Journal of Artificial Intelligence Practice (2025)
Clausius Scientific Press, Canada
DOI: 10.23977/jaip.2025.080316
ISSN 2371-8412 Vol. 8 Num. 3

Research on Robotic Arm Path Planning Algorithm Based on Bidirectional Target-Biased APF-Informed-RRT Algorithm*

Zhai Linlin[1,a,*], Fu Jiajie[1]

[1]*Shanghai Technical Institute of Electronics & Information, Wahong Highway, Shanghai, China*
[a]*zhaill1992@163.com*
**Corresponding author*

Keywords: Robotic Arm; Path planning algorithm; Bidirectional target bias; Adaptive step size; Artificial Potential Field (APF); B-spline smoothing

Abstract: Aiming at the problems of high search randomness, poor target bias, and low path quality in traditional robotic arm path planning algorithms, a Bidirectional Target-Biased APF-Informed-RRT* (BTB-APF-Informed-RRT*) algorithm is proposed. Firstly, a probabilistically adaptive target bias strategy is introduced based on the bidirectional Informed-RRT* to reduce the randomness of the original bidirectional RRT* search and improve sampling efficiency. Secondly, the Artificial Potential Field (APF) method is integrated into the path expansion process of the bidirectional search trees to reduce the number of algorithm iterations. Simultaneously, during the path growth phase, an adaptive step size growth strategy is adopted, which dynamically adjusts the step size according to the expansion trend of the search tree to avoid local optima and shorten the path search time until an initial path is generated. Finally, redundant nodes in the generated path are removed using the triangle inequality principle, and the path is smoothed using cubic B-spline curves to obtain the optimal planned path. Simulation comparisons of the improved algorithm were conducted in two-dimensional environments under three different scenarios. The results show that the improved algorithm effectively enhances relevant performance. Applying the improved algorithm to a physical platform further demonstrates its effectiveness and feasibility.

1. Introduction

With the rapid development of intelligent manufacturing and automation technology, robotic arms— as core equipment of execution units— have been widely used in complex operation scenarios such as assembly, welding, spraying, handling, and precision measurement. Path planning is a fundamental and key technology to ensure that robotic arms complete tasks autonomously, safely, and efficiently. Its goal is to find a collision-free feasible motion trajectory from the start state to the target state in a high-dimensional configuration space (C-Space) filled with obstacles, while also meeting certain optimization indicators (e.g., shortest path, time optimality, or energy efficiency).

Traditional path planning algorithms, such as graph search algorithms like A* and Dijkstra algorithms, are effective in low-dimensional discrete spaces. However, their computational complexity increases exponentially with dimensionality, making them difficult to apply to the high-dimensional continuous C-Space of robotic arms. The Rapidly-exploring Random Tree (RRT) algorithm, based on random sampling, has become a research hotspot in robotic arm path planning due to its probabilistic completeness, the absence of a need for explicit environmental model construction, and good adaptability to high-dimensional spaces. Many scholars have proposed improvements to it. Wu Kai [1] et al. proposed an improved RRT algorithm: to enhance sampling guidance and search efficiency, they introduced a target-biased sampling strategy with prior conditions and an adaptive step size strategy. Liu Yang [2] et al., addressing issues such as high sampling randomness, low planning efficiency, and tortuous paths prone to oscillation in traditional RRT* algorithms in complex 3D scenes, proposed an improved RRT* algorithm that integrates a Sobol sequence sampling strategy, the artificial potential field method, and a path optimization strategy. By proposing a low-discrepancy Sobol sequence sampling strategy, they improved the shortcomings of repeated sampling in random sampling strategies; during the growth of the expansion tree, they proposed a method integrating the artificial potential field to guide the generation of new nodes, thereby enhancing path search capability and accelerating convergence speed. Nevertheless, these algorithms still suffer from inherent drawbacks such as blind random sampling, low search efficiency, tortuous paths, and non-optimality.

To overcome the above problems, scholars have proposed many improvement schemes. The RRT* algorithm proposed by Karaman et al. [3] achieves asymptotic path optimality by introducing "rewiring" and "reparenting" operations, but its convergence speed is slow. The RRT-Connect [4] (or bidirectional RRT, Bi-RRT) algorithm proposed by Kuffner et al. grows two random trees simultaneously from the start and goal points, and significantly improves search efficiency through an alternating expansion strategy. The Informed-RRT* algorithm proposed by Gammell et al. [5] limits the sampling area to an elliptical hypersphere with the start and end points as foci after finding an initial path, thus focusing the optimization search and accelerating convergence. Furthermore, integrating heuristic information into the RRT framework is another important research direction. The Artificial Potential Field (APF) method [6] guides the search by simulating the attraction of the target point and the repulsion of obstacles, and is widely combined with RRT to enhance target orientation and local obstacle avoidance capability. However, traditional APF is prone to falling into local minima.

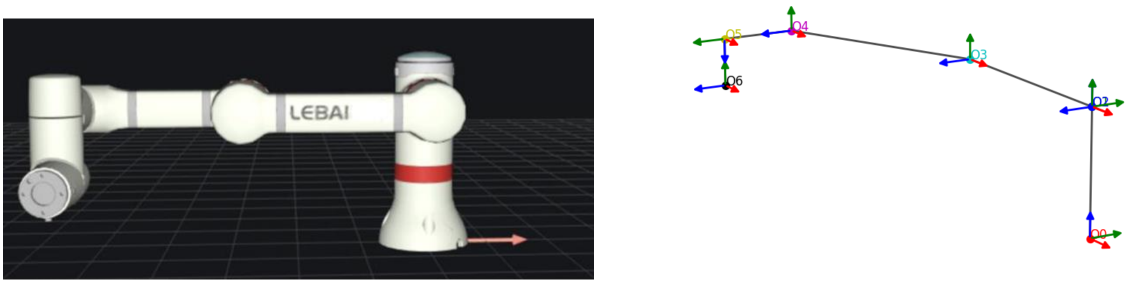
Although existing research has made significant progress, current bidirectional RRT* algorithms still face problems such as excessive sampling randomness, unsatisfactory convergence speed in complex environments, and generated initial paths with numerous redundant nodes and poor smoothness. Although the target bias strategy can guide the search to a certain extent, a fixed bias probability is difficult to adapt to complex and dynamic obstacle environments.

Aiming at the above problems, this paper deeply integrates multiple strategies based on existing research and proposes a Bidirectional Target-Biased APF-Informed-RRT* (BTB-APF-Informed-RRT*) algorithm. This algorithm incorporates a probabilistically adaptive bias strategy in the sampling phase to dynamically adjust sampling tendency, making sampling more targeted; in the new node expansion phase, it deeply integrates APF with the bidirectional search process based on the distance to obstacles, effectively guiding the trees to grow toward the goal and avoid obstacles. Moreover, an adaptive step size mechanism based on environmental information is introduced to enhance the algorithm's search performance in different scenarios (e.g., narrow passages and open areas). By incorporating a robotic arm collision detection model, the generated path is processed with triangle inequality pruning and cubic B-spline smoothing to obtain a shorter and smoother final path.

2. Robotic Arm Path Planning

2.1. Robotic Arm Kinematic Modeling

An industrial robotic arm is a multi-degree-of-freedom linkage mechanism. A mathematical model is required to establish the relationship between the coordinates of the robotic arm's end-effector in the world coordinate system and the joint angles. This paper takes the Lefei collaborative robot LM3 as the research object. The CAD model of the industrial robot is shown in Figure 1(a), and the robotic arm model established using the Denavit-Hartenberg (DH) method is shown in Figure 1(b). The DH parameter table of the robotic arm can be obtained as shown in Table 1.



(a) Lebai Robot Model (b) Robotic Arm D-H Model*

Figure 1: Robot Model

Table 1: Robotic Arm D-H Parameter Table

Joint*i*	$\alpha_{i\text{-}1}$/(°)	$a_{i\text{-}1}/mm$	$\theta_{i\text{-}1}$/(°)	d_i/mm
1	0	0	0	0.21583
2	1.5708	0	0	0
3	0	-0.28	0	0
4	0	-0.26	0	0.12063
5	1.5708	0	0	0.09833
6	-1.5708	0	0	0.08343

The meanings of each parameter in the table are: $\alpha_{i\text{-}1}$ representing the torsion angle of the connecting rod; $a_{i\text{-}1}$ Indicate the length of the connecting rod; di represents the offset of the connecting rod. Based on these four parameter quantities, describe the transformation matrix of the i-th connecting rod relative to the previous connecting rod i-1 using the following equation:

$$ {}^{i\text{-}1}_{\;\;i}T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1} $$

For a six-axis serial robotic arm, the homogeneous transformation matrix from the base coordinate system to the end-effector coordinate system $_{6}^{0}T$ can be obtained by multiplying the six transformation matrices corresponding to each joint.

$$ {}^{0}_{6}T = {}^{0}_{1}T(\theta_1)\,{}^{1}_{2}T(\theta_2)\,{}^{2}_{3}T(\theta_3)\,{}^{3}_{4}T(\theta_4)\,{}^{4}_{5}T(\theta_5)\,{}^{5}_{6}T(\theta_6) \tag{2} $$

Collision detection is essential for robotic arm path planning. The LM3 industrial robot consists of 6 joints, and detecting each joint can determine whether the robotic arm collides with obstacles. To simplify the model, the robotic arm links are equivalent to cylinders, and irregular obstacles are enveloped using spheres and cuboids. Finally, the collision detection task is transformed into an interference intersection problem between regular geometric bodies, which improves computational efficiency.

2.2. RRT* Algorithm

The RRT* algorithm introduces rewiring (reparenting) and neighbor connection optimization mechanisms based on the basic RRT algorithm, achieving asymptotic optimality by continuously optimizing path cost. Its core steps are as follows:

(1) Random Sampling: Generate a random sample within the configuration space.

(2) Finding the Nearest Neighbor Node: Identify the closest node in the existing tree relative to the sample point.

(3) Rewiring (Reparenting): After generating a new node xnew, the algorithm does not simply set its parent node as the nearest node xnearest. Instead, it searches for all potential parent nodes xnear within a spherical neighborhood of xnew with radius r, calculates the cost from the start point to xnew via each xnear, and selects the node with the minimum cost as xnew's parent.

(4) Neighbor Connection Optimization: After rewiring the parent node, the algorithm rechecks the xnear nodes within the neighborhood. If the cost from the start point to xnear via xnew is lower than its original cost, the parent node of xnear is reset to xnew, and its cost is updated.

By repeatedly executing these processes, the path generated by RRT* gradually approaches the optimal path as the number of iterations increases.

2.3. Bidirectional RRT* (B-RRT*)

The Bidirectional RRT* algorithm introduces bidirectional search and a greedy connect algorithm based on the RRT* algorithm. It constructs random trees at the initial and target positions respectively, and the two trees grow toward each other until they meet— at which point the path search stops and a feasible path is generated. The B-RRT* algorithm maintains two random trees T_a and T_b simultaneously: one rooted at the start point xinit and the other at the goal point xgoal. In each iteration, the algorithm alternately expands the two trees. After expanding one tree to obtain a new node x_{new}, it does not immediately attempt to connect the two trees; instead, it tries to connect x_{new} to the other tree. Specifically, it finds the node x_{naer} closest to x_{new} in the other tree. If the connection between x_{new} and x_{naer} is collision-free and the path cost after connection is lower, a path is successfully found. Functional diagram is shown in Figure 2.This method greatly improves search efficiency, especially in narrow passage environments.

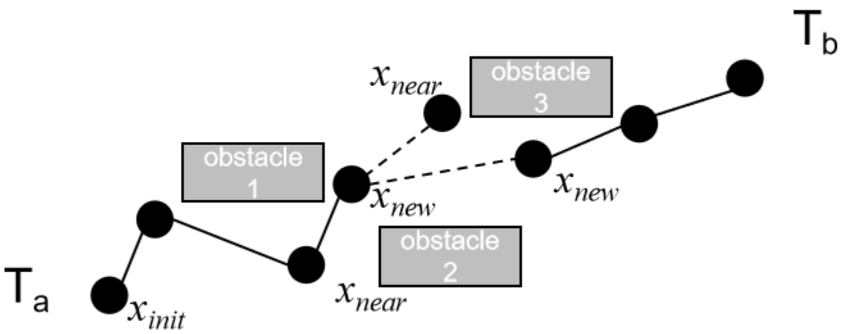


Figure 2: B-RRT* random trees

2.4. Artificial Potential Field (APF) Method

The Artificial Potential Field method models the robot's motion as movement in a virtual force field. The target point generates an attractive potential field U att(q) to attract the robot toward it:

$$U_{att}(q)=\frac{1}{2}\zeta d^2(q,q_{goal}) \tag{3}$$

Where ζ is the gravitational gain coefficient, and d(q,qgoal) is the distance from the current point q to the target point qgoal.

Obstacles generate a repulsive potential fieldUrep(q) to repel the robot away:

$$U_{rep}(q)=\begin{cases}\frac{1}{2}\eta\left(\frac{1}{d(q,q_{obs})}-\frac{1}{d_0}\right)^2, & d(q,q_{obs})\leq d_0\\ 0, & d(q,q_{obs})>d_0\end{cases} \tag{4}$$

Where η is the repulsive gain coefficient, d(q,qobs) is the distance from the current point to the obstacle, and d0 is the distance affected by the obstacle.

3. BTB-APF-Informed-RRT* Algorithm Design

3.1. Probabilistic Adaptive Target Bias Strategy

A fixed-probability target bias may fail in complex environments, as the algorithm frequently hits obstacles. This paper designs an adaptive target bias strategy: set the target bias probability pbias. During each sampling process, the algorithm selects the target point as the sample point with probability pbias; otherwise, it performs random sampling. This strategy effectively guides the trees to grow toward the target and reduces invalid sampling.

$$P_{bias}=P_{min}+(P_{max}-P_{min})\times e^{-\lambda\times n_{fail}} \tag{5}$$

where P_{max} and P_{min} represent the upper and lower bounds of the bias probability, λ denotes the decay coefficient, and n_{fail} is the number of consecutive expansion failures. When consecutive failures occur, the algorithm automatically reduces the target bias probability P_{bias}, increases the proportion of random sampling, and enhances the algorithm's ability to escape trap regions. When expansion is successful, reset $n_{fail}=0$, restore a higher P_{bias}, and strengthen target orientation.

3.2. Improved APF and RRT* Fusion Mechanism

The local minimum problem of traditional APF and its oscillation in narrow channels limit the method's performance. This paper improves APF and deeply integrates it into node expansion.

During node expansion, the direction of the APF resultant force is introduced as a component of the expansion direction, as shown in the following formula:

$$\vec{F}_{total}=\vec{F}_{att}+\vec{F}_{rep} \tag{6}$$

where $\vec{F}_{att}$ (the attractive force) points toward the target point, and $\vec{F}_{rep}$ (the repulsive force) points away from obstacles. The generation direction of the new node is determined by the weighted sum of the random direction and the resultant force direction.

To solve the problem of the target point being unreachable, the distance term d(q,qgoal) (from the current point to the target point) is introduced into the repulsion function:

$$U_{rep}(q)=\begin{cases}\frac{1}{2}\eta\left(\frac{1}{d(q,q_{obs})}-\frac{1}{d_0}\right)^2 d^2(q,q_{goal}), & d(q,q_{obs})\leq d_0\\ 0, & d(q,q_{obs})>d_0\end{cases} \tag{7}$$

When the robot approaches the goal, d(q,qgoal)→0, and the repulsive force also approaches 0—thus ensuring that the target point is the global minimum point of the potential field.

3.3. Adaptive Step Size Strategy

The step size is dynamically adjusted based on the distance from the current node to the nearest obstacle:

$$\text{StepSize}=\begin{cases}\text{Step}_{\text{min}}+(\text{Step}_{\text{max}}-\text{Step}_{\text{min}})\cdot\frac{d-d_{\text{safe}}}{d_{\text{max}}-d_{\text{safe}}}, & d\le d_{\text{max}}\\ \text{Step}_{\text{max}} & d>d_{\text{max}}\end{cases} \quad (8)$$

where d is the distance from the current node xnearest to the nearest obstacle, dsafe is the safe distance, and dmax is the maximum influence distance for step size adjustment. When the node is very close to an obstacle (d→dsafe}$), the step size approaches Stepmin for fine operation to prevent collision; when the node is in a safe area (d≥dmax), the maximum step size Stepmax is used for rapid exploration.

3.4. Path Post-Processing Optimization

3.4.1. B-Spline Smoothing

The pruned path is still a polyline and requires smoothing. B-spline curves are widely used due to their good local control and continuity. Given m+1 control points {D0,D1,...,Dm}, the expression for a k-th degree B-spline curve is:

$$C(u)=\sum_{i=0}^{m} N_{i,k}(u)D_i \quad (9)$$

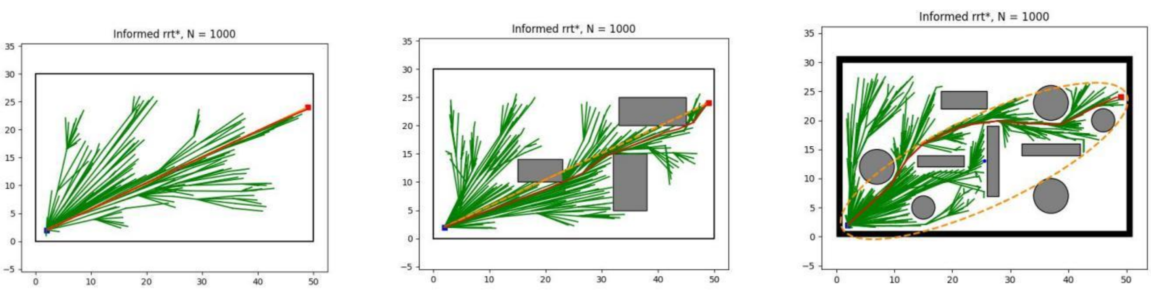
where $N_{i,k}(u)$ is the k-th degree B-spline basis function, which can be calculated using the de Boor-Cox recurrence formula.

This paper uses cubic B-spline curves (k=3), and takes the key points after path pruning as control points to fit a second-order continuous smooth path. This ensures the continuity of velocity and acceleration during the robotic arm's movement.

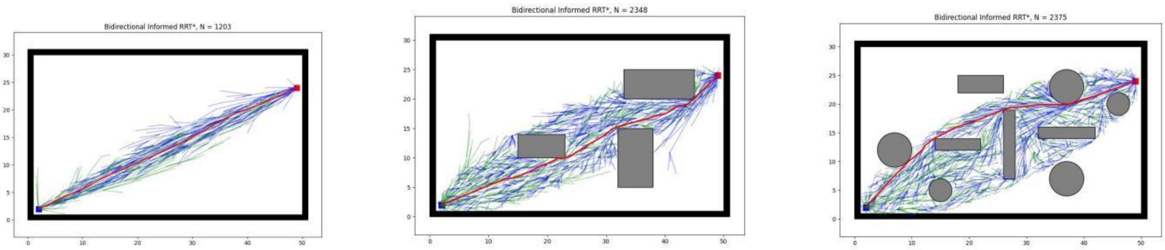
4. Simulation Experiments and Result Analysis

4.1. Experimental Environment and Parameter Settings

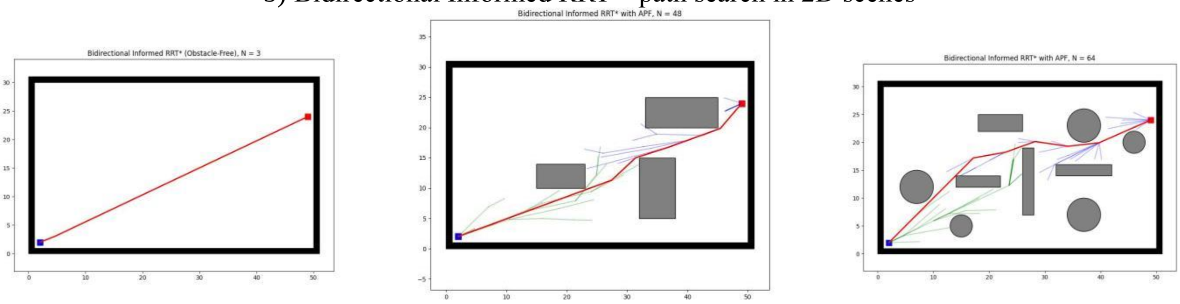
Simulations were conducted in a Python environment. The size of the 2D map was set to 50×30 units, with the start point at (2,2) and the goal point at (49,24). Three different scenario models were constructed: an obstacle-free scenario, a simple obstacle scenario, and a complex obstacle scenario. The path search results of each algorithm in the 2D scenarios are shown in Figure 3.



a) Informed RRT * path search in two-dimensional scenes



b) Bidirectional Informed RRT * path search in 2D scenes



c) The algorithm path search in this article under two-dimensional scenes

Figure 3: Path Search Results in 2D Scenarios

Table 2: Performance Comparison of Various Algorithms in 2D Scenarios

Scene	Algorithm	Average path length	Planning time	Path node count
Accessible Scene	Informed-RRT*	52.867	2.466	48
	Bidirectional-Informed-RRT*	52.078	1.045	40
	Algorithm in this article	51.894	0.105	3
Simple Scene	Informed-RRT*	55.073	5.366	60
	Bidirectional-Informed-RRT*	54.886	3.889	55
	Algorithm in this article	54.756	0.523	10
Complex Scene	Informed-RRT*	60.064	8.446	87
	Bidirectional-Informed-RRT*	59.887	6.380	68
	Algorithm in this article	58.385	0.638	18

As can be seen from Table 2, in the three different scenarios, the improved algorithm proposed in this paper showed improvements over the Informed-RRT* algorithm and the Bidirectional Informed-RRT* algorithm in terms of average path length, planning time, and number of path nodes— with particularly significant reductions in planning time and the number of path nodes.

The planned path was pruned and smoothed using cubic B-spline curves. After pruning and smoothing, the path became significantly smoother, providing a smooth and effective path for subsequent robotic arm simulation experiments and effectively avoiding oscillation in the robotic arm’s motion path.

A physical platform was built based on the simulation environment. An end-effector and a depth camera were installed on joint 6 to facilitate operation in special scenarios. The robotic arm could effectively avoid obstacles according to the pre-planned path and move from the start point to the target point without collision-demonstrating the effectiveness and practicality of the improved algorithm proposed in this paper.

5. Conclusion and Prospects

This paper proposed a robotic arm path planning algorithm based on the Bidirectional Target-Biased APF-Informed-RRT*. By integrating strategies including target-biased sampling, APF guidance, adaptive step size adjustment, and path optimization, the algorithm significantly improves the efficiency and quality of path planning. Simulation results show that the proposed algorithm outperforms traditional algorithms in terms of planning time, number of sampling points, and path quality. Comparative simulations of the improved algorithm with other algorithms in different scenarios confirm that the proposed algorithm has a faster convergence speed. Additionally, the feasibility of the algorithm on a physical robotic arm platform was verified.

Acknowledgements

Supported by the Chenguang Program of Shanghai Education Development Foundation and Shanghai Municipal Education Commission(Item Number:21CGB18).

Supported by Shanghai Vocational Education Association(Item Number:ZD202203).

Supported by the Chenguang Program of Shanghai Education Development Foundation and Shanghai Municipal Education Commission(Item Number:22CGB15).

References

[1] Wu Kai, Yang Jianzhong, Huang Si, et al. Efficient Path Planning for Robotic Arm Using Improved RRT Algorithm [J]. Mechanical Science and Technology, 2024.

[2] Liu Yang, Sun Rui, Luo Jie, et al. Path Planning for Measurement Robotic Arm Based on Improved RRT Algorithm [J]. Machine Design and Research, 2024, 40(5).

[3] Karaman S, Frazzoli E. Sampling-based algorithms for optimal motion planning[J]. The International Journal of Robotics Research, 2011, 30(7): 846-894.

[4] Kuffner J J, LaValle S M. RRT-connect: An efficient approach to single-query path planning[C]//Proceedings of the 2000 IEEE International Conference on Robotics and Automation. San Francisco, USA: IEEE, 2000: 995-1001.

[5] Gammell J D, Srinivasa S S, Barfoot T D. Informed RRT: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic[C]//2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. Chicago, USA: IEEE, 2014: 2997-3004.*

[6] Liu Xiaosong, Kang Lei, Shan Zebiao, et al. Robotic Arm Path Planning Based on Bidirectional Target-Biased APF-Informed-RRT Algorithm [J]. Journal of Electronic Measurement and Instrumentation, 2024, 38(6): 75-83. DOI: 10.13382/j.jemi.B2407196.*