Accelerated Informed RRT*: Fast and Asymptotically Path Planning Method Combined with RRT*-Connect and APF

Zhixin Tu[1,2], Wenbing Zhuang[1,2], Yuquan Leng[1,2],
and Chenglong Fu[1,2(✉)]

[1] Department of Mechanical and Energy Engineering, Southern University of Science and Technology, Shenzhen 518055, China
fucl@sustech.edu.com

[2] Shenzhen Key Laboratory of Biomimetic Robotics and Intelligent Systems and Guangdong Provincial Key Laboratory of Human-Augmentation and Rehabilitation Robotics in Universities, Southern University of Science and Technology, Shenzhen 518055, China

Abstract. In recent years, path planning algorithms have played a crucial role in addressing complex navigation problems in various domains, including robotics, autonomous vehicles, and virtual simulations. This abstract introduces a improved path planning algorithm called Informed RRT*-connect based on APF, which combines the strengths of the fast bidirectional rapidly-exploring random tree (RRT-connect) algorithm and the informed RRT* algorithm. The proposed algorithm aims to efficiently find collision-free paths with less iterations and time while minimizing the path length.

Unlike traditional RRT-based algorithms, Informed RRT*-connect based on Artificial Potential Fields (APF) incorporates a bidirectional connection and rewiring of a new sampling point to explore the search space. This enables the algorithm to connect both the start and goal nodes more effectively and quickly to find a initial solution, reducing the search time and provide a better initial heuristics sapling for the next optimal steps. Furthermore, Informed RRT*-connect introduces an informed sampling strategy that biases the sampling towards areas of the configuration space likely to yield better paths. This approach significantly reduces the exploration time to find a path and enhances the ability to discover optimal paths efficiently.

To evaluate the effectiveness of the Informed RRT*-connect algorithm, we conducted the simulation experiments on two different experiment

Z. Tu and W. Zhuang—Contribute equally to this work. This work was supported in part by the National Natural Science Foundation of China [Grant U1913205, 52175272], in part by the Science, Technology, and Innovation Commission of Shenzhen Municipality [Grant: ZDSYS20200811143601004, JCYJ20220530114809021], and in part by the Stable Support Plan Program of Shenzhen Natural Science Fund under Grant 20200925174640002.

H. Yang et al. (Eds.): ICIRA 2023, LNAI 14274, pp. 279–292, 2023.
https://doi.org/10.1007/978-981-99-6501-4_24

protocol. The results demonstrate that our approach outperforms existing state-of-the-art algorithms in terms of both planning efficiency and solution optimality.

Keywords: Path planning · RRT-connect · RRT* · Artificial Potential Fields (APF)

1 Introduction

The path planning problem [3] is a common and fundamental problem in the navigation and motion control of robotics, vehicles, and other mobile devices. Path planning aims to find a feasible and good path for agents in their complex interaction environment. To date, many methods have been proposed to solve the problem of path planning. Traditional approaches include potential field methods such as Artificial Potential Fields (APF) [8], which generate the path according to the attractive and repulsive forces that assumed to be generated by the goal point and obstacles. These field based methods often suffer from local optimal and lack guarantees for finding the global optimal path. Recently, researchers have explored grid-based methods that discretize the environment into a grid map and use graph-based search algorithms like A* [11] and D* [1] to find the optimal path if the path exists. While grid-based methods provide optimality guarantees, they often struggle with high-dimensional spaces, suffer from the curse of dimension, and face challenges in handling complex obstacles with irregular shapes.

Sampling-based path planning methods have emerged as a popular and effective approach to address the complexities of path planning in a wide range of domains. These methods rely on randomly sampling the configuration space to construct a graph representation of the environment, enabling the discovery of feasible paths. One prominent family of sampling-based algorithms is the rapidly-exploring random tree (RRT) algorithm and its variants, which have demonstrated remarkable success in generating collision-free paths. However, existing RRT-based algorithms often struggle to balance the trade-off between finding optimal paths and maintaining computational efficiency. One famous variants algorithm is called RRT-connect [4,7,12], which is the extension of the original RRT, focuses on connecting the start node and the goal node at the same time by two separated trees. RRT-connect can find a feasible path very quickly and saving the memory, but the resulting paths are often not the shortest. An asymptotically optimal sampling-based path planning algorithm is RRT* [5,9], which addresses the optimality drawback of RRT. RRT* refining the tree structure by rechoose the parent node of the new sampling node and rewire the tree around a distance to help reduce the redundant node. However, the drawback of RRT* is that it may be slow to converge towards an optimal solution within a limited number of iterations. In [2], Jonathan *et al.* proposed a method that utilizes the current path length and the distance between the start and goal points to generate ellipses as heuristic sampling regions to improve the performance of the

RRT* algorithm. This method significantly enhances the quality of the generated paths. However, due to the characteristics of RRT*, the algorithm requires a considerable amount of time to search for a feasible initial solution, resulting in sub-optimal optimization efficiency. Therefore, finding an optimal path within a short period of time is a prominent challenge in current path planning methods.

In this paper, we introduce a improved sampling-based path planning algorithm called Informed RRT*-connect based on APF, which combines the strengths of the RRT-connect algorithm and the optimal RRT* algorithm to address this challenge effectively. Moreover, the APF method utilizes obstacle information to guide the sampling function, allowing the sampled points to selectively deviate from obstacles while getting closer to the target point, which aims to further reduce the number of ineffective sampling attempts. Informed RRT*-connect based on APF could accelerate the process of Informed RRT* iteratively finding the optimal solution and can make up for the shortcomings of RRT-connect that cannot guarantee asymptotic optimality. Specifically, the proposed method first utilizes RRT-connect to quickly obtain a feasible path. While conducting sampling in RRT-connect, the path is optimized through the parent node re-selection and rewiring functions in RRT*. Subsequently, the initially obtained solution serves as the base solution for Informed RRT* optimization, where heuristic sampling is employed to rapidly converge the optimal path length. It is important to note that the APF method considers obstacle information as a priori for sampling bias throughout the entire planning process, thereby significantly enhancing the performance of the algorithm.

2 Preliminaries

2.1 Path Planning Problem Definition

The planning problem is defined similarly to [2,10]. Let $\chi \in \mathbb{R}^n$ be the state space, and the free space and the obstacle space are denoted as χ_{free} and χ_{obs}. Let x_{start} be the starting state and x_{goal} be the goal state. The path planning problem is to find a path $\sigma[0, T]$ from the starting state x_{start} to the goal state x_{goal} in the given space χ, denoted as

$$\sigma[0, T] \to \chi_{\text{free}}, \quad \sigma(0) = x_{\text{start}} \quad \text{and} \quad \sigma(T) = x_{goal}. \tag{1}$$

Assume α is the whole set of the feasible paths, then the optimal path σ^* with the minimum path length can be defined as

$$\begin{aligned} \sigma^* = \arg\min_{\sigma \in \alpha} & ||\sigma|| \\ \text{s.t. } & \sigma(0) = x_{\text{start}} \\ & \sigma(T) = x_{\text{goal}} \\ & \sigma(t) \in \chi_{\text{free}}(t) \quad \forall t \in [0, T]. \end{aligned} \tag{2}$$

where $||\cdot||$ is the calculator of the path length using Euclidean distance, T is the time taken for the entire path planning.

2.2 Informed RRT*

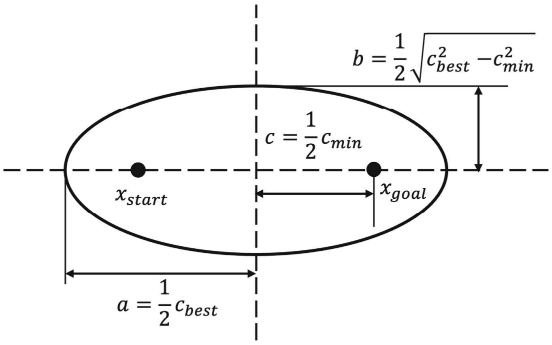
$b = \frac{1}{2}\sqrt{c_{best}^2 - c_{min}^2}$
$c = \frac{1}{2}c_{min}$
x_{start}
x_{goal}
$a = \frac{1}{2}c_{best}$

Fig. 1. The heuristic sampling set of Informed RRT* [2]. The x_{start} and x_{goal} are the focal points. a represents the semi-major axis length of the ellipse, and b represents the semi-minor axis length of the ellipse, and c represents the half length of the focal, respectively. c_{best} and c_{min} are the current path length and the theoretical shortest path length, respectively.

The informed RRT* algorithm [2,6] follows the following steps to find a feasible path and try to improve the path with the current solution. Informed RRT* primarily improves the performance of RRT* by generating an elliptical region based on the current feasible solution length and the theoretical shortest distance between the start and goal points. This region is used as a heuristic for RRT* sampling, aiming to enhance the efficiency and effectiveness of the algorithm. The informed ellipse sampling set can be expressed as the standard equation denoted by

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, \tag{3}$$

where a represents the semi-major axis length of the ellipse, and b represents the semi-minor axis length of the ellipse, as shown in Fig. 1. As the length of the discovered path continues to decrease, the corresponding ellipse sampling region becomes smaller. Consequently, the sampling region is further constrained to areas that have the potential to improve the path length. Through iterative iterations, Informed RRT* can converge towards a path that is close to optimal more rapidly than RRT*.

However, the Informed RRT* algorithm is the same as RRT* because when an initial path is not found, the elliptical sampling region can be considered infinite. This drawback is a notable limitation of Informed RRT*. It entails a significant time cost associated with searching for an initial path for optimization. Furthermore, due to the inherent randomness of RRT*, there is a possibility of obtaining a low-quality initial solution. Consequently, a substantial portion of the algorithm's execution time may be spent without achieving substantial improvements in path optimization.

3 Informed RRT*-Connect Based on APF

In this section, an accelerated Informed RRT* algorithm combined with RRT-connect and APF method is proposed to improve the Informed RRT*. The overall framework of the proposed algorithm is illustrated in Fig. 2. Its main idea is to utilize prior information about obstacles to generate a virtual potential field as a sampling bias. Additionally, it leverages the characteristics of RRT-connect, which uses bidirectional greedy connections, to rapidly find a high-quality initial solution. This approach accelerates the path optimization process of Informed RRT*. The algorithm is mainly divided into three parts: 1) APF Biased Sampling, 2) RRT*-connect (Initial path finding), 3) Informed RRT* with better initial solution (Informed heuristic sampling path optimization).

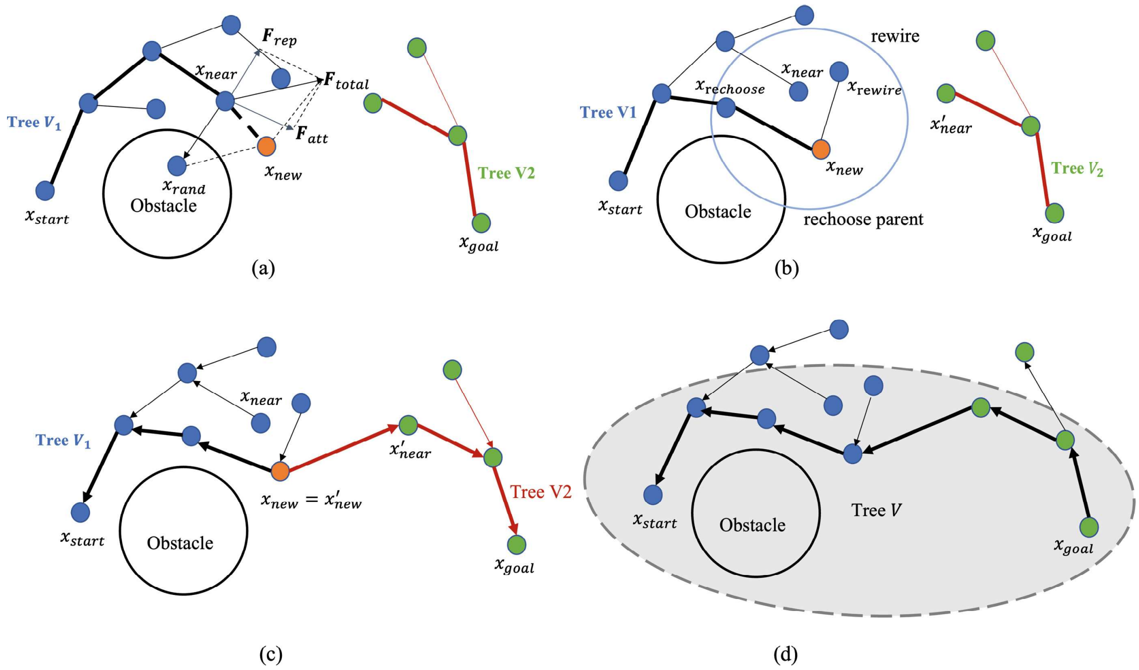

Fig. 2. The overall schematic diagram of the Informed RRT*-connect algorithm based on APF. (a) Illustration of the sampling bias guided by APF. Orange node x_{rand} is the biased sampling point guided by the APF and x_{rand} is the original sampling point. (b) Illustration of the parent node re-selection and node rewiring in RRT*-connect. The blue circle is the optimized node ranges with the rewire radius. x_{rechoose} and x_{rewire} are the selected optimized nodes. (c) Illustration of the generation of the initial path in RRT*-connect. The direction of the arrows inside the figure represents the parent-child relationships, pointing from a node to its parent node. (d) Illustration of the integration of bidirectional trees and the elliptical heuristic sampling based on the initial path. (Color figure online)

3.1 APF Biased Sampling

RRT-connect and Informed RRT* algorithm does not effectively utilize obstacle information, which can lead to low efficiency in sampling and path planning.

Therefore, it will consume a significant amount of time sampling on obstacles or in infeasible areas, especially when there are a large number of obstacles. To solve this problem, the APF method [13] is introduced into the whole sampling process, and an adaptive attractive potential field is set up at the target point, and an adaptive repulsive potential field is set up at the obstacle. Under the influence of the attractive and repulsive potential fields, newly sampled points will be biased, being pushed away from obstacles and towards the direction closer to the goal point.

The principle of APF is shown in Fig. 2(a). Assume the new sampling point is x_{rand} and x_{near} is the nearest point of the x_{rand} in the tree V_1. $\boldsymbol{F}_{\text{att}}$ is the adaptive attractive force generate by x_{goal} to x_{near}, which is proportional to the distance from the nearest point to the goal point,

$$\boldsymbol{F}_{\text{att}}(x_{\text{near}}) = k\rho(x_{\text{near}}, x_{\text{goal}}), \tag{4}$$

where $\rho(x_{\text{near}}, x_{\text{goal}}$ is the Euclidean distance from x_{near} to x_{goal}.

$\boldsymbol{F}_{\text{rep,i}}$ is the adaptive repulsion force generate by the obstacle i in the environment to x_{near}, which is inversely proportional to the distance from the nearest point to the contour of each the obstacles,

$$\boldsymbol{F}_{\text{rep,i}}\left(x_{\text{near}}\right) = \begin{cases} \beta\left(\frac{1}{d} - \frac{1}{\rho_0}\right)^2, & d < \rho_0 \\ 0, & d \geq \rho_0 \end{cases} \tag{5}$$

where $d = \rho(x_{\text{near}}, x_{\text{obs}})$ is the Euclidean distance from x_{near} to obstacle i. ρ_0 is the threshold distance of the repulsion force, which means that the repulsion force occurs only when the distance from x_{near} to the obstacle is less than the threshold.

Therefore, the resultant force F_{total} experienced at point x_{near} can be expressed as follows:

$$\boldsymbol{F}_{\text{total}}\left(x_{\text{rear}}\right) = \boldsymbol{F}_{\text{att}}\left(x_{\text{near}}\right) + \sum_{i}^{n} \boldsymbol{F}_{\text{rep,i}}\left(x_{\text{near}}\right) \tag{6}$$

where n is the total number of the obstacles in the environment.

Under the guidance of APF, the newly sampled point x_{rand} will undergo a certain displacement in the direction of the resultant force, resulting in a new biased sampling point, denoted as x_{new}. Specifically, this is achieved by synthesizing the unit vector between x_{rand} and x_{near} with the unit vector of the resultant force F_{total}. The point is then offset along this synthesized vector along the synthesized vector direction by a distance equal to the magnitude of the vector multiplied by a half of the step size ϵ, which can be calculated by

$$x_{\text{new}} = x_{\text{near}} + \frac{\varepsilon}{2}\left(\frac{x_{\text{rand}} - x_{\text{near}}}{\left|x_{\text{rand}} - x_{\text{near}}\right|} + \frac{\boldsymbol{F}_{\text{total}}\left(x_{\text{near}}\right)}{\left|\boldsymbol{F}_{\text{total}}\left(x_{\text{near}}\right)\right|}\right) \tag{7}$$

where ϵ is the step size that is a parameter of the algorithm; $\frac{x_{\text{rand}} - x_{\text{near}}}{|x_{\text{rand}} - x_{\text{near}}|}$ is the unit vector from x_{near} to the x_{rand} and $\frac{F_{\text{total}}(x_{\text{near}})}{|F_{\text{total}}(x_{\text{near}})|}$ is the unit vector in the total force direction, respectively.

The biased sampling guided by APF allows for better utilization of obstacle information in the map, reducing the number of invalid samples and greatly improving the efficiency of the sampling process. As shown in Fig. 2(a), when a new sampling point falls within an obstacle, the influence of APF enables the point to be biased towards the area outside the obstacle. This makes the sampling process effective and partially addresses the issue of low sampling efficiency in sample-based path planning algorithms. The APF-biased sampling function is shown in Algorithm 1.

Algorithm 1. $APF\_biased(x_{rand}, x_{near}, x_{goal}, x_{obs})$

```
$F_{attract} \leftarrow Calculate\_attract(x_{near}, x_{goal})$;
for $Obs$ in $x_{obs}$ do
    $F_{repulsion} += Calculate\_repusion(x_{near}, Obs)$;
$F_{total} = F_{attract} + F_{repulsion}$;
$Delta = 0.5 * \varepsilon * F_{total}/norm(F_{total})$;
$x_{new} = x_{rand} + Delta$;
return $x_{new}$
```

3.2 RRT*-Connect Fast Solution Initialization

To address the issue of lengthy computation time in finding feasible paths using the RRT algorithm and to allocate more time for improving path quality, we have combined the optimization node approach of RRT* with the fast feasibility searching capability of RRT-connect. This integration has resulted in the development of the RRT*-connect method. This method enables the rapid finding of feasible paths with higher quality, which can then serve as the starting point for optimization in Informed RRT*.

As shown in Fig. 2(b), RRT*-connect follows a similar approach to RRT-connect, where two trees V_1 and V_2, are grown outward from the start and goal points, respectively, to quickly connect the two trees and obtain a feasible path. Moreover, we have incorporated the optimization node technique from RRT* by introducing parent node re-selection and rewiring. These steps are utilized to optimize the parent node relationships between the sampled nodes and their neighboring nodes. Specifically, once the sampling point x_{new} is determined, a search is conducted within a designated rewire radius to locate nearby nodes. These nodes are then evaluated as potential parent nodes for x_{new}. The evaluation involves comparing the path length from x_{new} to the start point, selecting the node x_{rechoose} that results in the shortest path be the parent node of x_{new}. Subsequently, rewire process is to assigned the x_{new} as the parent node for the nodes found within the rewire radius, and their path length changes are examined. Among these nodes, the node x_{rewire} with the shortest path length, after rewiring its parent node to x_{new}, is selected.

Algorithm 2. *RRTstar_connect* $(x_{\text{start}}, x_{\text{goal}})$

```
 1: $V_1 \leftarrow \{x_{\text{start}}\}; E_1 \leftarrow \emptyset; G_1 \leftarrow (V_1, E_1)$
 2: $V_2 \leftarrow \{x_{\text{goal}}\}; E_2 \leftarrow \emptyset; G_2 \leftarrow (V_2, E_2)$;
 3: $i \leftarrow 0$;
 4: while $i < N$ do
 5:    $x_{\text{rand}} \leftarrow Sample(i); i \leftarrow i + 1$;
 6:    $x_{\text{nearest}} \leftarrow Nearst(G_1, x_{rand}))$;
 7:    $x_{rand} \leftarrow Steer(x_{nearest}, x_{rand})$;
 8:    $x_{new} \leftarrow APF\_biased(x_{rand}, x_{nearest}, x_{goal})$;
 9:    if $ObstacleFree(x_{nearst}, x_{new})$ then
10:       $x_{neighbor} \leftarrow Neighbor(V_1, x_{new})$
11:       $V_1 \leftarrow V_1 \cup x_{new}$;
12:       $E_1 \leftarrow E_1 \cup (x_{nearest}, x_{new})$;
13:       $Choose\_parent(x_{new}, x_{neighbor})$
14:       $Rewire(x_{new}, x_{neighbor})$
15:       $x'_{nearest} \leftarrow Nearst(G_2, x_{new}))$;
16:       $x'_{new} \leftarrow Steer(x_{nearest}, x_{new}))$;
17:       if $ObstacleFree(x'_{nearest}, x'_{new})$ then
18:          $V_2 \leftarrow V_2 \cup x'_{new}$;
19:          $E_2 \leftarrow E_2 \cup (x'_{nearest}, x'_{new})$;
20:          while NOT $x'_{new} = x_{new}$ do
21:             $x''_{new} \leftarrow Steer(x'_{new}, x_{new}))$;
22:             if $ObstacleFree(x''_{new}, x'_{new})$ then
23:                $V_2 \leftarrow V_2 \cup x''_{new}$;
24:                $E_2 \leftarrow E_2 \cup (x''_{nearest}, x'_{new})$;
25:                $x'_{new} \leftarrow x''_{new}$
26:             else
27:                break;
28:       if $x'_{new} = x_{new}$ then
29:          return $(G_a, G_b)$
30:    if $|V_2| < |V_1|$ then
31:       $Swap(G_a, G_b)$
```

After the optimization steps are performed, the node x'_{near} in tree V_2, which is closest to x_{new}, is extended towards x_{new}. The extension continues until the extended node x'_{new} and x_{new} coincide, indicating a successful connection between the two trees. At this point, a better initial solution can be obtained by combining the tree V_1 and V_2, as shown in Fig. 2(c). The proposed RRT*-connect with APF-biased sampling can be shown in Algorithm 2. Line 8, 13, and 14 is the improved part of the original RRT-connect algorithm.

3.3 Informed RRT* Optimization

After RRT*-connect completes the initial path finding, the nodes in the V_1 and V_2 trees have been connected together, but since the two trees are connected from the target point to the new sampling nodes. The parent relationship of nodes in V_2 needs to be reversed to match the V_1 to merge to a new tree V, as shown in Fig. 2(d).

Finally, an corresponding ellipse is generated according to the length of the generated initial path and the distance between the start point and the end point to guide the sampling range of RRT* in the path optimization process. Then the size of the ellipse is further reduced according to the optimized path length to further reduce the sampling range and converge to the optimal path.

The whole process of the algorithm is shown in Algorithm 3. Line 5, 6, 7, 8, and 14 is the improved content of the Informed RRT*.

4 Simulation Results and Discussion

In this section, simulation was conducted to fully verify the effectiveness of the proposed algorithm compared to different existing RRT path planning algorithms. The simulation is divided into two parts: 1) Simulation with fixed iterations. This part is to compare the average path length and the successful rate of finding the path with a fixed number of iterations. 2) Simulation with fixed path length. This part is to compare the average time and the average iterations consumed to find a fixed length of path with the three different algorithm.

4.1 Simulation with Fixed Iterations

In the first simulation, two maps with rectangular and circle obstacles is used to test the performance of four different algorithms, include Informed RRT*-connect based on APF, Informed RRT*, RRT*, RRT, and RRT-connect, under a fixed number of iterations. Specifically, we will compare the success rate of finding paths (for RRT-connect and RRT) and the path lengths obtained by the other three optimization algorithms.

The start point and end point are located in the lower left corner and upper right corner of the map, as shown in the Fig. 3. In the tow maps, the maximum number of iterations N of the five algorithms is set to 1000 and 2000, and the step size is set to 1. By running the algorithms 500 times in a loop, the corresponding evaluation metric are calculated for comparison. The specific results are shown in Table 1 and Table 2.

We can see from both Table 1 and Table 2 that Informed RRT*-connect based on APF has successfully found the path in all the iterations, which is the highest among the other algorithms. The length of the path found by Informed RRT*-connect is shortest, compared with the same asymptotically optimal RRT* and Informed RRT*. Specifically, compared to Informed RRT*, although it can eventually find a nearly optimal path, due to the acceleration of RRT*-connect in finding the initial solution, the overall planning time is around 4% shorter than Informed RRT*. It is worth noting that due to the introduction of rewire and APF sampling methods, Informed RRT*-Connect based on APF outperforms RRT-connect in finding the initial path, with an average reduction of about 18%

in iteration count. In the two maps, the average initial path length found by RRT*-Connect based on APF is 67.696 and 70.754, respectively, which is also shorter than the average path length 69.076 and 76.162 found by RRT-connect.

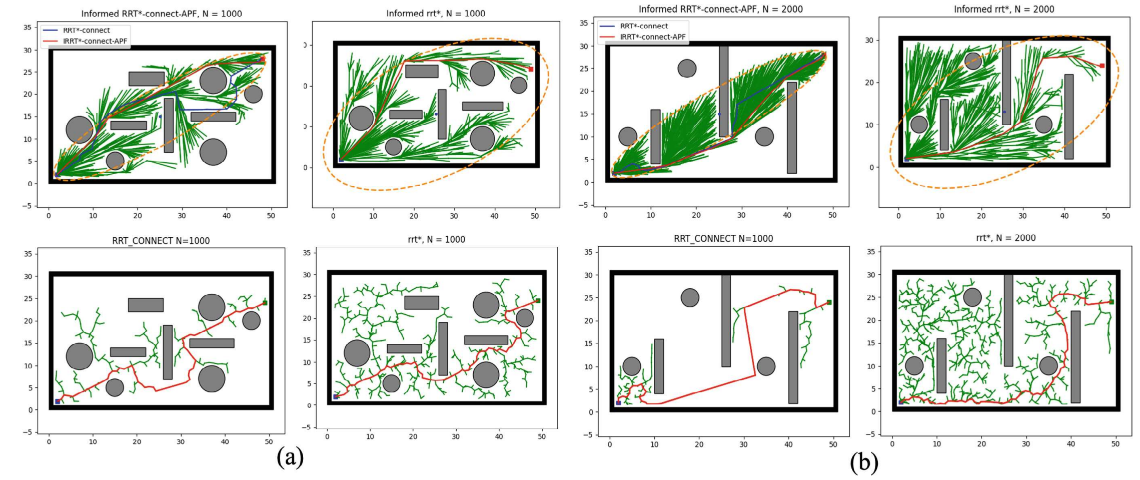

Fig. 3. Comparative illustration of the algorithm results from the first set of simulation experiments. The four images, from left to right and top to bottom, depict the path planning results of Informed RRT*-connect based on APF, Informed RRT*, RRT-connect, and RRT algorithms, respectively. The blue point is the starting point x_{start} and the red point is the goal point x_{goal}, respectively. In the first plot of each simulation, the blue line is the initial solution found by RRT*-connect, and the red line is the final optimal path. (a) The results in Map A. Map A is a 50∗30 two-dimensional map, consisting of four rectangles and four circles evenly distributed. (b) The results in Map B. Map B is a 50∗30 two-dimensional map, consisting of three rectangles and three circles. (Color figure online)

Table 1. The Performance Comparison in Map A (N = 1000)

	Ave time	Ave path length	Iter for finding path	Success rate
RRT	0.1119	70.534	619.11	0.702
RRT*	0.3996	69.135	-	0.674
RRT-connect	0.0483	69.076	354.18	0.978
IRRT*	3.800	57.051	-	0.682
IRRT*-connect-APF	3.951	56.260	291.60	1.000

Table 2. The Performance Comparison in Map B (N = 2000)

	Ave time	Ave path length	Iter for finding path	Success rate
RRT	0.1274	77.124	1039.98	0.911
RRT*	1.1225	73.940	-	0.932
RRT-connect	0.0592	76.162	773.42	0.992
IRRT*	8.6232	58.102	-	0.960
IRRT*-connect-APF	7.9003	55.930	638.90	1.000

4.2 Simulation with Fixed Path Length

The second simulation was conducted on five randomly generated $30 * 50$ maps with circle obstacles. The criteria for generating these maps included the presence of 25 circular obstacles, ensuring that the distance between any two obstacles was not less than 1 unit. The radius of the obstacles ranged from 1 to 5 units. The size of each map was set to 50 units in width and 30 units in height. To prevent any overlap between the randomly generated obstacles and the starting and ending points, the coordinates of the obstacles were constrained within the range $([4, 46], [4, 26])$. The coordinates of the starting point were $(2, 2)$, and the coordinates of the ending point were $(48, 28)$. The five random maps can be seen in Fig. 4.

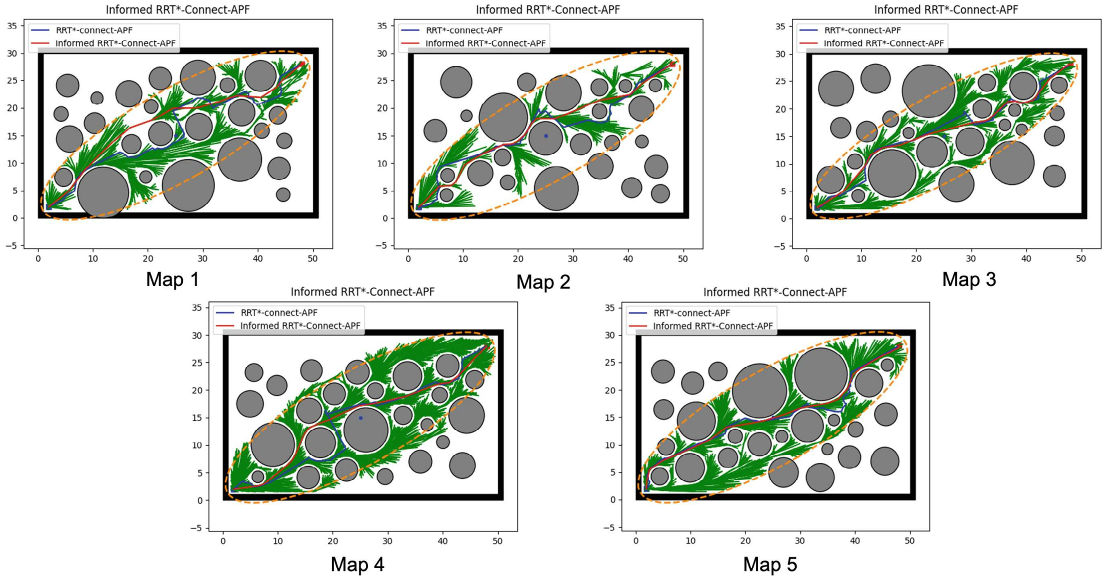

Fig. 4. Path planning results of Informed RRT*-connect based on APF on five randomly generated test maps, with the blue dots representing the start points x_{start} and the red dots representing the goal points x_{goal}. (Color figure online)

In the random maps, the start and target points are fixed, thus the theoretical shortest distance of the path can be calculated as 52.839. Due to the presence

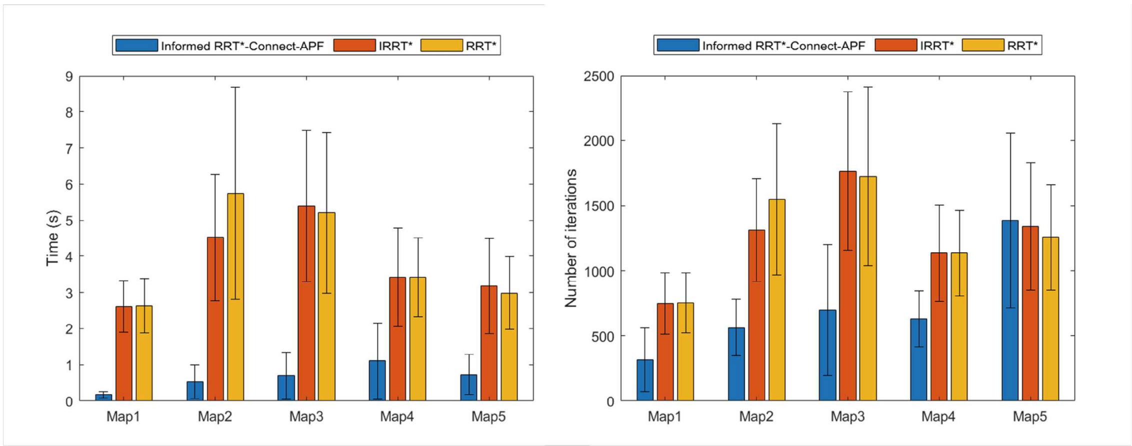

Fig. 5. Comparison of the time and iteration count required by the three algorithms to find the optimal path on five different test maps.

of obstacles, the feasible path length will increase. The path is considered to be optimal when its length is less than 110% of the theoretical shortest length, i.e. 55.480. Once an optimal path is found, the algorithm terminates, and the running time and number of iterations required are recorded. In Simulation 2, all three algorithms will be iterated 100 times on each random map.

The average runtime and iteration count of the three algorithms on each map are compared in Fig. 5. From the results, it can be observed that across all five random maps, the proposed Informed RRT*-Connect based on APF algorithm can find the optimal path in less than 1 s, while RRT* and Informed RRT* require at least 3 to 4 times more time to find the same optimal path. Regarding the iteration count, on four out of the five maps, Informed RRT*-Connect based on APF has fewer iterations compared to the other two algorithms, approximately 50% of the iteration count of the other algorithms.

The experimental results demonstrate that the proposed algorithm, utilizing the acceleration of RRT-Connect and the bias sampling of APF, can quickly find a better quality initial path, enabling rapid convergence in subsequent optimization steps. Compared to RRT* and Informed RRT*, we speculate that the main reason for the improvement is that the proposed algorithm reduces the sampling time spent in obstacle regions and globally invalid areas, thus significantly enhancing the algorithm's performance.

Algorithm 3. *Informed RRTstar_connect based on APF*

```
$V \leftarrow \emptyset$
$E \leftarrow \emptyset$
$X_{\mathrm{soln}} \leftarrow 0$
$T \leftarrow (V, E)$
$(G_a, G_b) = RRTstar\_connect(x_{start}, x_{goal})$
$V \leftarrow V_1 \cup Reverse(V_2)$
$C_{best} = Cost(V[-1])$
$x_{\mathrm{soln}} \leftarrow x_{\mathrm{soln}} \cup \{V[-1]\}$
for iteration $= 1$ to $N$ do
    $C_{\mathrm{best}} \leftarrow \min_{x \in x_{\mathrm{soln}}}(\mathrm{Cost})$
    $x_{\mathrm{rand}} \leftarrow \mathrm{Sample}(x_{\mathrm{start}}, x_{\mathrm{goal}}, C_{\mathrm{best}})$
    $x_{\mathrm{nearest}} \leftarrow \mathrm{Nearest}(V, x_{\mathrm{rand}})$
    $x_{\mathrm{rand}} \leftarrow \mathrm{Steer}(x_{\mathrm{nearest}}, x_{\mathrm{rand}})$
    $x_{new} \leftarrow APF\_biased(x_{rand}, x_{nearest}, x_{goal})$;
    if $\mathrm{CollisionFree}(x_{\mathrm{nearest}}, x_{\mathrm{new}})$ then
        $V \leftarrow V \cup \{x_{\mathrm{new}}\}$
        $x_{\mathrm{near}} \leftarrow \mathrm{Near}(V, x_{\mathrm{new}}, r_{\mathrm{RRT}^*})$
        $x_{\mathrm{min}} \leftarrow x_{\mathrm{nearest}}$
        $C_{\mathrm{min}} \leftarrow \mathrm{Cost}(x_{\mathrm{min}}) + c \cdot \mathrm{Line}(x_{\mathrm{nearest}}, x_{\mathrm{new}})$
        for $V_{\mathrm{near}} \in x_{\mathrm{near}}$ do
            $C_{\mathrm{new}} \leftarrow \mathrm{Cost}(x_{\mathrm{near}}) + c \cdot \mathrm{Line}(x_{\mathrm{near}}, x_{\mathrm{new}})$
            if $C_{\mathrm{new}} < C_{\mathrm{min}}$ then
                if $\mathrm{CollisionFree}(x_{\mathrm{near}}, x_{\mathrm{new}})$ then
                    $x_{\mathrm{min}} \leftarrow x_{\mathrm{near}}$
                    $C_{\mathrm{min}} \leftarrow C_{\mathrm{new}}$
        $E \leftarrow E \cup \{(x_{\mathrm{min}}, x_{\mathrm{new}})\}$
        for $V_{\mathrm{near}} \in \mathrm{Near}(V, x_{\mathrm{new}}, T_{\mathrm{RRT}^*})$ do
            $C_{\mathrm{near}} \leftarrow \mathrm{Cost}(x_{\mathrm{near}})$
            $C_{\mathrm{new}} \leftarrow \mathrm{Cost}(x_{\mathrm{new}}) + c \cdot \mathrm{Line}(x_{\mathrm{new}}, x_{\mathrm{near}})$
            if $C_{\mathrm{new}} < C_{\mathrm{near}}$ then
                if $\mathrm{CollisionFree}(x_{\mathrm{new}}, x_{\mathrm{near}})$ then
                    $x_{\mathrm{parent}} \leftarrow \mathrm{Parent}(x_{\mathrm{near}})$
                    $E \leftarrow E \cup \{(x_{\mathrm{parent}}, x_{\mathrm{near}})\}$
                    $E \leftarrow E \cup \{(x_{\mathrm{new}}, x_{\mathrm{near}})\}$
        if $\mathrm{InGoalRegion}(x_{\mathrm{new}})$ then
            $x_{\mathrm{soln}} \leftarrow x_{\mathrm{soln}} \cup \{x_{\mathrm{new}}\}$
return $T$
```

5 Conclusion and Future Work

In this paper, we proposed a accelerated Informed RRT* algorithm to find the optimal path in a fast way. The proposed algorithm, through the utilization of bidirectional connection and sampling bias, achieves rapid identification of the optimal path. The results from two distinct sets of simulation experiments demonstrate a significant improvement of the proposed method compared to RRT* and Informed RRT* under fixed iteration count and fixed path length conditions. This underscores the importance of effectively utilizing obstacle information and minimizing unnecessary sampling in sampling-based path planning algorithms, as it leads to substantial enhancements in both time optimality and length optimality of the generated paths.

The future work includes evaluating the algorithm's performance in the presence of dynamic obstacles and deploying the algorithm on physical robot hardware systems and testing its effectiveness.

References

1. Ferguson, D., Stentz, A.: Using interpolation to improve path planning: the field D* algorithm. J. Field Robot. **23**(2), 79–101 (2006)
2. 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. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2997–3004. IEEE (2014)
3. Gasparetto, A., Boscariol, P., Lanzutti, A., Vidoni, R.: Path planning and trajectory planning algorithms: a general overview. In: Motion and Operation Planning of Robotic Systems: Background and Practical Approaches, pp. 3–27 (2015)
4. Kang, J.G., Lim, D.W., Choi, Y.S., Jang, W.J., Jung, J.W.: Improved RRT-connect algorithm based on triangular inequality for robot path planning. Sensors **21**(2), 333 (2021)
5. Karaman, S., Frazzoli, E.: Sampling-based algorithms for optimal motion planning. Int. J. Robot. Res. **30**(7), 846–894 (2011)
6. Kim, M.C., Song, J.B.: Informed RRT* with improved converging rate by adopting wrapping procedure. Intel. Serv. Robot. **11**, 53–60 (2018)
7. Kuffner, J.J., LaValle, S.M.: RRT-connect: an efficient approach to single-query path planning. In: Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), vol. 2, pp. 995–1001. IEEE (2000)
8. Lee, M.C., Park, M.G.: Artificial potential field based path planning for mobile robots using a virtual obstacle concept. In: Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003), vol. 2, pp. 735–740. IEEE (2003)
9. Solovey, K., Janson, L., Schmerling, E., Frazzoli, E., Pavone, M.: Revisiting the asymptotic optimality of RRT. In: 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 2189–2195. IEEE (2020)
10. Wang, J., Chi, W., Li, C., Wang, C., Meng, M.Q.H.: Neural RRT*: learning-based optimal path planning. IEEE Trans. Autom. Sci. Eng. **17**(4), 1748–1758 (2020). https://doi.org/10.1109/TASE.2020.2976560
11. Warren, C.W.: Fast path planning using modified A* method. In: [1993] Proceedings IEEE International Conference on Robotics and Automation, pp. 662–667. IEEE (1993)
12. Xinggang, W., Cong, G., Yibo, L.: Variable probability based bidirectional RRT algorithm for UAV path planning. In: The 26th Chinese control and decision conference (2014 CCDC), pp. 2217–2222. IEEE (2014)
13. Xinyu, W., Xiaojuan, L., Yong, G., Jiadong, S., Rui, W.: Bidirectional potential guided RRT* for motion planning. IEEE Access **7**, 95046–95057 (2019)