A Novel Path Planning Scheme Based on Improved Bi-RRT* Algorithm

Shengwei Liu

College of Electronic Information and Automation, Tianjin University of Science and Technology, 300222, China

Miao Zhang

College of Electronic Information and Automation, Tianjin University of Science and Technology, 300222, China

E-mail: 884565943@qq.com, miaozhang@tust.edu.cn

Abstract

The Bi-RRT* algorithm is a path planning algorithm for industrial robots. In this paper, Bi-RRT* algorithm is studied and improved. The improved Bi-RRT* algorithm reduces the iteration time by introducing artificial potential field method. And, the path cost is reduced and the path smoothness is improved by introducing greedy algorithm. Finally, the improved Bi-RRT* algorithm was simulated in three dimensional environment, and the superiority of the improved Bi-RRT* algorithm was demonstrated by comparative experiments.

Keywords: Path planning, RRT, Artificial potential field, The greedy algorithm

1. Introduction

Path planning is a crucial technique for industrial robots [1], [2]. The purpose of path planning is to generate a collision-free path from the start point to the target point. Due to the complexity and variability of the planning environment, how to quickly plan a collision-free optimal path has become a hot topic in current research. The RRT algorithm [3] (Rapidly Expanding Random Tree, rapidly exploring random tree) is a stochastic algorithm that can be directly applied to the planning of non-completely constrained systems, and is particularly suited to high-dimensional systems with multiple degrees of freedom. Therefore, the RRT algorithm has an advantage over other algorithms in the three dimensional environment. Subsequently, the RRT* algorithm [4] is proposed to optimize paths through parent node re-selection and rewiring operations. In order to improve the fastness of RRT* algorithm, bidirectional RRT* (Bi-RRT*) algorithm [5] is proposed. However, the Bi-RRT* algorithm still suffers from long iteration time, high path cost and poor smoothing. Therefore, in the following, we will introduce the Bi-RRT* algorithm and improve the algorithm.

The rest of this article is organized as follows. The second section introduces the principle of Bi-RRT* algorithm and describes the improvement scheme, after which the pseudo-code of the algorithm is given. The third part conducts comparative experiments between the improved Bi-RRT* algorithm and the Bi-RRT* algorithm, and demonstrates the superiority of the improved Bi-RRT* algorithm. The fourth part summarizes the main content of this paper, and introduces future work.

2. Principles of Bi-RRT* algorithm and improvement methods.

The Bi-RRT* algorithm has a bi-directional search and optimization path. However, the randomness of its sampling points is too strong, the path optimization time is too long, and the smoothness of the generated path is too low. Therefore, we use APF algorithm to generate sampling points and reduce the randomness of sampling points. For the path part, a greedy algorithm based on triangular inequalities is used for optimization.

2.1. Bi-RRT algorithm*

For RRT algorithm, in each iteration, select a start point as the root node. Generate a random sample point Q_{rand} in the configuration space, find the nearest node Q_near to the sample point Q_{rand}, connect Q_rand and Q_near with a specified step step to generate a new node Q_new, if there is no collision then the new node will be added to the tree, and finally, judge the value of the distance domain from the end point, if it is less than the value of the set domain, then it means that a feasible path is found. Backtrack the path and output the path.

Compared to the RRT algorithm, RRT* algorithm after generating a new node to the new node as the center of the circle, to specify R as the radius of a circle, in the radius circle of the nodes in the tree, to re-select the parent node, selecting the parent node with the lowest overall path cost. As shown in Fig.1, the blue line is the base step, and the node closest to within the red circle with as the center of the circle and a radius of R is node 10, connecting node 10 to node 11 with a blue line of length of the base step. However, in order to find the node with the lowest cost, the RRT* algorithm iterates over the node 7, node 8, and node 9 within the red circle, and re-selects the parent node. From Fig. 1, node 7 is selected as the parent of the new node 11 with the lowest total cost of the path, the connection of node 11 to node 10 is deleted, and the green line is the path of node 7 connecting the new node 11, and node 7 is used as the parent of node 11, which is the process of re-selecting the parent of the node by RRT*. At the same time, nodes around in the existing tree are reconnected to that is the new parent node, which reduces the overall path cost.

Compared with the RRT* algorithm, the Bi-RRT* algorithm simultaneously constructs two trees from the start and goal points until the structures of these two trees intersect.

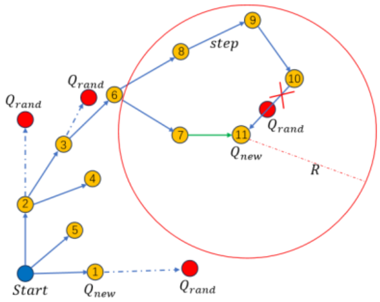

Fig.1. RRT* algorithm

2.2. APF sample

Since the RRT algorithm has too much randomness in the random generation of sampling points, we use the APF algorithm to reduce the randomness of sampling points. After the sampling points are randomly generated, we change the positions of the sampling points according to the combined force of the gravitational force of the sampling points on the target point and the repulsive force on the obstacles.

The goal point produces attraction potential energy U_{att} to the sampling point Q_{rand}, and the obstacle produces repulsion potential energy U_{rep} to the sampling point, and the combined potential energy $U = U_{att} + U_{rep}$. The Q_{rand} point advances towards the direction of the combined force of attraction of the target point and repulsion of the obstacle, F which is the negative gradient of the combined potential energy.

The point Q_{rand} is subjected to gravitational and repulsive potentials as shown in Eq. (1).

$$\begin{cases} U_{att} = \frac{1}{2}\, k_a \left[(x - x_g)^2 - (y - y_g)^2\right] \\ U_{rep} = \frac{1}{2}\, k_r \left[\frac{1}{\sqrt{(x-x_o)^2-(y-y_o)^2}} - \frac{1}{p_0}\right] \end{cases} \tag{1}$$

Where k_a represents the scaling factor of the gravitational potential field, which is used to regulate the magnitude of the attraction potential energy. k_r represents the scaling factor of the repulsive potential field, which is used to regulate the magnitude of the repulsive potential energy. (x_g, y_g) represents the co-ordinates of the target point, and (x_o, y_o) represents the co-ordinates of the obstacle. p_0 represents the maximum distance at which the obstacle exerts a repulsive force on the Q_{rand} point.

Force is a negative gradient of potential energy, as shown in the Eq. (2) and Eq. (3). The gravitational and repulsive forces are calculated as shown in the equation.

$$F = -\nabla\ U \tag{2}$$

$$\begin{cases} F_{att} = -k_a \left[(x - x_g)\,\vec{\imath} + (y - y_g)\,\vec{\jmath}\right] \\ F_{req} = -\,U'_{rep,x}\,\vec{\imath} - U'_{rep,y}\,\vec{\jmath} \end{cases} \tag{3}$$

As shown in the Fig.2, after the sample point Q_{rand} is generated, their combined gravitational and repulsive forces are calculated using the artificial market method and the Q_{rand_APF} point is regenerated.

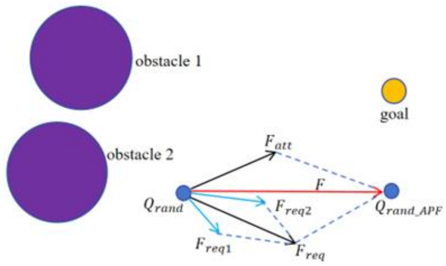

Fig.2. RRT* algorithm

The fusion of the APF algorithm in complex environments not only enhances the obstacle avoidance ability of the algorithm, but also makes the tree generated by the algorithm grow towards the target point. This reduces the iteration time and the path cost of the algorithm.

2.3. The greedy algorithm

The principle of the greedy algorithm is to achieve global optimality through local optimality, using a step-by-step approach to constructing the optimal solution. As shown in Fig.3, after the algorithm re-selects the less costly node 7 as the parent node, it then goes to reconnect its grandfather node 6, and if there is no collision, its grandfather node 6 serves as his parent node. In this way, the algorithm expands the tree branches optimally at each step, which will drastically reduce the path cost and increase the smoothness of the path.

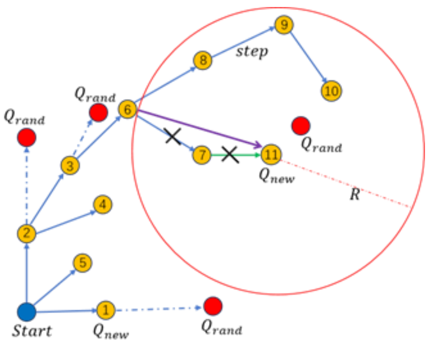

Fig.3. Greedy algorithm

Based on the above improvements, Improved Bi-RRT* algorithm is designed and the pseudo code is shown in Algorithm 1.

```
Algorithm 1: Improved Bi-RRT* algorithm
Input: Map, Xinit, Xgoal
Output: The path from start to goal: path
T1.init(), T2.init()
For i=1 to n do
    Qrand ← sample (Qstart Qgoal Map)
    Qrand ← Qrand_APF ← APF_calculate (Qrand, Map )
        Collsion (Qnew Qnear stepdynamic)=0
        Qnew ←steer (Qrand Qnear step)
        Q'near ←NearCollsion (T Qnew)
        Qmin ←chooseparent (Q'near Qnear Qnew)
        if collsion (Qnew Qminparent)
            Qmin ←Qminparent
        end
    T.add (Qmin Qnew)
    T.rewire()
        if distance (Q1new Q2new)<R
            return path
            pathcount←pathcount+1
        end
end
```

3. Simulation

In this section, two groups of three-dimensional simulations are designed in MATLAB. Two experiments are taken with 2000 sampling points in simple and complex obstacle environments, respectively. The experiment starts at [10, 10, 10] and ends at [900 900 900].

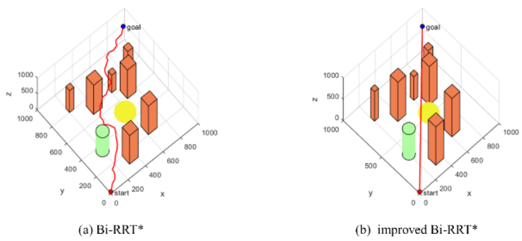

Fig.4. Simulation in simple environment

Table 1: Simulation date in simple environment

	Metric	Bi-RRT*	Improved Bi-RRT*
The first path	Path cost/pix	1852.25	1618.25
	Time/s	0.52	0.28
The last path	Path cost/pix	1791.33	1541.93
	Time/s	1.74	1.59

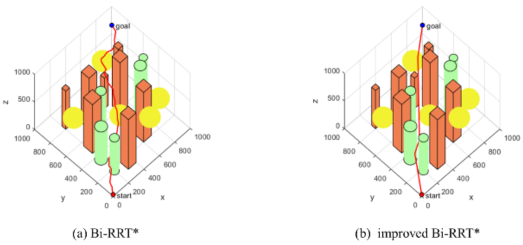

Fig.5. Simulation in complex environment

Table 2: Simulation date in complex environment

	Metric	Bi-RRT*	Improved Bi-RRT*
The first path	Path cost/pix	1793.20	1621.54
	Time/s	0.46	0.29
The last path	Path cost/pix	1756.51	1554.11
	Time/s	1.85	1.61

As shown in Fig.4 and Fig.5, improved Bi-RRT* algorithm has lower path cost and better smoothing compared to Bi-RRT* algorithm. As can be seen from Table 1, the time for the first path generation of improved Bi-RRT* algorithm is reduced by 46% and the path cost is reduced by 13% compared to Bi-RRT* algorithm in simple environment. And, the time for the last path generation of improved Bi-RRT* algorithm is reduced by 9% and the path cost is reduced by 14% compared to Bi-RRT* algorithm in simple environment. As can be seen from Table 2, the time for the first path generation of improved Bi-RRT* algorithm is reduced by 37% and the path cost is reduced by 10% compared to Bi-RRT* algorithm in complex environment. And, the time for the last path generation of improved Bi-RRT* algorithm is reduced by 13% and the path cost is reduced by 12% compared to Bi-RRT* algorithm in complex environment. These experimental results prove that the improved Bi-RRT* algorithm is more efficient compared to the Bi-RRT* algorithm.

4. Conclusion

This paper introduces the principle of the Bi-RRT* algorithm and improves the Bi-RRT* algorithm. By introducing the APF algorithm, the exploration ability of the Bi-RRT* algorithm in complex space is enhanced, and the generated sampling points are more favorable for the generation of optimal paths. In the path generation stage, the greedy algorithm is used to continuously reduce the path cost and increase the smoothness of the path. Finally, the improved Bi-RRT* algorithm is compared with the Bi-RRT* algorithm in MATLAB, and the experiment proves that the improved Bi-RRT* algorithm requires less iteration time and generates a lower path cost and has better smoothness. Since the improved Bi-RRT* algorithm does not take into account the optimization of obstacle avoidance for dynamic obstacles, in the future, we will improve the improved Bi-RRT* algorithm by improving the dynamic obstacle avoidance algorithm to improve the exploration ability in dynamic space.

References

1. Y. Wang, H. Li, Y. Zhao, et al, A Fast Coordinated Motion Planning Method for Dual-Arm Robot Based on Parallel Constrained DDP, *IEEE/ASME Transactions on Mechatronics*, 2024, 29 (3): pp.2350–2361.
2. V. N. Hartmann, A. Orthey, D. Driess, et al, Long-Horizon Multi-Robot Rearrangement Planning for Construction Assembly, *IEEE Transactions on Robotics*, 2023, 39(1): pp.239–252.
3. S. M. LaValle, J. J. Kuffner, Randomized kinodynamic planning, *The international journal of robotics research*, 2001, 20(5): pp.378–400.
4. S. Karaman, E. Frazzoli, Sampling-based algorithms for optimal motion planning, *The international journal of robotics research*, 2011, 30 (7): pp.846–894.
5. S. Klemm, J. Oberländer, A. Hermann, et al, RRT*-Connect: Faster, asymptotically optimal motion planning, *IEEE international conference on robotics and biomimetics*, 2015, pp. 1670–1677.

Authors Introduction

Mr. Shengwei Liu

In 2022, he received his Bachelor of Engineering degree from the School of Electronic and Control Engineering, North China Institute of Aerospace Engineering, China. He is pursuing a master's degree in engineering from Tianjin University of Science and Technology.

Ms. Miao Zhang

She is a postgraduate tutor of Tianjin University of Science and Technology. In 2019, she received a doctorate from the University of Windsor, Ontario, Canada. The research direction is intelligent algorithms design filters, the control system design of industrial robots and control theory.