dimensional space with dimensions of 1000 mm × 1000 mm × 1000 mm was constructed, and the coordinates of the obstacles and target points were defined within this space in three-dimensional space and the coordinates of the start point were set as [0,0,0] and the coordinates of the target point set as [999,999,999]. One rectangular obstacle, one cylindrical obstacle, and one sphere obstacle were set in the space of the simple environment; in the experiment of the complex three-dimensional space, four rectangular obstacles, four cylindrical obstacles, and four sphere obstacles.

In simulation experiments, reasonable selection and setting of algorithm parameters are crucial for algorithm performance. For the algorithm, parameters such as step length, maximum number of iterations, target threshold, force field parameters, and obstacle buffer are set. The step length determines the distance of each tree node expansion, and the initial step length of both the original RRT and the improved RRT algorithms in the simulation experiments is 40 mm. The maximum number of iterations is set to 5000, the target threshold is set to 40 mm, and the algorithm determines that the path has been found when the distance between the node and the target node is less than this value. The force field parameters include the attraction coefficient kattr = 1.0 and the repulsion coefficient krep = 100.0, the attraction guides the node towards the target while the repulsion keeps the node away from the obstacle. In order to avoid the generated paths being too close to the obstacles, a 40 mm obstacle buffer is set in this paper to ensure the safety of the paths.

In a simple two-dimensional environment, obstacle parameters as shown in Table 1. There are four obstacles in the environment, and the experiment is repeated 15 times to compare the performance of the two algorithms in a simple environment; in the path length comparison experiment, as shown in Figure 4, the improved RRT algorithm generates a path length that is basically shorter than the original RRT algorithm, and the improved RRT algorithm generates an average path length of 1857.15 mm, and the average path length generated by the original RRT algorithm is 1882.88 mm, which is reduced by about 16%. In the comparison experiment of search time, as shown in Figure 5, there is a substantial reduction in the search time of the improved RRT algorithm, and the average search time of the improved algorithm is 1.57 s, and the search time of the original RRT algorithm is 3.60 s, which is a reduction of about 56.5% on average.

Table 1. Parameters of 2D simple environmental obstacles.

Serial Number	Rectangle (Length × Width, mm)	Centroid Coordinates (mm)	Circle Radius (mm)	Circle Center Coordinates (mm)
1	300 × 100	−400,200	96	−100,200
2	200 × 350	−400,575	150	−800,500

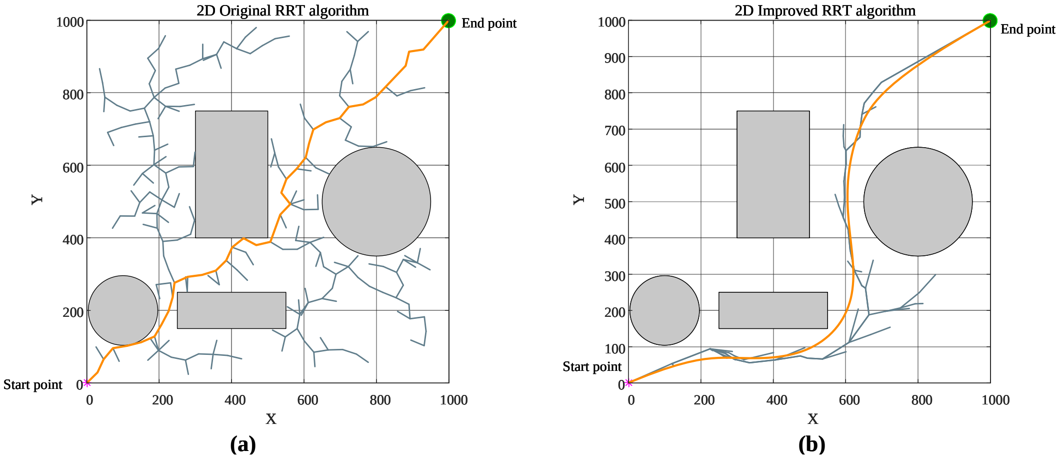


Figure 4. Effect of the original RRT algorithm (**a**) and the improved RRT algorithm (**b**) in a two-dimensional simple environment.

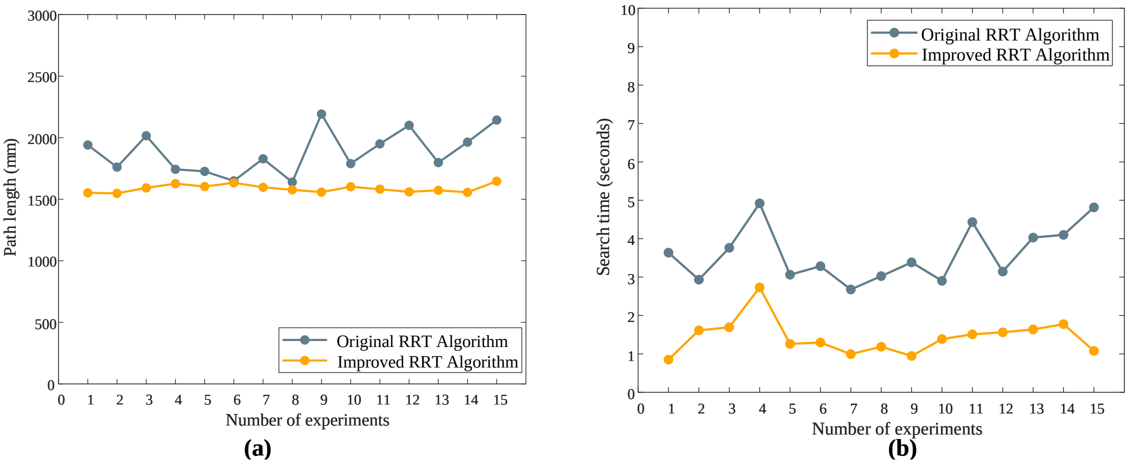
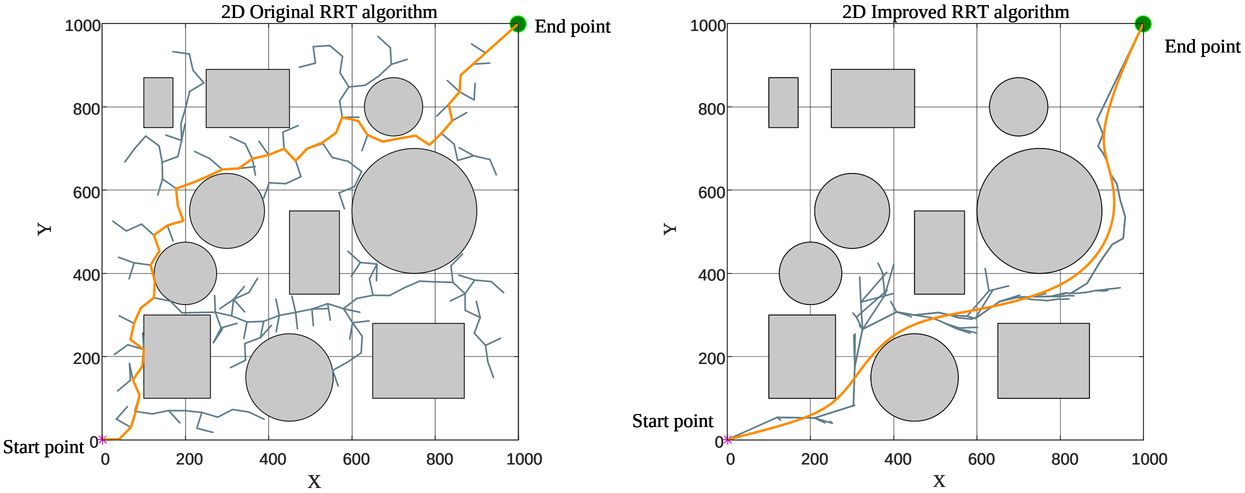


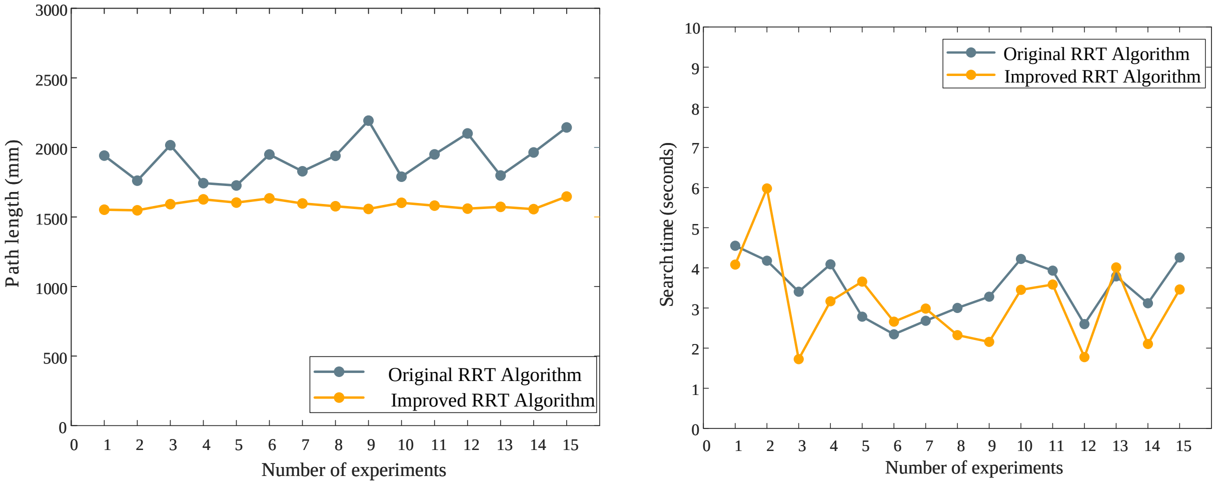
Figure 5. Comparison of path length (**a**) and search time (**b**) in a two-dimensional simple environment.

In the complex two-dimensional environment, the obstacle parameters are shown in Table 2; there are 10 obstacles in the environment, and the experiment is repeated 15 times to compare the performance of the two algorithms in complex environments. In the comparison experiment of the path length, as shown in Figure 6, the path of the improved RRT algorithm is shorter than that of the original RRT algorithm, and the average path length generated by the improved RRT algorithm is 1834.35 mm, and the average path generated by the original RRT algorithm generates an average path of 1588.22 mm, and the average path length generated by the improved RRT algorithm is about 13.4% shorter than the original RRT. In the search time comparison experiment, as shown in Figure 7, the improved RRT algorithm slightly improves the search time, and the average search time of the improved algorithm is 3.14 s, and the original RRT algorithm is 3.58 s, with an average reduction of about 9.8%.



(a) **(b)**

Figure 6. Effect of the original RRT algorithm (**a**) and the improved RRT algorithm (**b**) in a two-dimensional complex environment.



(a) **(b)**

Figure 7. Comparison of path length (**a**) and search time (**b**) in a two-dimensional complex environment.

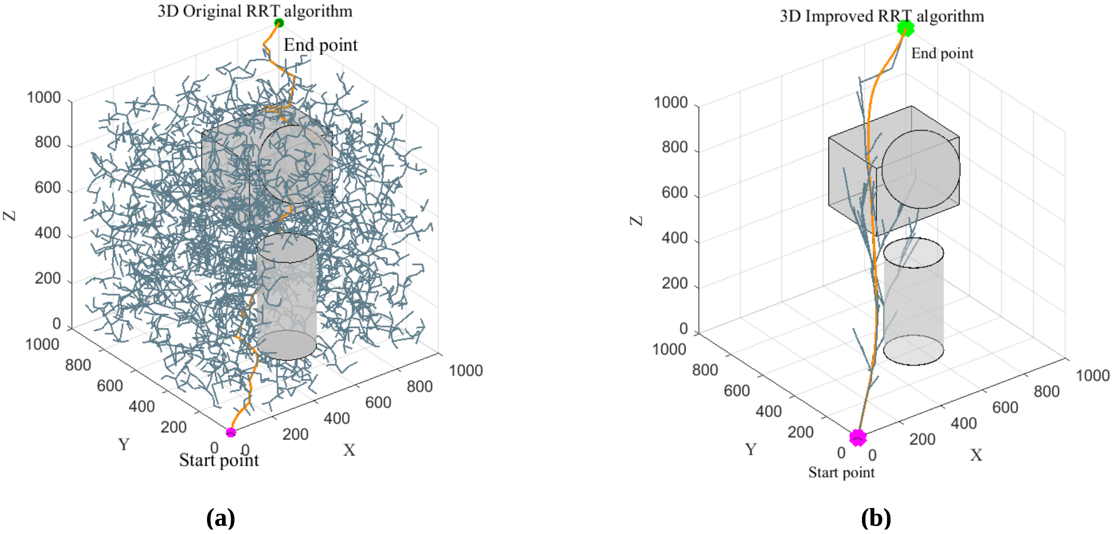
Table 2. Parameters of obstacles in two-dimensional complex environments.

Serial Number	Rectangle (Length × Width, mm)	Centroid Coordinates (mm)	Circle Radius (mm)	Circle Center Coordinates (mm)
1	160 × 200	−180,200	75	−200,400
2	200 × 140	−350,820	105	−450,150
3	220 × 180	−760,190	70	−700,800
4	120 × 200	−510,450	90	−300,550
5	70 × 120	−135,810	150	−750,550

In the simple three-dimensional environment, the obstacle parameters are shown in Table 3, and there are three obstacles in the environment, and the experiment is repeated 15 times to compare the performance of the two algorithms in the complex environment. In the comparison experiment of the path length, as shown Figure 8, the improved RRT algorithm generates a shorter path length than the original RRT algorithm in all aspects, and the improved RRT algorithm generates the average path length of 1855.79 mm, and the average path length generated by the original RRT algorithm is 2472.02 mm, which is reduced by about 24.9%. In the comparison experiment of search time, as shown in Figure 9, the improved RRT algorithm is shorter than the original RRT algorithm in terms of the average search time, and the average search time of the improved algorithm is 3.22 s, and the search time of the original RRT algorithm is 3.86 s, which is reduced by about 16.5%.

Table 3. Parameters of 3D simple environment obstacles.

Serial Number	Cuboid (Length × Width × Height, mm)	Centroid Coordinates (mm)	Cylinder (Radius × Height, mm)	Centroid Coordinates (mm)	Sphere (Radius, mm)	Sphere Center Coordinates (mm)
1	400 × 300 × 300	−750,750,570	115 × 430	−365,125,415	150	−650,450,750



(a) **(b)**

Figure 8. Effect of the original RRT algorithm (**a**) and the improved RRT algorithm (**b**) in a three-dimensional simple environment.

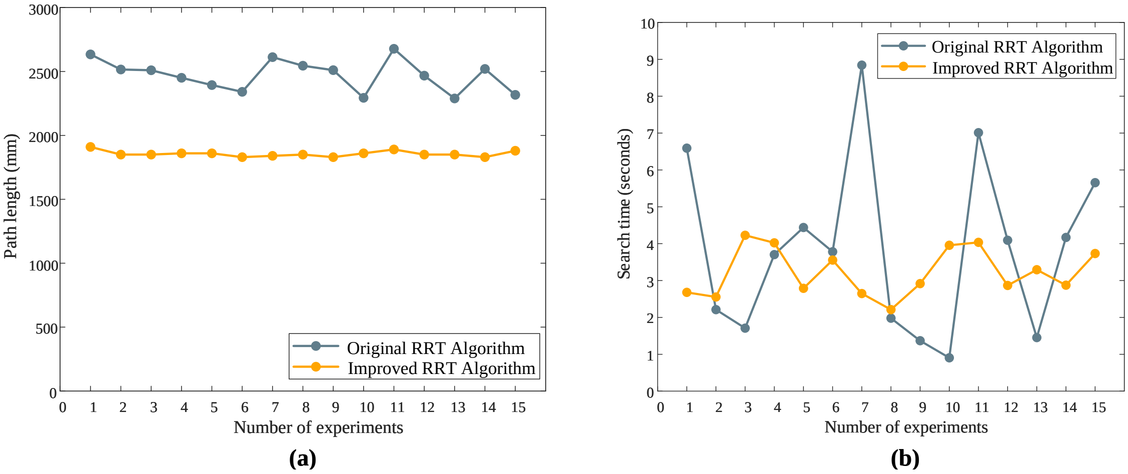


Figure 9. Comparison of path length (**a**) and search time (**b**) in a three-dimensional simple environment.

In the complex three-dimensional environment, the obstacle parameters are shown in Table 4. There are 12 obstacles in the environment, and the experiment is repeated 15 times to compare the performance of the two algorithms in complex environments. In the comparison experiment of the path length, as shown in Figure 10, the improved RRT algorithm generates a comprehensively shorter path length than the original RRT algorithm, and the average path length generated by the improved RRT algorithm is 1975.21 mm, and the average path generated by the original RRT algorithm is 2533.39 mm, which is reduced by about 22.0%. In the comparison experiment of search time, as shown in Figure 11, the improved RRT algorithm is less than the original RRT algorithm in the average search time, and the average search time of the improved algorithm is 6.09 s, while the search time of the original RRT algorithm is 7.09 s, which is reduced by about 14.1% on average.

Table 4. Parameters of 3D complex environment obstacles.

Serial Number	Cuboid (Length × Width × Height, mm)	Centroid Coordinates (mm)	Cylinder (Radius × Height, mm)	Centroid Coordinates (mm)	Sphere (Radius, mm)	Sphere Center Coordinates (mm)
1	300 × 100 × 300	−450,100,170	80 × 200	(200,200,10)	150	−750,450,650
2	100 × 300 × 400	−150,450,400	100 × 850	(850,850,50)	175	−450,780,300
3	300 × 300 × 500	−350,300,250	100 × 700	(750,200,0)	200	−300,700,800
4	150 × 500 × 150	(175,650,75)	80 × 600	(290,500,0)	100	−350,250,450

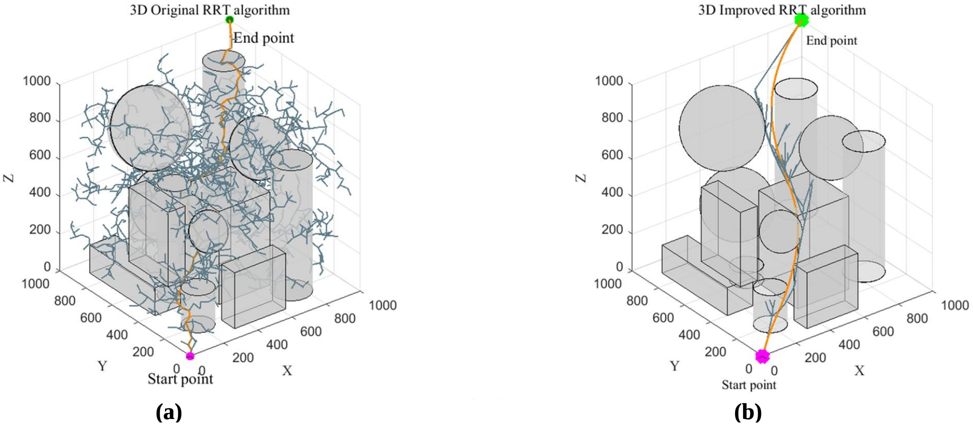


Figure 10. Effect of the original RRT algorithm (**a**) and the improved RRT algorithm (**b**) in a three-dimensional complex environment.

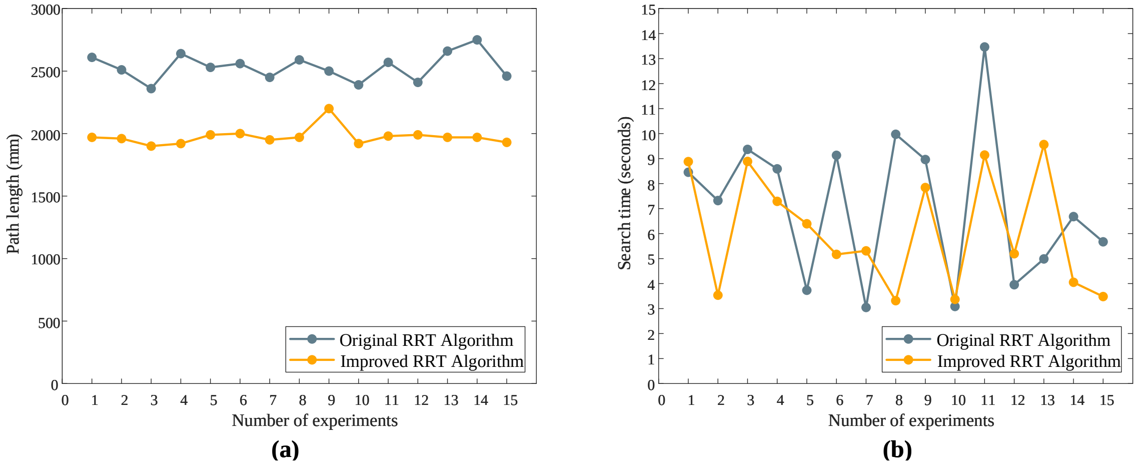


Figure 11. Comparison of path length (**a**) and search time (**b**) in a three-dimensional complex environment.

5. Analysis of Experimental Results

By introducing the gravitational and repulsive fields of the artificial potential field algorithm, the extended nodes converge to the target point, which effectively reduces the path length compared with the original RRT algorithm. In terms of path length, by comparing the original RRT algorithm with the improved RRT algorithm for multiple experiments, it is found that the improved RRT algorithm generates shorter path lengths in various environments compared with the original RRT algorithm. In simple environments, the path lengths of the two-dimensional experiments are shorter by about 16% and 3D experiments by about 24.9%, and in complex environments by about 13.4% for 2D experiments and 22% for 3D experiments.

The use of dynamic step size optimizes the tree expansion process, avoids generating a large number of redundant nodes, and improves the efficiency of the algorithm. In terms of search time, the improved RRT algorithm reduces the search time substantially in most cases, especially in simple environments, where the improvement is most significant, with an average search time reduction of about 56.5% for 2D experimental results and about 16.5% for 3D experimental results in simple environments. Reductions were also seen in complex environments, with search times reduced by approximately 9.8 percent in two-dimensional experiments and approximately 14.1 percent in three-dimensional experiments.

These results indicate that the improved RRT algorithm is more efficient and effective in path planning compared to the traditional RRT algorithm, making it particularly suitable for applications that require quick and accurate path planning. In this study, the integration of gravitational and repulsive fields from the artificial potential field algorithm into the improved RRT algorithm significantly enhanced the efficiency and optimization of path planning. The modified algorithm not only effectively reduced path lengths but also optimized the tree expansion process by dynamically adjusting step sizes, reducing the generation of redundant nodes.

6. Future Prospects

While the improved RRT algorithm shows promising results, its computational demands and sensitivity to local minima remain challenging. The dynamic adjustment of step size and weighting increases computational complexity, especially in environments with many obstacles or higher-dimensional spaces. Future work could focus on optimizing these adjustments, possibly through parallel computing or more efficient algorithms. Additionally, although dynamic weighting helps reduce local minima issues, the algorithm may still get trapped in suboptimal solutions in complex environments. Combining RRT-APF with global optimization methods or learning-based approaches could further improve its robustness and path quality [16,17], making it more suitable for real-time applications in dynamic, cluttered settings.

Author Contributions: B.Z. responsible for programming and article writing; C.L. completed the correction and proofreading of the article. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author, within reasonable limits.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

RRT	Rapidly-exploring Random Tree
APF	Artificial Potential Field
AIS	Automatic Identification System
DP	Douglas–Peucker
RDP	Ramer–Douglas–Peucker
UAV	Unmanned Aerial Vehicle
ACO	Ant Colony Optimization
BVP	Boundary Value Problem
HPO	Hyperparameter Optimization
RRT*	Rapidly-exploring Random Tree Star

References

1. Ganesan, S.; Ramalingam, B.; Mohan, R.E. A hybrid sampling-based RRT* path planning algorithm for autonomous mobile robot navigation. *Expert Syst. Appl.* **2024**, *258*, 125206. [CrossRef]
2. Fan, J.; Chen, X.; Wang, Y.; Chen, X. UAV trajectory planning in cluttered environments based on PF-RRT* algorithm with goal-biased strategy. *Eng. Appl. Artif. Intell.* **2022**, *114*, 105182. [CrossRef]
3. Liu, L.; Wang, X.; Yang, X.; Liu, H.; Li, J.; Wang, P. Path planning techniques for mobile robots: Review and prospect. *Expert Syst. Appl.* **2023**, *227*, 120254. [CrossRef]
4. Zhang, H.; Wang, Y.; Zheng, J.; Yu, J. Path Planning of Industrial Robot Based on Improved RRT Algorithm in Complex Environments. *IEEE Access* **2018**, *6*, 53296–53306. [CrossRef]
5. Mashayekhi, R.; Idris, M.Y.I.; Anisi, M.H.; Ahmedy, I. Hybrid RRT: A Semi-Dual-Tree RRT-Based Motion Planner. *IEEE Access* **2020**, *8*, 18658–18668. [CrossRef]
6. Chen, J.; Zhao, Y.; Xu, X. Improved RRT-Connect Based Path Planning Algorithm for Mobile Robots. *IEEE Access* **2021**, *9*, 145988–145999. [CrossRef]
7. Gu, Q.; Zhen, R.; Liu, J.; Li, C. An improved RRT algorithm based on prior AIS information and DP compression for ship path planning. *Ocean Eng.* **2023**, *279*, 114595. [CrossRef]
8. Kim, M.C.; Song, J.B. Informed RRT* with improved converging rate by adopting wrapping procedure. *Intell. Serv. Robot.* **2018**, *11*, 53–60. [CrossRef]
9. Chai, Q.; Wang, Y. RJ-RRT: Improved RRT for Path Planning in Narrow Passages. *Appl. Sci.* **2022**, *12*, 12033. [CrossRef]
10. Gao, J.; Geng, X.; Zhang, Y.; Wang, J. Underwater Vehicle Path Planning Based on Bidirectional Path and Cached Random Tree Star Algorithm. *Appl. Sci.* **2024**, *14*, 947. [CrossRef]
11. Wu, J.; Zhao, L.; Liu, R. Research on Path Planning of a Mining Inspection Robot in an Unstructured Environment Based on an Improved Rapidly Exploring Random Tree Algorithm. *Appl. Sci.* **2024**, *14*, 6389. [CrossRef]
12. Yang, F.; Fang, X.; Gao, F.; Zhou, X.; Li, H.; Jin, H.; Song, Y. Obstacle Avoidance Path Planning for UAV Based on Improved RRT Algorithm. *Discret. Dyn. Nat. Soc.* **2022**, *2022*, 4544499. [CrossRef]
13. Wang, J.; Chi, W.; Li, C.; Meng, M.Q.-H. Efficient Robot Motion Planning Using Bidirectional-Unidirectional RRT Extend Function. *IEEE Trans. Autom. Sci. Eng.* **2022**, *19*, 1859–1868. [CrossRef]
14. Zhao, W.; Tan, A.; Ren, C. An Innovative Path Planning Algorithm for Complex Obstacle Environments with Adaptive Obstacle Density Adjustment: AODA-PF-RRT*. *Electronics* **2024**, *13*, 4047. [CrossRef]
15. Guo, Y.; Liu, X.; Jia, Q.; Liu, X.; Zhang, W. HPO-RRT*: A sampling-based algorithm for UAV real-time path planning in a dynamic environment. *Complex Intell. Syst.* **2023**, *9*, 7133–7153. [CrossRef]

16. Cavus, M.; Allahham, A.; Adhikari, K.; Giaouris, D. A hybrid method based on logic predictive controller for flexible hybrid microgrid with plug-and-play capabilities. *Appl. Energy* **2024**, *359*, 122752. [CrossRef]
17. Cavus, M.; Ugurluoglu, Y.F.; Ayan, H.; Allahham, A.; Adhikari, K.; Giaouris, D. Switched Auto-Regressive Neural Control (S-ANC) for Energy Management of Hybrid Microgrids. *Appl. Sci.* **2023**, *13*, 11744. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.