Multi-Strategy Improved Rapid Random Expansion Tree (RRT) Algorithm for Robotic Arm Path Planning

Yuan Sun, Shoujun Zhang
Shanghai DianJi University, Shanghai, China

***Abstract*—The purpose of this paper is to propose an improved RRT algorithm that incorporates multiple improvement strategies to solve the problems of low efficiency, long and unsmooth paths in the traditional rapid random expansion tree (RRT) algorithm for path planning of robotic arms. The algorithm first uses a bidirectional tree extension strategy to generate trees from both the starting point and the target position simultaneously, improving search efficiency and reducing redundant paths. Secondly, the algorithm introduces target bias sampling in combination with local Gaussian sampling, which renders the sampling points more focused on the target area, and dynamically adjusts the distribution to improve sampling efficiency and path connection speed. Concurrently, the algorithm is equipped with an adaptive step size strategy, which dynamically adjusts the expansion step size according to the target distance, thereby achieving a balance between rapid expansion over long distances and precise search at close range. Finally, a collision-free operation is ensured by a path verification mechanism, and the path is smoothed using cubic B-splines and minimum curvature optimisation techniques, significantly improving the smoothness of the path and the feasibility of the robot arm movement. As demonstrated by simulation experiments, the improved RRT algorithm exhibits a reduction in the average path length by 18.15%, planning time by 96.29%, the number of nodes by 92.13%, and the number of iterations by 91.60%, in comparison with the conventional RRT algorithm, when operating in complex map mode. These findings substantiate the efficacy and practicality of the improved RRT algorithm in the domain of robotic arm path planning.**

***Keywords*—Robotic arm; RRT algorithm; path planning; target-biased sampling; Gaussian sampling; bidirectional tree extension; adaptive step-size**

I. Introduction

Robotic arms have become a staple of industry in fields as diverse as medicine, aerospace, and shipbuilding. The growth of social demand, coupled with the continuous development of technology, has resulted in a gradual expansion of robotic arms into applications in narrow and complex environments. In this context, path planning emerges as a pivotal technology, instrumental in navigating through confined and intricate environments. In such environments, robotic arms often encounter difficulties in operating effectively and planning a suitable trajectory, as evidenced by numerous studies. The necessity for effective path planning in such environments is therefore paramount. The efficacy of such planning is twofold: it enables the robot arm to manoeuvre with agility and circumvent potential collisions with surrounding objects, while concomitantly enhancing work efficiency and precision. This, in turn, fulfils the higher industrial and social imperatives that are now in place.

At present, the following path planning algorithms are employed with the greatest frequency: the artificial potential field method [1][2][3], the A* algorithm [4][5][6], the ant colony algorithm [7][8][9], the genetic algorithm [10][11][12], and the rapid random expansion tree (RRT) algorithm [13][14][15]. An improved RRT* algorithm based on the traditional RRT algorithm was proposed by Karaman et al. [16]. The incorporation of graph optimisation and pruning theory enables the achievement of an asymptotically optimal path, which is both complete and optimal. However, this approach significantly increases the search time. Nasir et al. [17] proposed the RRT*-Smart algorithm, which employs heuristics to enhance node expansion capabilities and optimise the path through biased sampling. Nonetheless, this algorithm is less adaptable due to its overreliance on parameter adjustment. Wei et al. [18] proposed a smooth RRT algorithm based on the maximum curvature constraint to generate continuous executable trajectories, but because it only uses target bias expansion, it is less efficient and the path fitting deviation is large. The bidirectional RRT algorithm proposed by Kuffner et al. [19] enhances planning efficiency by growing a random tree from both the starting and end points. However, it still employs the random growth strategy and sampling method of traditional RRT, which exhibits the problem of local optimality. Additionally, it exhibits poor possibility in complex environments and narrow areas, and its efficiency requires enhancement. In their seminal work, Wu et al. [20] proposed the Fast-RRT algorithm, a pioneering advancement in the field. This algorithm employs a fast sampling strategy and a random steering expansion strategy, aiming to enhance the efficiency of finding an approximate optimal path by fusing and adjusting the path. However, it is important to note a limitation in the application of this algorithm. Specifically, its use is primarily constrained to two-dimensional environments, and its efficacy in multidimensional spaces is not well-documented.

The rapid random expansion tree (RRT) algorithm has become a significant method in the field of path planning due to its high search efficiency, wide applicability, and the fact that it does not require global modelling of the environment. However, the traditional RRT algorithm is not without its shortcomings, namely the random sampling process, which is inefficient and results in a protracted search time. Additionally, the presence of numerous redundant nodes can compromise the quality of the path, and the ability to swiftly identify a feasible path near the target point or in areas with obstacles is also hindered. These limitations constrain the applicability of the RRT algorithm in complex environments. To address these issues, this paper proposes an improved algorithm for the traditional RRT algorithm. The proposed strategy involves the implementation of a double-tree expansion approach, which involves the simultaneous expansion of trees from both the starting point and the target point. This strategy has been shown to enhance the efficiency of path search, leading to faster connection path discovery and reduced redundant expansion. Additionally, the integration of a target bias sampling strategy enhances the probability of sampling points being in close proximity to the target area, thereby accelerating the convergence of the algorithm. Gaussian distribution sampling is introduced in the target vicinity, and the target area is searched in detail by dynamically adjusting the sampling range. The combination of Gaussian sampling and target bias improves the efficiency of path generation. An adaptive step size is introduced, which dynamically adjusts the step size according to the distance between the current node and the target point, improving the efficiency and accuracy of path planning by achieving rapid expansion over long distances and precise search near the target. A collision detection and avoidance mechanism is integrated into the path expansion process, ensuring the generated path is free of collisions, enhancing its safety and practical applicability. Finally, cubic B-splines and minimum curvature optimisation are incorporated into the generated path to enhance its smoothness, reduce sharp turns, and improve its feasibility.

The improved RRT algorithm has been shown to exhibit notable enhancements in terms of search efficiency, path quality and adaptability. These improvements render it particularly well-suited for applications in complex restricted environments, such as robotic arm path planning.

The subsequent arrangement of this paper is as follows. Section II introduces the RRT algorithm. Section III introduces and derives the improved parts of the improved RRT algorithm. Section IV simulates the RRT algorithm, the RRT* algorithm, the RRT-Connect algorithm, and the improved RRT algorithm, and compares the data obtained by the four algorithms to demonstrate the superiority and feasibility of the proposed improved RRT algorithm in path planning. Section V is the conclusion of this paper.

II. Principle of the RRT Algorithm

1) Initialise the extended tree T with a step size δ, a starting point $x_{\text{init}}$and a goal point x_{goal}. Add the starting point x_{init} to the extended tree T as a root node.

2) Create a random point x_{rand} in the robot's workspace.

3) Find the node x_{near} in the extended tree T that is closest to x_{rand}.

4) Extend fromx_{near} towards x_{rand} with a step size δ to obtain a new node x_{new}.

5) Perform an obstacle collision detection on the line segment between x_{new} and x_{near}. If the detection fails (the path intersects with an obstacle), discard x_{new} and return to step 2) to start a new round of sampling. If the detection is successful, proceed to step 6).

6) Add x_{new} to the extended tree T and set x_{near} as the parent node of x_{new}.

7) Determine whether x_{new} has reached the target point x_{goal} (i.e. the distance between x_{new} and x_{goal} is less than a tolerance threshold).

If the target point is not reached, go to step 2) and continue sampling. If the target point is reached, the path planning is successful, stop the algorithm and output the path according to the extended tree T.

As shown in Fig. 1.

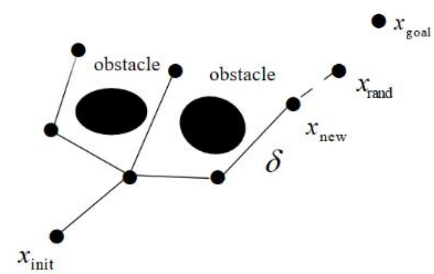


Fig. 1. Schematic diagram of RRT algorithm.

III. Improved RRT Algorithm

A. Targeted Bias Sampling

In order to enhance the efficacy of the algorithm, a target bias sampling strategy has been implemented. This strategy establishes a probability value, p_{rand}, which is distributed uniformly between 0 and 1. When generating a random sample point, x_{rand}, the method of generating the sample point is determined according to the result of comparing a random value, p, with p_{rand}. Specifically, when $p < p_{\text{rand}}$, the target point x_{goal} is directly selected as the sampling point; when $p > p_{\text{rand}}$, a random sampling point x_{rand} is generated within the search space. The specific mathematical expression of this strategy is as follows:

$$x_{\text{rand}} = \begin{cases} x_{\text{goal}}, & p < p_{\text{rand}} \\ \text{sample}, & p \geq p_{\text{rand}} \end{cases} \tag{1}$$

The term 'Sample' is used to denote a state point that has been randomly generated from the search space, whilst x_{goal} indicates a predefined goal point. The goal bias sampling strategy has been introduced with a view to increasing the sampling probability of the aforementioned goal point during the growth of the random tree, thereby accelerating the expansion of the tree towards the goal region. This strategy has been shown to significantly improve search efficiency whilst also effectively reducing the generation of invalid nodes and the number of algorithm iterations. Furthermore, the value adjusted by p_{rand} can dynamically balance the proportion of goal point bias sampling and random sampling to adapt to search environments of different complexities as shown in Fig. 2.

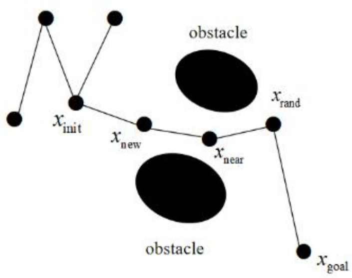


Fig. 2. Schematic diagram of the target bias strategy.

B. Local Gaussian Sampling

The principle underlying local Gaussian sampling involves the generation of sampling points through the introduction of a Gaussian distribution in proximity to the target point or other significant locations. The implementation of this method entails the random generation of points according to a Gaussian distribution, with the distribution density of the generated points being controlled by the standard deviation, σ.In instances where the target point is distant from the current point, an increase in σresults in a more dispersed distribution of sampling points, encompassing a broader area. Conversely, when the target point is proximate, a reduction in σ results in a more concentrated distribution of sampling points, thereby enhancing the local search capability. The formula for Gaussian sampling is as follows:

$$\mathrm{f}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{2}$$

μ: mean value, σ: standard deviation, x: random variable

The amalgamation of Gaussian sampling and target-biased sampling results in the generation of a high-density distribution of sampling points in close proximity to the target point, whilst preserving the randomness intrinsic to global exploration. This amalgamation offers substantial advantages in enhancing the efficiency of path planning, reducing the number of iterations and path length, and is particularly well-suited to path planning problems in complex environments.

C. Bidirectional Tree Extension

First, two random trees T_1 and T_2 are constructed, with x_{start} as the root node of T_1 for expansion and x_{goal} as the root node of T_2 for expansion. Then, random sampling generates two sampling points x_{rand1} and x_{rand2}, which are used to expand the two trees respectively. For T_1, the closest node x_{near1} to the sampling point x_{rand1} is found among its existing nodes, and a new node x_{near1} is created by expanding with a fixed step δ on the line x_{near1} pointing to x_{rand1}.

Similarly, for T_2, the closest node x_{near2} to the sampling point x_{rand2} is found and expanded at fixed steps δ on the line x_{near2} pointing to x_{rand2}, generating a new node x_{near2}.

The next step is to check if there is a collision on the connecting line between the generated new node x_{near1} and x_{near2}.

If the connecting line passes through an obstacle, the new node is discarded and re-sampled, and the last valid retained node is returned; if the connecting line is free of obstacles, x_{near1} and x_{near2} are connected to complete the connection of the two trees.Expand T_1 and T_2 alternately according to the above method until the distance between the adjacent new nodes of the two trees is less than the threshold of the step size δ. At this point, T_1 and T_2 are successfully connected and the path is generated. As shown in Fig. 3.

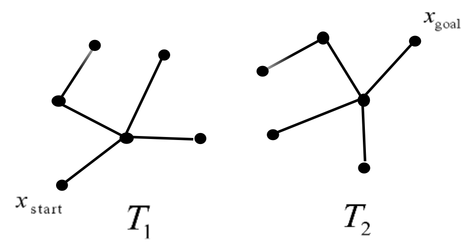


Fig. 3. Schematic representation of double-tree expansion.

D. Adaptive Step Size

It is evident that traditional RRT algorithms utilise a fixed step size for expansion, a practice that may give rise to several issues. Inefficiency is a notable concern, as the fixed step size can impede the efficacy of expansion, particularly in open areas. This limitation can result in an excessive number of unnecessary searches. Moreover, the fixed step size can compromise accuracy, particularly in the vicinity of the target or in narrow areas. In such instances, the step size may either miss the target entirely or encounter difficulties in navigating a complex environment, ultimately leading to failure. The adaptive step size strategy has been developed to address these issues by dynamically adjusting the step size according to the distance between the expansion node and the target point. The concept of an adaptive step size is outlined below. The fundamental principle of this approach entails the dynamic adjustment of the expansion step size, thereby facilitating the manifestation of distinct search behaviours in diverse environmental contexts. In environments that are distant from the target, a larger step size is employed to expedite the search coverage. Conversely, in close proximity to the target, a reduction in the step size is implemented to enhance the search accuracy. In complex environments characterised by dense obstacles, a further reduction in the step size is initiated to augment the success rate of traversing the path.

The employment of adaptive step size facilitates the expansion of the tree, thereby enabling efficient exploration of the global environment and the execution of precise searches in complex regions or in proximity to the target. This is achieved through the dynamic adjustment of the expansion step size. As shown in Fig. 4.

$$L_{\text{adaptive}} = \begin{cases} L_{\max}, & \alpha \cdot d_{\text{goal}} + \beta \geq L_{\max} \\ \alpha \cdot d_{\text{goal}} + \beta, & \alpha \cdot d_{\text{goal}} + \beta < L_{\max} \end{cases} \tag{3}$$

L_{adaptive}: adaptive step size, L_{max}: maximum allowable step size, $d_{\text{goal}} = \left\| p_{\text{current}} - p_{\text{goal}} \right\|$: Euclidean distance from current node to goal point, α: coefficient, β: offset, p_{current}: indicates the position of the current tree node, p_{goal}: indicates the position of the goal point.

Adaptive step size constitutes a dynamic optimisation strategy, which is employed throughout the expansion process of the improved RRT algorithm. This strategy enhances global search efficiency and mitigates ineffective expansion by integrating it with the concepts of target bias sampling, local Gaussian sampling, and double-tree expansion. The strategy enhances local accuracy and optimises path availability and smoothness, thereby accelerating the dual-tree connection and enhancing the success rate and speed of planning. Finally, the enhanced adaptability to complex environments is suitable for real robot path planning needs.

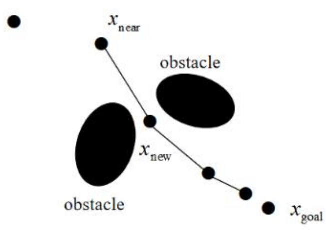


Fig. 4. Adaptive step size schematic.

E. Path Smoothing

Robotic arms have been observed to be susceptible to sudden acceleration at inflection points during the process of path planning. This instability necessitates the implementation of a smoothing technique to ensure the stability of the path. Cubic B-spline represents an interpolation method employed for the smoothing of paths, whereby data points are fitted by means of segmented polynomial functions, thereby enhancing the smoothness of the path as shown in Fig. 5.

$$Q(m) = \sum_{k=0}^{m} R_k\, G_{k,m}(s), s \in [0,1] \quad (4)$$

In this study, the equation of the control point of the kth segment is denoted byR_k, and the basis function of the nth B-spline is denoted by $G_{k,m}$.

$$G_{k,m}(s) = \frac{1}{m!}\sum_{v=0}^{m-k}(-1)^v\, T_{m+1}^{v}(s+m-k-v) \quad (5)$$

$$T_{m+1}^{v} = \frac{(m+1)!}{v!(m+1-v)!} \quad (6)$$

Minimum curvature smoothing is a process of path smoothing in which the path is rendered more natural by minimising the curvature of the path. The objective of minimum curvature smoothing is to minimise the integral of the square of the curvature, i.e:

$$min \int k^2\,(s)ds \quad (7)$$

Among them:

$$K = \frac{x'y''-y'x''}{(x'^2+y'^2)^{3/2}} \quad (8)$$

The first-order derivatives of the path are denoted by x' and y', whilst the second-order derivatives are denoted by $x''$and y''.

The combination of cubic B-splines and minimum curvature optimisation generates smooth, continuous and natural paths with high implementability and efficiency.

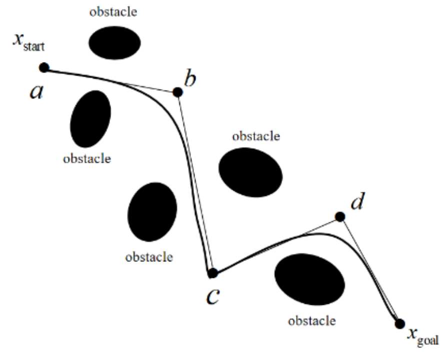


Fig. 5. Schematic diagram of three times B-spline curve fitting.

F. Improved RRT Algorithm Process

Step 1: Initialize the map, initialize the obstacle positions, and initialize the parameters. Initialize the two trees: $T_1$with the starting point as the root node and T_2 with the end point as the root node.

Step 2: The target offset strategy is used to generate sampling points. The target point is directly selected with a set probability, quickly guiding the path to converge towards the target. In all other cases, sampling points are generated randomly to enhance exploration. Local Gaussian sampling is applied to the random sampling points, generating Gaussian distribution sampling points near the target point, with the offset dynamically adjusted by the target distance.

Step 3: Expand T_1, find the node closest to the sampling point in T_1, and use a fixed step size to expand at a long distance, quickly approaching the sampling point, and dynamically reducing the step size at a short distance to improve the expansion accuracy and avoid over-expansion. Generate a new node according to the adjusted step size, and verify whether the path collides with obstacles. If the path does not collide, add the new node to T_1; if the path is invalid, skip the current sampling point and return to regenerate the sampling point.

Step 4: Expand T_2. T_2 expands towards the new node added to T_1 and executes the same logic as T_1.

Step 5: If the new node T_2 is successfully expanded and the distance between the nodes of the two trees is less than the step size, the two trees are considered to be connected. If the two trees are not successfully connected, the resampling stage is entered.

Step 6: Generate the complete path by retracing it from the two trees, smooth the path using a cubic B-spline three times, and further reduce sharp turns and improve path smoothness through curvature optimization.

Step 7: End

The flow of the RRT improvement algorithm is shown in Fig. 6.

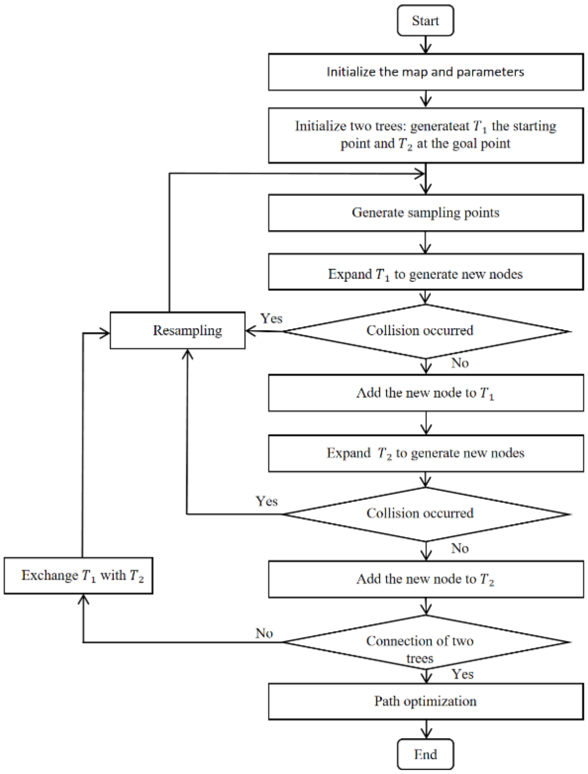
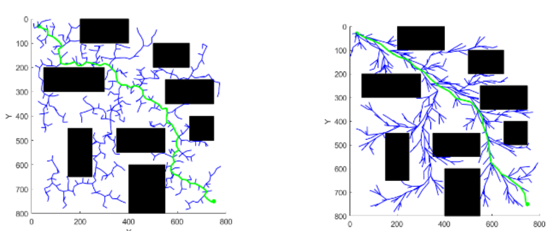


Fig. 6. Flowchart of the improved RRT algorithm.

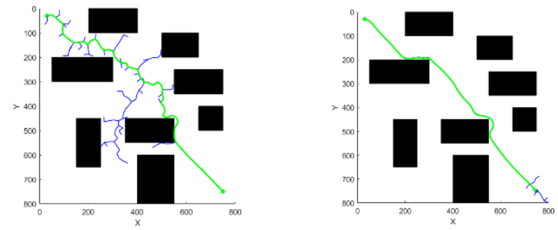
IV. Experimental Design and Analysis

A. Simulation Experiments in a Two-Dimensional Environment

The simulation experiment is based on the MATLAB R2021b platform. The hardware configuration of the simulation platform consists of an AMD Ryzen7 4800H processor, running the Windows 10 operating system, with a total running memory of 32GB. The experiment is designed to conduct three maps, each measuring 800×800, with the origin of the coordinates positioned in the upper left corner. The simulation experiments were executed on the MATLAB platform. The initial starting point of the four algorithms is (30, 30), the target point is (750, 750), the step delta is 20, the maximum number of searches is 3,000, and the target bias probability of the Improved RRT algorithm is 0.3. Each map was executed 100 times under each algorithm.



(a) RRT algorithm (b) RRT* algorithm



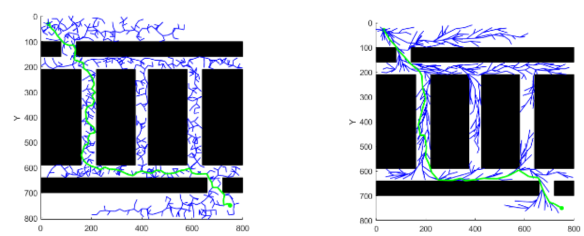
(c) RRT-Connect algorithm (d) Improved RRT algorithm

Fig. 7. Four Algorithmic path planning in normal map mode.

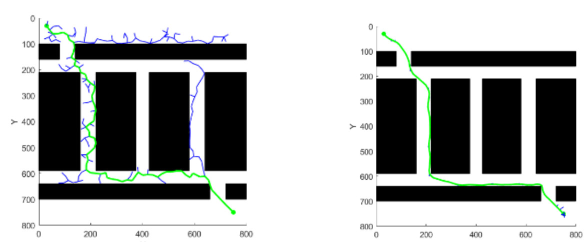
TABLE I COMPARISON OF THE RESULTS OF THE FOUR ALGORITHMS IN NORMAL MAP MODE

Algorithm type	Average path length /mm	Average running time /s	Average nodes	Average iterations
RRT	1348.98	13.23	963.13	1321.44
RRT*	1099.91	13.71	965.88	1311.19
RRT-Connect	1302.15	1.03	134.11	143.92
Improved RRT	1114.84	0.46	64.72	79.88

The results of the normal map mode experiment are shown in Fig. 7. The analysis of the experimental data in Table I shows that the improved RRT algorithm exhibits significant optimization effects compared to the conventional RRT algorithm when there are fewer obstacles. Specifically, the improved algorithm has a 17.36% reduction in the average path length, a 96.52% reduction in the average running time, a 93.28% reduction in the average number of nodes, and a 93.96% reduction in the average number of iterations. These results show that the improved RRT algorithm is significantly better than the traditional RRT algorithm in terms of both path planning efficiency and path quality.



(a) RRT algorithm (b) RRT* algorithm



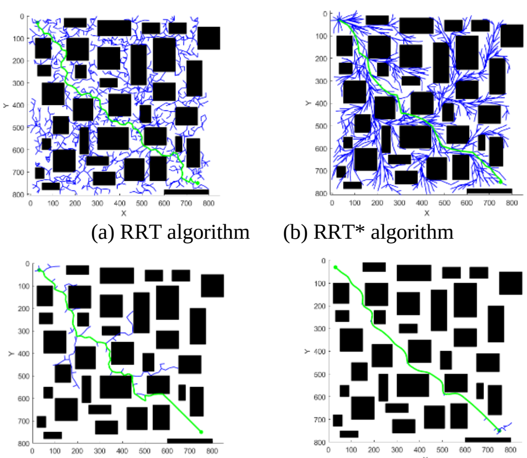
(c) RRT-Connect algorithm (d) Improved RRT algorithm

Fig. 8. Four Algorithmic path planning in narrow map mode.

TABLE II COMPARISON OF THE RESULTS OF THE FOUR ALGORITHMS FOR PATH PLANNING IN THE NARROW MAP MODE

Algorithm type	Average path length /mm	Average running time /s	Average nodes	Average iterations
RRT	1494.86	5.27	563.30	1397.24
RRT*	1245.49	5.13	557.89	1414.16
RRT-Connect	1460.40	1.23	168.66	310.06
Improved RRT	1249.52	0.27	36.42	107.61

The experimental results in the narrow map mode are shown in Fig. 8, and the corresponding data are shown in Table II. The experimental data show that the improved RRT algorithm has a significant optimization effect in the case of extremely narrow passages compared to the traditional RRT algorithm. In the narrow map mode, the improved RRT algorithm has an average path length that is 16.41% shorter than the traditional RRT algorithm, an average running time that is 94.88% shorter, an average number of nodes that is 93.53% lower, and an average number of iterations that is 92.30% lower. Experimental data show that the improved RRT algorithm requires a shorter path, less time, and fewer nodes and iterations to search in a confined environment compared to the other three algorithms.



(a) RRT algorithm (b) RRT* algorithm

(c) RRT-Connect algorithm (d) Improved RRT algorithm

Fig. 9. Four Algorithmic path planning in complex map mode.

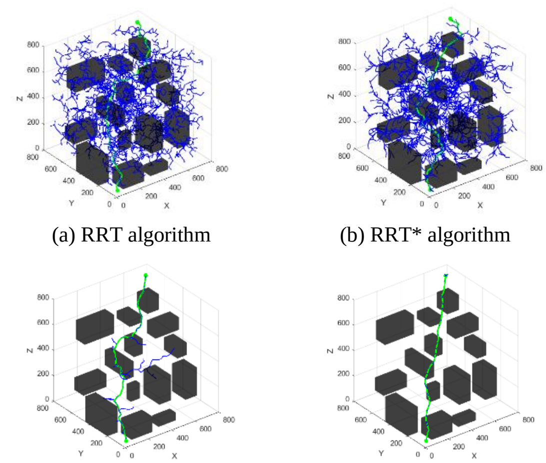
TABLE III COMPARISON OF THE RESULTS OF THE FOUR ALGORITHMS FOR PATH PLANNING IN COMPLEX MAP MODE

Algorithm type	Average path length /mm	Average running time /s	Average nodes	Average iterations
RRT	1374.28	9.96	792.24	1398.35
RRT*	1128.03	8.78	757.77	1334.64
RRT-Connect	1317.84	1.38	163.75	249.60
Improved RRT	1124.88	0.37	62.33	117.53

The experimental results in the complex map mode are shown in Fig. 9, and the corresponding data are shown in Table III. The experimental data show that in the case of dense obstacles and complex road conditions, the improved RRT algorithm shows a significant performance improvement compared to the traditional RRT algorithm. Specifically, the average path length is reduced by 18.15%, the average running time is reduced by 96.29%, the average number of nodes is reduced by 92.13%, and the average number of iterations is reduced by 91.60%. The results show that the improved RRT algorithm can significantly improve the search efficiency, optimize the path quality, and reduce the computational resource consumption when dealing with path planning tasks in complex scenarios.

B. Simulation Experiment in a Three-Dimensional Environment

In order to improve the RRT algorithm, a 3D map was designed for the experiment, and the size of the map was 800×800×800. Simulation experiments were performed on the MATLAB platform. The starting point of the four algorithms was (30, 30, 30), the target are (750, 750, 750), the step size is 30, and the maximum number of searches is 5000. The target bias probability of the improved RRT algorithm is 0.3. Each map is run 20 times for each algorithm.



(a) RRT algorithm (b) RRT* algorithm

(c) RRT-Connect algorithm (d) Improved RRT algorithm

Fig. 10. Four Algorithmic path planning in 3D map mode.

TABLE IV COMPARISON OF THE RESULTS OF THE FOUR ALGORITHMS FOR PATH PLANNING IN 3D MAP MODE

Algorithm type	Average path length /mm	Average running time / s	Average nodes	Average iterations
RRT	1798.07	107.24	2855.00	3014.14
RRT*	1684.26	94.51	2451.43	2787.29
RRT-Connect	1650.97	1.53	124.60	109.30
Improved RRT	1335.17	0.78	58.90	89.80

The relevant data is summarised in Table V.

TABLE V SUMMARY TABLE OF DIFFERENT ALGORITHM PARAMETERS IN FOUR MAP MODES

map mode	Algorithm type	Average path length /mm	Average running time /s	Average nodes	Average iterations
normal map	RRT	1348.98	13.23	963.13	1321.44
	RRT*	1099.91	13.71	965.88	1311.19
	RRT-Connect	1302.15	1.03	134.11	143.92
	Improved RRT	1114.84	0.46	64.72	79.88
narrow map	RRT	1494.86	5.27	563.30	1397.24
	RRT*	1245.49	5.13	557.89	1414.16
	RRT-Connect	1460.40	1.23	168.66	310.06
	Improved RRT	1249.52	0.27	36.42	107.61
complex map	RRT	1374.28	9.96	792.24	1398.35
	RRT*	1128.03	8.78	757.77	1334.64
	RRT-Connect	1317.84	1.38	163.75	249.60
	Improved RRT	1124.88	0.37	62.33	117.53
3D map	RRT	1798.07	107.24	2855.00	3014.14
	RRT*	1684.26	94.51	2451.43	2787.29
	RRT-Connect	1650.97	1.53	124.60	109.30
	Improved RRT	1335.17	0.78	58.90	89.80

The experimental results in 3D map mode are shown in Fig. 10, and the corresponding data are shown in Table IV. The experimental results show that due to its limitations, the traditional RRT algorithm tends to generate a large number of branches and redundant nodes in a 3D simulation environment, resulting in a long path planning time and poor path quality. The improved RRT algorithm showed significant optimization effects in these aspects. The improved RRT algorithm reduced the average path length by 25.74%, the average running time by 99.27%, the average number of nodes by 97.94%, and the average number of iterations by 97.02%. Overall, the improved RRT algorithm performed particularly well in 3D environments. It outperformed the traditional RRT algorithm in terms of path quality, planning speed, and resource utilization.

V. CONCLUSION

This paper proposes an improved algorithm based on the RRT algorithm, which incorporates a dual-tree expansion strategy, target bias sampling, local Gaussian sampling, adaptive step length, cubic B-spline smoothing, and minimum curvature optimization. This algorithm effectively solves the problems of path smoothing, node redundancy, and search failure in traditional RRT algorithms, significantly improving search efficiency and path planning reliability, while enhancing the adaptability and practicality of the algorithm in complex environments. Simulation results show that the improved RRT algorithm is significantly better than the traditional RRT algorithm in terms of key performance indicators such as path length, planning time, node count, and iteration count. The average path length is reduced by 18.15%, the planning time is reduced by 96.29%, the number of nodes is reduced by 92.13%, and the number of iterations is reduced by 91.60%. The improved algorithm can significantly reduce the planning time and sampling redundancy, while generating shorter and higher quality paths. The experimental results fully verify the efficiency and feasibility of the algorithm in complex environments. In future work, other aspects will need to be improved, such as extending the path planning of the robotic arm to a dynamic multi-dimensional obstacle environment.

REFERENCES

[1] Zhang N, Cui C, Wu G. Path planning of a 5-dof robotic arm based on BiRRT-APF algorithm considering obstacle avoidance. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science. 2022;236(16):9282-9292.

[2] M. Zhuang, G. Li and K. Ding, "Obstacle Avoidance Path Planning for Apple Picking Robotic Arm Incorporating Artificial Potential Field and A* Algorithm," in IEEE Access, vol. 11, pp. 100070-100082, 2023.

[3] C. Bai, J. Zhang, J. Guo and C. P. Yue, "Adaptive Hybrid Optimization Learning-Based Accurate Motion Planning of Multi-Joint Arm," in IEEE Transactions on Neural Networks and Learning Systems[J], vol. 34, no. 9, pp. 5440-5451, Sept. 2023.

[4] X. Gan, Z. Huo and W. Li, "DP-A*: For Path Planing of UGV and Contactless Delivery," in IEEE Transactions on Intelligent Transportation Systems[J], vol. 25, no. 1, pp. 907-919, Jan. 2024.

[5] K. Lin, Y. Li, S. Chen, D. Li and X. Wu, "Motion Planner With Fixed-Horizon Constrained Reinforcement Learning for Complex Autonomous Driving Scenarios," in IEEE Transactions on Intelligent Vehicles[J], vol. 9, no. 1, pp. 1577-1588, Jan. 2024.

[6] Tang X, Zhou H, Xu T. Obstacle avoidance path planning of 6-DOF robotic arm based on improved A* algorithm and artificial potential field method. Robotica. 2024;42(2):457-481.

[7] Chao Liu, Lei Wu, Wensheng Xiao, Guangxin Li, Dengpan Xu, Jingjing Guo, Wentao Li,An improved heuristic mechanism ant colony optimization algorithm for solving path planning,Knowledge-Based Systems,Volume 271,2023,110540,ISSN 0950-7051.

[8] Junguo Cui, Lei Wu, Xiaodong Huang, Dengpan Xu, Chao Liu, Wensheng Xiao,Multi-strategy adaptable ant colony optimization algorithm and its application in robot path planning,Knowledge-Based Systems,Volume 288,2024,111459,ISSN 0950-7051.

[9] J. Fu et al., "Multirobot Cooperative Path Optimization Approach for Multiobjective Coverage in a Congestion Risk Environment," in IEEE Transactions on Systems, Man, and Cybernetics: Systems[J], vol. 54, no. 3, pp. 1816-1827, March 2024.

[10] Yuan, J.; Liu, Z.; Lian, Y.; Chen, L.; An, Q.; Wang, L.; Ma, B. Global Optimization of UAV Area Coverage Path Planning Based on Good Point Set and Genetic Algorithm. Aerospace 2022, 9, 86.

[11] Ritam Sarkar, Debaditya Barman, Nirmalya Chowdhury,Domain knowledge based genetic algorithms for mobile robot path planning having single and multiple targets,Journal of King Saud University - Computer and Information Sciences,Volume 34, Issue 7,2022,Pages 4269-4283,ISSN 1319-1578.

[12] Suresh, K.S., Ravichandran, K.S., and Venugopal, S. ‘Multi-objective Genetic Algorithm for Mobile Robot Path Planning in Industrial Automation’. 1 Jan. 2023 : 6829 – 6842.

[13] Dong, Z.; Zhong, B.; He, J.; Gao, Z. Dual-Arm Obstacle Avoidance Motion Planning Based on Improved RRT Algorithm. Machines 2024, 12, 472.

[14] Yan Wang, Wensong Jiang, Zai Luo, Li Yang, Yanqing Wang,Path planning of a 6-DOF measuring robot with a direction guidance RRT method,Expert Systems with Applications,Volume 238, Part D,2024,122057,ISSN 0957-4174.

[15] Meilin Kang, Qinhu Chen, Zeming Fan, Chuan Yu, Yixin Wang, Xiaojun Yu,A RRT based path planning scheme for multi-DOF robots in unstructured environments,Computers and Electronics in Agriculture,Volume 218,2024,108707,ISSN 0168-1699.

[16] Karaman, Sertac and Emilio Frazzoli. "Sampling-based algorithms for optimal motion planning." The International Journal of Robotics Research 30 (2011): 846 - 894.

[17] F. Islam, J. Nasir, U. Malik, Y. Ayaz and O. Hasan, "RRT∗-Smart: Rapid convergence implementation of RRT∗ towards optimal solution," 2012 IEEE International Conference on Mechatronics and Automation, Chengdu, China, 2012, pp. 1651-1656.

[18] Wei, K.; Ren, B. A Method on Dynamic Path Planning for Robotic Manipulator Autonomous Obstacle Avoidance Based on an Improved RRT Algorithm. Sensors 2018, 18, 571.

[19] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), San Francisco, CA, USA, 2000, pp. 995-1001.

[20] WU Z P，MENG Z J，ZHAO W L，et al.Fast-RRT : a RRT-based op-timal path finding method［J］. Applied Sciences，2021，11（24）：11777.