Scientific Research Publishing

Open Journal of Applied Sciences, 2024, 14, 1214-1236
https://www.scirp.org/journal/ojapps
ISSN Online: 2165-3925
ISSN Print: 2165-3917

Path Planning for Robotic Arms Based on an Improved RRT Algorithm

Wei Liu[1], Zhennan Huang[2], Yingpeng Qu[3], Long Chen[1]

[1]School of Mechanical Engineering, Hunan University of Science and Technology, Xiangtan, China
[2]Officers College of PAP, Chengdu, China
[3]Beijing Millet Mobile Software Co., Beijing, China
Email: 18700187539@163.com

How to cite this paper: Liu, W., Huang, Z.N., Qu, Y.P. and Chen, L. (2024) Path Planning for Robotic Arms Based on an Improved RRT Algorithm. *Open Journal of Applied Sciences*, **14**, 1214-1236.
https://doi.org/10.4236/ojapps.2024.145079

Received: April 6, 2024
Accepted: May 11, 2024
Published: May 14, 2024

Copyright © 2024 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/

Open Access

Abstract

The burgeoning robotics industry has catalyzed significant strides in the development and deployment of industrial and service robotic arms, positioning path planning as a pivotal facet for augmenting their operational safety and efficiency. Existing path planning algorithms, while capable of delineating feasible trajectories, often fall short of achieving optimality, particularly concerning path length, search duration, and success likelihood. This study introduces an enhanced Rapidly-Exploring Random Tree (RRT) algorithm, meticulously designed to rectify the issues of node redundancy and the compromised path quality endemic to conventional RRT approaches. Through the integration of an adaptive pruning mechanism and a dynamic elliptical search strategy within the Informed RRT* framework, our algorithm efficiently refines the search tree by discarding branches that surpass the cost of the optimal path, thereby refining the search space and significantly boosting efficiency. Extensive comparative analysis across both two-dimensional and three-dimensional simulation settings underscores the algorithm's proficiency in markedly improving path precision and search velocity, signifying a breakthrough in the domain of robotic arm path planning.

Keywords

Robotic Arm, Path Planning, RRT Algorithm, Adaptive Pruning Optimization

1. Introduction

As robotics continues to evolve, path planning has garnered increasing attention and research within the field of robotics applications. Path planning involves generating a complete, collision-free path through a space filled with obstacles,

starting from an initial position and leading to a target location, connected by a series of waypoints. With the ongoing advancement of robotic technology and the increase in the degrees of freedom of robotic arms, the complexity of path planning has also escalated. The objective of path planning is to maximize the distance from obstacles in the space while ensuring the path found is the shortest possible. Sampling-based methods are considered an efficient solution, especially algorithms based on the Rapidly-Exploring Random Tree (RRT) [1]. These algorithms are particularly popular due to their effectiveness in exploring the state space and have been widely applied in robotic path planning.

Among algorithms based on random sampling, the RRT algorithm is a quintessential example. Introduced by Professor Steven M. La Valle in 1998, it is a probabilistically complete and fast-converging path planning algorithm suited for navigating through complex environments with obstacles [2]. The RRT algorithm operates by randomly sampling within a space, effectively avoiding obstacles to generate a complete search path. However, in complex environments where robotic arms operate, the RRT algorithm may not adequately address path planning challenges. The random sampling points and the paths generated can be less smooth, with too many turning points, resulting in suboptimal paths for the robotic arm.

Currently, path planning algorithms for robotic arms based on random sampling are still under active research, leading to the emergence of a wide variety of RRT algorithm variants. The primary optimization goal for most of these variants is to reduce randomness, yet they still rely on sampling methods. This approach aims to enhance the search efficiency and improve the probabilistic completeness of the algorithms. Kuffner [3] and others introduced the RRT-Connect algorithm, which operates by constructing two RRTs simultaneously—one starting from the initial point and the other from the target point. They explore the space around them, gradually moving towards each other using a simple greedy heuristic method. This strategy accelerates the search process, and once the two trees meet, a complete path from the start to the target point is formed. However, like the RRT, the RRT-Connect algorithm lacks asymptotic optimality and can only guarantee a 100% success rate in finding a path as the number of sampling nodes approaches infinity. Kuwata [4] introduced the Dubins path planning algorithm, which consists of straight lines and multiple circular arcs, leading to discontinuous path curvatures. Karaman [5] and Frazzoli proposed a variant of RRT, the RRT* algorithm, and proved its asymptotic optimality. The RRT* algorithm represents one of the most significant advancements in RRT research, taking a substantial step forward. By introducing features such as tree rewiring and optimal neighbor search, the RRT* algorithm with asymptotic optimality significantly enhances the quality of path planning for robotic arms.

To address the slow convergence issue of the RRT* algorithm, Jonathan [6], Siddhartha, and others proposed a new algorithm called Informed RRT* to enhance the efficiency of path planning. This method, building on an existing path found by the RRT* algorithm, uses the start and end points of this path as

foci to construct an ellipse with the major axis defined by the straight-line distance between these points. It then searches for paths within this ellipse, continually updating the sampling area until the optimal path is obtained. This approach retains the probabilistic completeness and optimality of the RRT* algorithm while making significant strides in speeding up convergence and improving solution quality. The RRTSmart algorithm, introduced by Fahad and colleagues, incorporates path optimization and intelligent sampling techniques [7]. It uses the first path found by RRT as a guide for intelligent sampling to explore the configuration space, resulting in straighter paths with fewer waypoints and facilitating optimal path planning for robots. Daniel [8] proposed the AM-RRT* algorithm for online path planning in complex dynamic environments, which enhances performance in obstacle-rich settings using auxiliary distance metrics. Connell D [9] developed a dynamic re-planning method based on RRT*, demonstrating excellent performance in re-planning paths and laying the groundwork for finding more optimal paths in the presence of unpredictably moving obstacles. Otte M [10] from MIT introduced RRT-X, a sampling-based asymptotically optimal single-query re-planning algorithm capable of real-time dynamic navigation in changing environments. For dynamic obstacle avoidance, Adiyatov [11] and colleagues proposed the RRTFND algorithm, which employs a greedy algorithm's heuristic to rectify paths hindered by dynamic obstacles. This method, compared with RRT and RRT*FN, validated its effectiveness in dynamic settings.

Professor Sun Fuchun [12] from Tsinghua University proposed a goal-directed version of the Rapidly-Exploring Random Tree method (RRT-GD) for redundant robotic arm path planning. This method focuses on both the state of the end effector and the motion of the joints, employing the Newton-Raphson method for kinematic inversion; the RRT-GD algorithm often achieves speeds more than ten times that of conventional RRT algorithms. Yang Hanjiang [13] and others introduced a hybrid robotic arm obstacle avoidance path planning algorithm based on joint configuration space, simplifying the models of robotic arms and obstacles and using exhaustive methods to search for collision-free paths in the serial robotic arm joint configuration space. Professor Liu Chengju [14] from Tongji University and colleagues developed an improved RRT algorithm addressing the issues of slow speed and poor effectiveness in robotic obstacle avoidance. This path planner adapts to dynamic moving obstacles with strategies like goal-directed policies, added gravitational components, and path smoothing, combined with path caching and dynamic expansion of random trees, effectively avoiding moving obstacles and ensuring path safety and real-time performance. Professor Wan Fangyi [15] from Northwestern Polytechnical University and others proposed an improved RRT* algorithm, the F-RRT* algorithm, in 2021, which optimizes paths by creating parent nodes for random points and repeatedly using the triangle inequality throughout the process, resulting in better initial solutions and faster convergence than the RRT* algorithm. Professor Meng Zhijun [16] from Beijing University of Aeronautics and Astronautics and col-

leagues introduced a new RRT-based pathfinding algorithm, Fast-RRT, aimed at quickly finding near-optimal paths by sampling only in the unexplored space of the random tree, enhancing search speed and algorithm stability. For performance issues in narrow passages, a random steering strategy was proposed. Wang [17] from the Chinese University of Hong Kong introduced a motion-constrained bidirectional rapidly exploring random tree algorithm with effective branch pruning, KB-RRT*, which employs an efficient branch pruning strategy to identify less costly parent nodes and remove nearby high-cost nodes. Zhang Y [18] and others, in 2023, proposed Bi-AM-RRT*, a fast and effective sampling-based motion planning algorithm for dynamic environments, incorporating the AM method to optimize robotic motion planning performance in dynamic obstacle environments and employing a bidirectional search strategy to reduce search time. Kashyap A K *et al.* [19]. Proposed the RA-CD-WOA algorithm in their research on humanoid robots, demonstrating robustness and effectiveness in robot navigation control. Additionally, they also introduced a method for managing humanoid robots walking on uneven terrain and dealing with both static and dynamic obstacles [20]. It is inspected by implementing a novel Enhanced DAYANI Arc Contour Intelligent (EDACI) Algorithm that designs trajectory by searching feasible points in the environment. It provides an optimum steering angle, and step optimization is performed by Broyden-Fletcher-Goldfarb-Shanno (BFGS) Quasi-Newton method that leads to guide the humanoid robot stably to the target. Additionally, Kashyap A K also proposed a hybrid trajectory planning method for artificial markets [21], which combines the faster local search BFGS quasi-Newton method with the global search APF method to achieve an efficient and faster hybrid trajectory planning technique.

2. Research on Robotic Arm Path Planning Algorithms

2.1. RRT Algorithms

The RRT algorithm is a sampling-based approach that utilizes stochastic tree expansion [22]. Beginning from a specified starting point, it constructs a search tree which is then incrementally expanded through random sampling, achieving complete coverage of the environmental space until a target point is reached. This process creates a continuous path between the initial and target points. The strengths of the RRT algorithm lie in its probabilistic completeness and rapid search capabilities. Completeness refers to the exhaustive coverage of the search space, where, given sufficient time and iterations, the algorithm conducts a comprehensive and unobstructed exploration of the entire map area, facilitating path planning between any two points.

The principle of node search in the RRT algorithm is illustrated in **Figure 1**.

The specific implementation process of the RRT algorithm is as follows:

1) Define a set of variables, input parameters, and output values. "Tree" represents the generated tree structure, and "plan" denotes the planned path. This includes the definition of the map's boundary limits and a flag for reaching the target, as

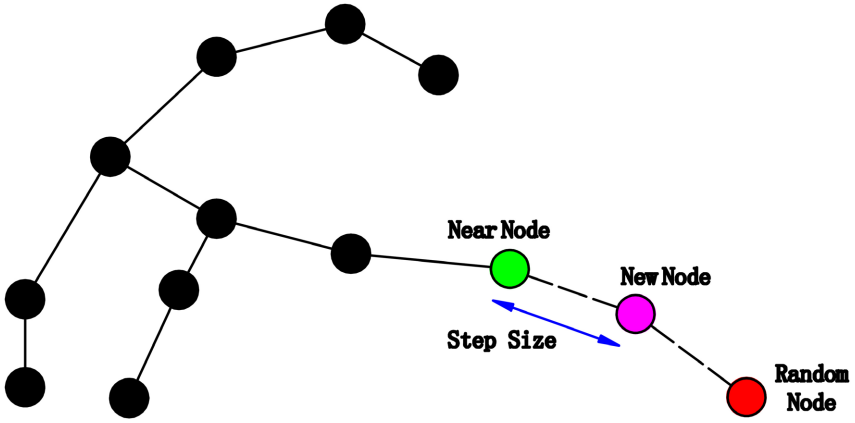


Figure 1. RRT Tree expansion process.

well as the coordinates of the initial node and the destination point. Set the sampling step size, the maximum number of search nodes, map resolution, and other parameters.

2) Initialize the parameters, creating a new node that takes the starting point of the path planning as the initial node Qstart. From this initial node, generate the random tree, "tree," setting the node's cost to zero.

3) Generate random sampling points, Qrand, within the unknown search domain based on a given probability value. These points will guide the expansion of the random tree in subsequent steps.

4) Calculate the Euclidean distance between all existing nodes in the random tree and Qrand, identifying the node closest to Qrand, named Qnearest.

5) Iterative sampling is conducted through the Extend function, progressing from Qnearest towards Qrand direction by a predetermined step size to generate a new node, Qnew. Subsequently, a collision check is performed to verify the absence of obstacles along the path from Qnearest to Qnew. If the path is clear, Qnew is incorporated into the tree structure. Encountering an obstacle halts expansion, prompting a return to step 2 for repetition, continuing this process until a clear, complete path connecting to the target point is established.

During execution, the algorithm continuously generates new nodes and attempts to connect them until it finds a path from the initial node to the target point or reaches the limit of maximum search node attempts.

Several characteristics of the RRT algorithm:

1) The RRT algorithm is inclined towards expansion in unexplored areas; it generates nodes in space through random sampling, obviating the need for a global map.

2) The RRT algorithm possesses the characteristic of probabilistic completeness, meaning that as the number of iterations increases, it can always guarantee finding a sampling point in the target area with sufficient probability, while avoiding entrapment in local optima.

3) The RRT algorithm is fast in searching, particularly in simple environments where it can quickly find a complete path. However, in more complex settings,

the RRT may require numerous iterations to discover a feasible path, thereby increasing computational costs.

The RRT algorithm, owing to its unique advantages, is widely applied in the field of path planning. Nevertheless, its drawbacks are also apparent, as evidenced by Figure 1, which shows the paths found by the algorithm to be excessively long. Consequently, numerous scholars have conducted extensive research based on the RRT algorithm, proposing various improved algorithms to mitigate its shortcomings.

Although the probabilistic completeness of RRT ensures that a path can be found given sufficient time, this does not imply that the path found by the RRT algorithm is optimal or most efficient. The randomness inherent in the RRT algorithm can lead to paths that are not ideal in terms of length and shape, especially in applications requiring the rapid identification of high-quality paths within complex environments.

2.2. RRT* Algorithm

The RRT* algorithm generates globally optimal paths by converging to the optimum within the tree structure, while incorporating heuristic strategies to avoid local optima. Based on its probabilistic completeness, it also exhibits asymptotic optimality. RRT* conducts a search among the new node's neighbors to select the parent node with the lowest path cost and employs a rewiring process for further optimization of the path cost. With its property of asymptotic optimality, the algorithm ensures convergence to the optimal solution as the runtime increases.

Under collision detection conditions, the latest sampling point, NewNode, expands with a radius r, with all nodes within this sampling space being subsets of the new node. Sequentially, the cost sums from the initial coordinate point to the nearest node and to the new node are calculated. The node with the lower cost is chosen as the parent of the new node. If there are no obstacles between the nearest node and the new node, then the new node is added to the tree structure. This iterative process continues until a feasible path is found or requirements are met. The following formula represents the search radius:

$$r = \gamma \left(\frac{\log n}{n} \right)^{1/d} \tag{2.1}$$

In the formula: r represents the search radius; γ is a planning constant based on the environment; n denotes the dimensional count of the space for robotic arm path planning; and d stands for the dimensionality of the configuration space.

The process of reselecting parent nodes in the RRT* algorithm is illustrated in Figure 2. The numbers connecting the nodes in the diagram represent the path cost between two nodes, which can be expressed through the Euclidean distance. The formula for calculating Euclidean distance is as follows:

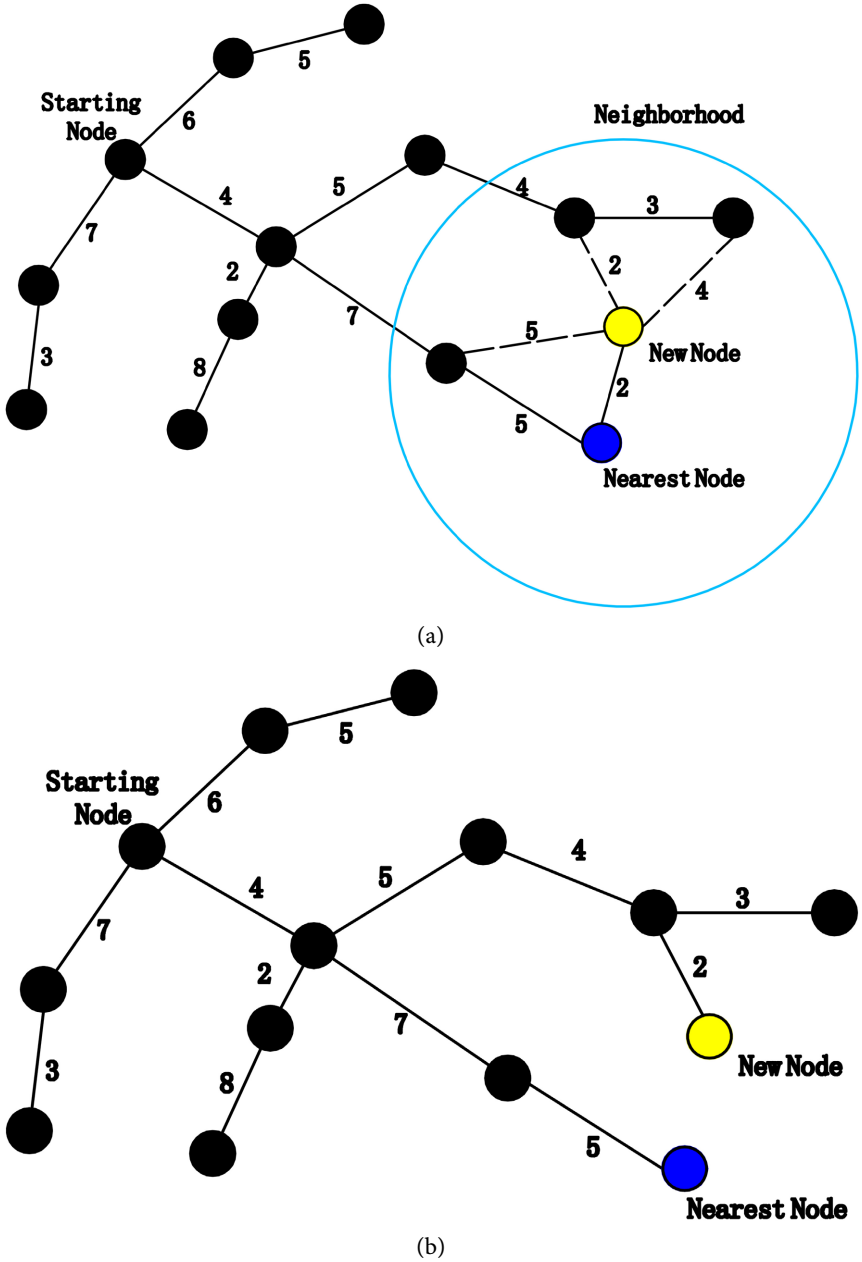


Figure 2. The process of re-selecting the parent node.

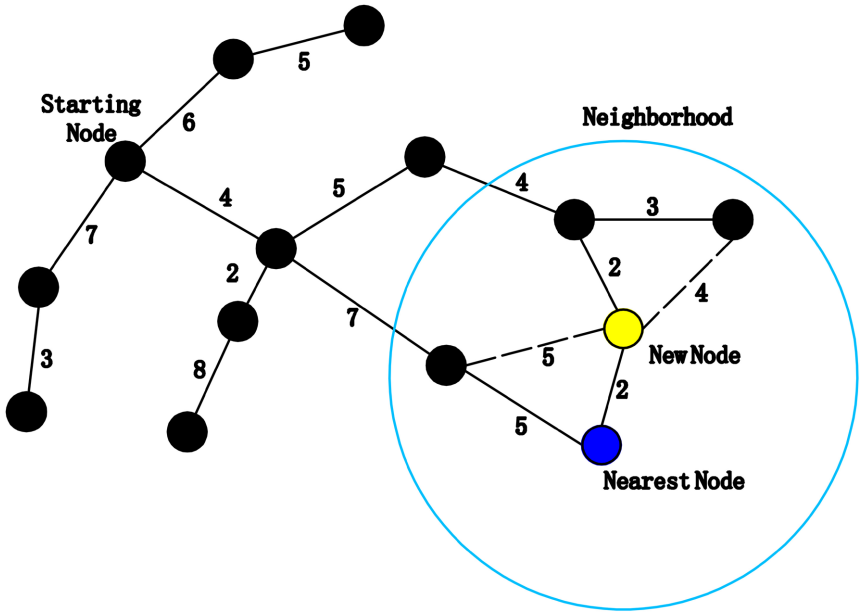
$$h = \sqrt{\left(x_1 - x_2\right)^2 + \left(y_1 - y_2\right)^2} \tag{2.2}$$

In the formula: h-represents the Euclidean distance between two nodes; $\left(x_1, y_1\right)$ and $\left(x_2, y_2\right)$ are the coordinate values of the two nodes, respectively.

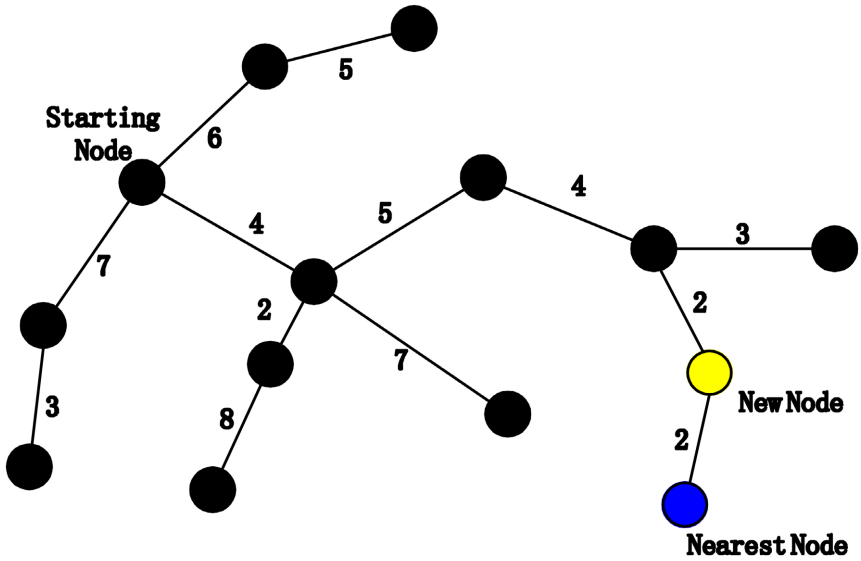
From **Figure 2(a)**, it is evident that there are multiple paths from the initial node to the new node, allowing for the selection of the optimal path based on different costs along the way. Through calculation, the optimal path cost is determined to be 4 + 5 + 4 + 2 = 15. The costs for other paths are 4 + 7 + 5 = 16, 4 + 5 + 4 + 3 + 4 = 20, and 4 + 7 + 5 + 2 = 18, respectively. Hence, the optimal path has the minimum cost. At this juncture, the nodes on the optimal path can

be designated as the new node's parent nodes, replacing the originally nearest node, as shown in Figure 2(b).

In the RRT* algorithm, each node is assigned a path cost. When a new node is generated, the algorithm checks not only for the shortest path from the new node to existing ones in the tree but also whether connecting existing nodes to the new one could reduce their costs. If such a change decreases the path cost, a rewiring process is implemented, optimizing the search tree's structure, shortening the path length, and enhancing search efficiency. This rewiring process involves reconnecting nodes on the search tree, improving the efficiency of the entire search process, as depicted in Figure 3.



(a)



(b)

Figure 3. Schematic diagram of node rewiring.

2.3. Informed RRT* Algorithm

The Informed RRT* algorithm is an improvement upon the RRT* algorithm, integrating heuristic sampling methods to enhance the efficiency and quality of path planning. Once an initial path is found, Informed RRT* replaces global uniform sampling with a heuristic elliptical sampling approach to reduce ineffective sampling nodes and optimize path search. By defining an elliptical sampling region based on the length of the initial path, Informed RRT* restricts the sampling process within this ellipse. The ellipse is centered around the generated path, with its foci at the start and end points, and its major axis equal to the length of the current shortest path. In subsequent iterations, new sampling points are generated exclusively within this elliptical region. This method effectively narrows the search space, improves search efficiency, and is more likely to generate new nodes within the optimal path region. The transformation of the sampling region is illustrated in **Figure 4**.

The algorithm first utilizes the RRT* algorithm to determine an initial path, obtaining the initial optimal path cost C_{best}. Then, based on parameters such as C_{best}, an elliptical sampling region is constructed, within which sampling operations are performed. As the value of C_{best} gradually decreases, the sampling range of the ellipse simultaneously shrinks, aiding in the gradual refinement and determination of an effective planning path. The standard equation of the ellipse is given by Equation (2.3).

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}=1 \tag{2.3}$$

In the equation, a and b are nonzero constants. The coordinates of the ellipse foci are $(c, 0)$ and $a^2=b^2+c^2$.

Utilizing the properties of ellipses, namely that the distance between the two foci of an ellipse is less than the sum of the distances from any point outside the ellipse to the two foci, and greater than the sum of the distances from any point inside the ellipse to the foci, the planning regarding the elliptical sampling region in the Informed RRT* algorithm is outlined as follows:

In path planning, set the initial node and the target node as the two foci of the ellipse, letting

$$\begin{cases} a=\dfrac{C_{\text{best}}}{2} \\ c=\dfrac{C_{\text{min}}}{2} \end{cases} \tag{2.4}$$

Substituting (2.4) into (2.3) yields:

$$b=\frac{\sqrt{C_{\text{best}}^2-C_{\text{min}}^2}}{2} \tag{2.5}$$

The elliptical model for sampling in the Informed RRT* algorithm can be seen in **Figure 5**.

One of the main advantages of the Informed RRT* algorithm is its asymptotic

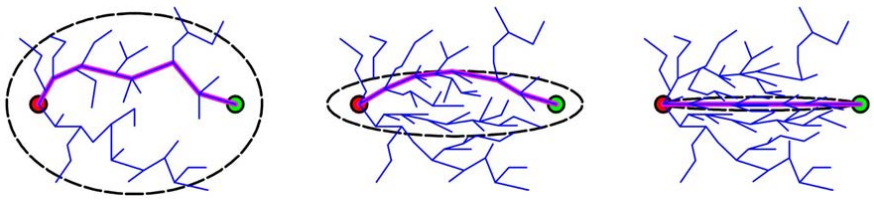


Figure 4. Schematic diagram of sampling area transformation of Informed RRT* algorithm.

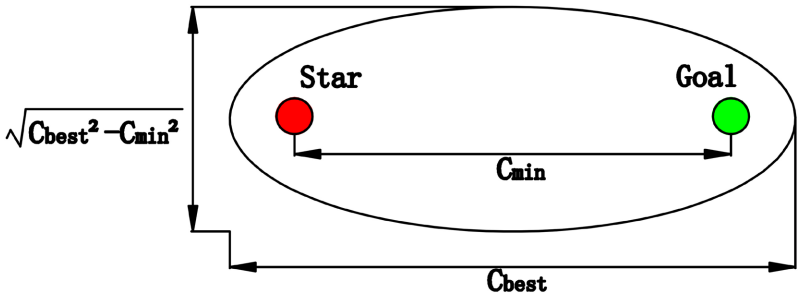


Figure 5. Informed RRT* algorithm sampling range selection.

optimality, and compared to the original RRT* algorithm, it can more effectively narrow down the search space, thereby accelerating convergence. On the other hand, the efficiency of the algorithm still depends on the sampling strategy and parameter settings. For instance, in high-dimensional spaces, determining an appropriate elliptical search area becomes more challenging, which can impact the performance of the algorithm. Moreover, in overly complex environments, the algorithm may require more time to converge to the optimal solution, leading to an increase in the number of sampled nodes and extended search times. Additionally, the performance of the algorithm heavily relies on the accurate calculation of heuristic information, which may be difficult to achieve in certain complex or dynamically changing environments.

3. Research on Improved RRT Algorithms

The improved RRT algorithm is a path planning algorithm that integrates information heuristic search, adaptive pruning optimization strategies, and dynamic elliptical region sampling. Its aim is to enhance the efficiency and quality of path planning through more precise sampling methods and path optimization techniques, achieving superior path planning even in high-dimensional or complex environments.

3.1. Definition of the Robotic Arm Path Planning Problem

State Space: This refers to the set representing all possible positions of the robotic arm during its movement, defined as R. Qfree represents the obstacle-free region within the space, and Qobs denotes the space occupied by obstacles. Qstart is defined as the initial state of the robotic arm's path, and Qgoal as the target state of the arm, where both Qstart and Qgoal are included in Qfree.

Sampling Function (Sample): In path planning, the coordinates of required nodes are obtained through random sampling. This method of random sampling is the most common approach, ensuring that nodes are uniformly distributed.

Distance Function (Distance): This function calculates the path cost between two node states, assuming the region between them is free of obstacles. The cost between two nodes is determined using the Euclidean distance, as illustrated in Equation (2.2).

Nearest Node Function (Nearest): This function retrieves the number of nodes in a tree structure, identifies a new random point to expand the tree, and calculates the Euclidean distance between this node coordinate and the first node in the tree structure, setting this as the initial value for the minimum distance. It then continues to traverse through the nodes in the tree structure, calculating the distance between the current node and the random point, to find the node Qnearest, which is closest to the random point in terms of Euclidean distance, thereby identifying the nearest node. If this distance is less than the minimum distance, the function updates the minimum distance accordingly.

Collision Detection Function (Obstacle): This is a typical Boolean function used to determine whether the connection between two nodes is obstructed by obstacles, *i.e.*, whether the path would collide with any obstacles. The function outputs true when there are obstacles between the two nodes that cause a collision; it outputs false when there is no collision.

3.2. Adaptive Pruning Optimization Strategy

The improved RRT algorithm incorporates an adaptive pruning optimization mechanism, adding an adaptive pruning threshold. This means the algorithm does not solely rely on the cost of the current optimal path to determine pruning criteria but dynamically adjusts this threshold based on actual conditions encountered during the search process. It also takes into consideration the characteristics of obstacles in different environments, such as the density of obstacles, the complexity of the path, and various dimensions, to dynamically adjust the pruning threshold. In areas dense with obstacles, it may be necessary to lower the pruning threshold to delay pruning and preserve more potential path options. Additionally, the algorithm integrates a local path re-evaluation algorithm to assess paths in the existing search tree, deciding whether to retain or modify them.

This optimization strategy can directly manipulate the tree structure, effectively reducing the search space and further enhancing the efficiency of the algorithm. Due to the continuous pruning and path optimization process, the improved RRT algorithm can shorten the path length and enhance path smoothness. The algorithmic process of its adaptive pruning strategy is shown in **Table 1**.

3.3. Dynamic Elliptical Search Strategy

The improved RRT algorithm further optimizes based on the elliptical search

Table 1. Pruning optimization strategy algorithm process.

Adaptive Pruning Optimization Strategy	
Input: T, C_best Output: T_pruned	
1	T_pruned ← ∅
2	T_pruned ← Qstart;
3	for Q = 1 to N do
4	if Cost(Q) > C_best, then
5	Continue;
6	T_pruned.add(Q);
7	for Q in T_pruned:
8	for Q_child in Q.children;
9	if Q_child.cost > C_best:
10	T_pruned.remove(Q_child)
11	end if
12	end for
13	end for
14	return T_pruned

strategy of the Informed RRT* algorithm by using a dynamically adjusted elliptical sampling area to guide the path search. This elliptical area is determined based on the current Cbest. Sampling within the ellipse means that all new sampling points are more likely to contribute to a shorter path. As the algorithm progresses and shorter paths are discovered, the ellipse correspondingly shrinks. Concurrently, the random tree outside the ellipse undergoes pruning in combination with the pruning strategy, ensuring that the entire search process remains within the ellipse, further focusing on areas likely to yield shorter paths. Compared to the elliptical search strategy of the Informed RRT* algorithm, this approach is more efficient in finding and maintaining the optimal path, thus enhancing the quality of path planning. It not only limits the sampling area but also dynamically adjusts this area to reflect real-time path information. Through pruning, it directly affects the structure of the tree, making the search process more focused and efficient.

When a shorter path is discovered, the major axis length of the ellipse is further updated, thereby narrowing the sampling area. As a result, as the algorithm progresses, the sampling space gradually concentrates in the area most likely to yield the optimal path. The dynamic ellipse can be adjusted more precisely with each discovery of a superior path. By integrating an adaptive pruning strategy, the search focus can be more effectively concentrated around the potential areas near the current optimal path. This aids in rapidly refining and optimizing the path. The dynamic ellipse search strategy is shown in Table 2.

Table 2. Dynamic ellipse search strategy process.

Dynamic Elliptical Search Strategy	
Input: T, S_start, S_goal Output: Path	
1	T = Tree(start) t
2	Cbest = ∞;
3	ellipse = space;
4	while not stop_condition()
5	if Cbest < ∞;
6	Update ellipse(Cbest);
7	q_{rand} = Sample_point()
8	$q_{nearest}$ = Nearest node(q_{rand});
9	q_{new} = Extend tree($q_{nearest}$, q_{rand}):
10	if is valid(q_{new})
11	T.add(q_{new});
12	cost = calculate cost(q_{new})
13	if cost < Cbest
14	Cbest = cost;
15	end
16	return get_best_path()

The Update function is used to update the elliptical area, the Sample function performs sampling within the ellipse, and the Extend function is responsible for incorporating new sampling points into the search tree.

4. Path Planning Simulation Validation

To demonstrate the effectiveness and efficiency of this method, a large number of comparative experiments were conducted in different simulation environments. This section presents the details of the experiments and compares and discusses the results. The experiments analyzed and compared the RRT, RRT*, Informed RRT*, and the improved RRT algorithms through simulation experiments in simple, complex, and maze environments, providing detailed comparisons. The simulation hardware used was a Lenovo ThinkBook laptop, running 64-bit Windows 11 Home Edition, with an Intel(R) Core(TM) i5-13500H processor. The simulation platform was Unity 2021. The simulation environments are illustrated, where red represents the starting point, green represents the target point, and black rectangles represent obstacles.

The parameters in the environment were set to keep the starting and target points consistent. The step length between nodes was set to 1, and the maximum number of nodes in the algorithm's search tree as well as the number of iterations N was set to 5000. The target direction probability was set to 10, meaning

that every 10th random point would not be random but instead placed at the target location, based on empirical values. The experimental results will analyze the path planning outcomes from three aspects: path length, time, and success rate. These three parameters represent the average values obtained from running each of the four algorithms 50 times.

4.1. Two-Dimensional Simulation Experiment Analysis

In the three environments, the RRT, RRT*, Informed RRT*, and improved RRT algorithms were validated for path planning. The maximum number of nodes in the algorithm's search tree, as well as the iteration count N was set to 5000, with a search radius of 2. Each set of algorithms was run 50 times to obtain average values.

1) Path Planning in Environment 1

In obstacle environment 1, a path planning instance is illustrated in Figure 6. Here, the starting point is marked in red, with coordinates set at (−7.33, 6.57), and the target node is marked in green, with coordinates set at (7.43, −6.18).

From Figure 6, it is evident that in the context of the obstacles in Environment 1, the improved RRT algorithm performs the best. The path it plans fully adheres to the edges of the obstacles, and all sampled nodes are located within the ellipse. The average values of path length, search time, and search success rate for the four algorithms in Environment 1 are presented in Table 3.

From Table 3, it is observed that the improved RRT algorithm yields the best results among the four algorithms, proving its superior performance under the

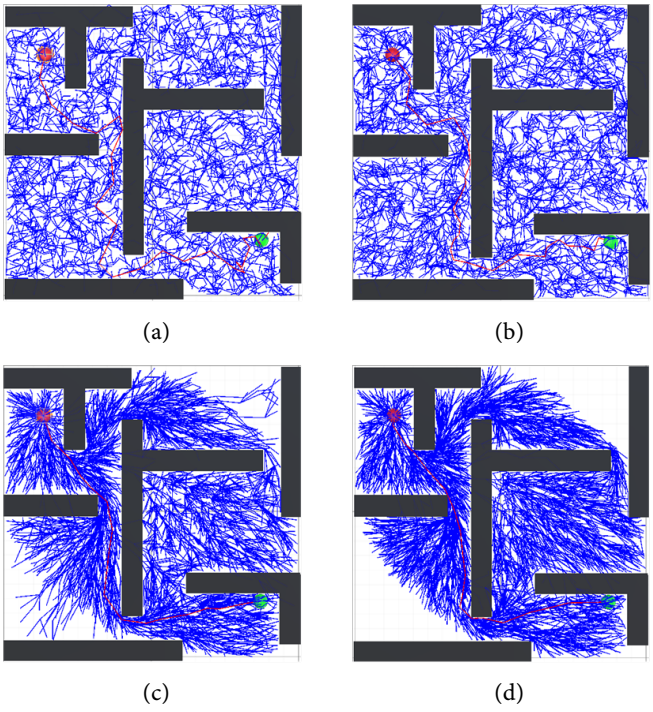


Figure 6. Path planning in environment 1: (a) RRT algorithm, (b) RRT* algorithm, (c) Informed RRT* algorithm, (d) Improved RRT algorithm.

Table 3. Performance comparison of four algorithms in environment 1.

Algorithms	Step Size	Path Length/m	Search Time/s	Search Success Rate
RRT	1	37.99	9.66	100%
RRT*	1	33.64	12.07	100%
Informed RRT*	1	27.18	10.59	100%
Improved RRT	1	25.33	8.18	100%

obstacle conditions of this environment. Compared to the RRT, RRT*, and Informed RRT* algorithms, the improved RRT algorithm achieved reductions in path length by 33.32%, 24.70%, and 6.81%, respectively. The search times were reduced by 15.32%, 32.22%, and 22.75%, respectively. Given the relatively simple placement of obstacles in Environment 1, all four algorithms achieved a 100% path success rate.

2) The path planning in Environment 2

In Environment 2, which is cluttered with obstacles, one instance of path planning is shown in Figure 7. The starting point is marked in red, with coordinates set at (−6.56, 6.2), while the goal node is marked in green, with coordinates set at (5.71, −5.9).

From Figure 7, it can be observed that under the obstacles in Environment 2, the path found by the improved RRT algorithm is optimal, seamlessly fitting around the obstacles, while the sampled nodes still remain within the ellipse. The average values of path length, search time, and search success rate parameters for the four algorithms in Environment 2 are presented in Table 4.

In Environment 2, obstacles are characterized by their chaotic and disordered nature, which necessitates higher standards for pathfinding. As evidenced in Table 4, the enhanced RRT algorithm remains superior in all metrics compared to its predecessors. Specifically, when compared to the original RRT algorithm, the path length has been reduced from 35.92 meters to 23.27 meters, marking an improvement of 35.22%. In comparison to RRT*, the improvement is from 32.16 meters to 23.27 meters, which is a 27.64% enhancement. Against Informed RRT*, the reduction is from 26.70 meters to 23.27 meters, translating to a 12.85% improvement. In terms of search time, the refined algorithm shows a reduction of 25.69%, 47.73%, and 45.49% respectively when compared to the first three algorithms. Moreover, within the context of complex and unorganized obstacles, both RRT and RRT* algorithms exhibit a decline in path success rates.

3) The path planning in Environment 3

Obstacle Environment 3 represents a more complex maze configuration. The path planning outcomes of the four algorithms for a specific instance are illustrated in Figure 8. In this scenario, the starting point is set at coordinates (−6.56, 6.2), while the target node is positioned at (5.71, −5.9).

Figure 8 demonstrates that the paths generated by the Informed RRT* and P-Informed RRT* algorithms are shorter than those produced by the RRT and RRT* algorithms. A comparison of the four algorithms across 50 simulation

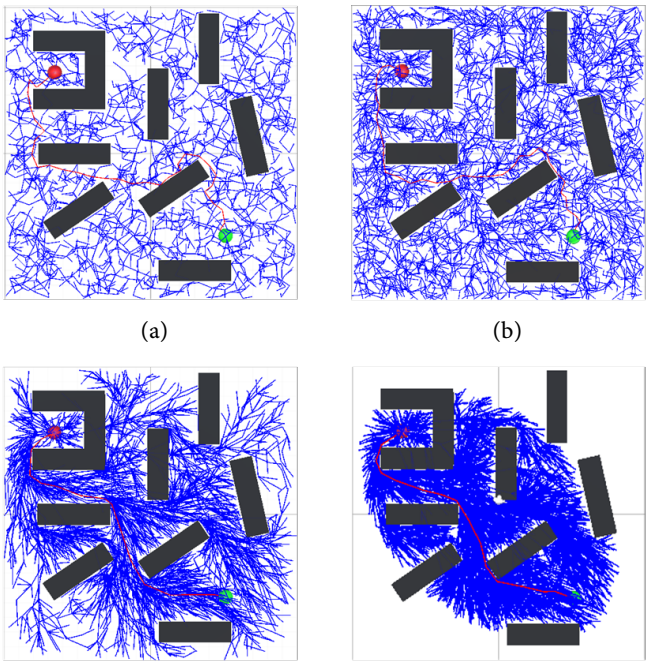


Figure 7. Path planning of four algorithms in environment 2: (a) RRT algorithm, (b) RRT* algorithm, (c) Informed RRT* algorithm, (d) Improved RRT Algorithm.

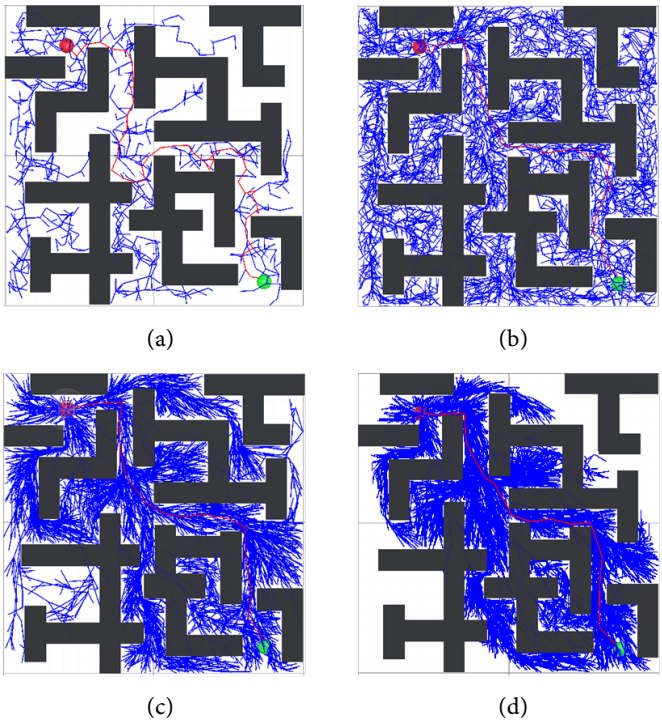


Figure 8. Path planning of four algorithms in environment 3: (a) RRT algorithm, (b) RRT* algorithm, (c) Informed RRT* algorithm, (d) P-Informed RRT* algorithm.

trials in Environment 3 is presented in Table 5, showing their performance metrics.

Table 4. Performance comparison of four algorithms in environment 2.

Algorithms	Step Size	Path Length/m	Search Time/s	Search Success Rate
RRT	1	35.92	8.37	85%
RRT*	1	32.16	11.90	88%
Informed RRT*	1	26.70	11.41	100%
Improved RRT	1	23.27	6.22	100%

Table 5. Performance comparison of four algorithms in environment 3.

Algorithms	Step Size	Path Length/m	Search Time/s	Search Success Rate
RRT	1	41.52	9.66	74%
RRT*	1	36.16	12.07	81%
Informed RRT*	1	25.11	10.60	100%
Improved RRT	1	24.19	5.27	100%

The data indicates that, compared to the RRT, RRT*, and Informed RRT* algorithms, the improved RRT algorithm achieved a reduction in path length by 41.74%, 33.10%, and 3.66%, respectively. In terms of search time, reductions were 45.45%, 56.34%, and 50.28%, respectively. Regarding the success rate of searches, the RRT and RRT* algorithms had success rates of 74% and 81%, respectively. This suggests that, in the more complex maze environment of Environment 3, the performance of the RRT and RRT* algorithms in path planning is relatively inferior compared to the other environments.

In summary, a comparative analysis of path planning data across three environments of varying complexity was conducted for four algorithms: RRT, RRT*, Informed RRT*, and an improved RRT algorithm. The comparison focused on three aspects: path length, search time, and success rate. Figure 9 and Figure 10 reveal that the improved RRT algorithm significantly outperforms the others in terms of path length and search time. On average, it reduced the path length by 35.77%, 28.48%, and 7.77% compared to the other three algorithms, respectively. Search time was reduced by 28.82%, 45.43%, and 39.51%. Moreover, this algorithm maintained a high success rate, demonstrating its stability and reliability under various conditions.

4.2. Three-Dimensional Simulation Experiment Analysis

To further verify the effectiveness of the improved RRT algorithm, this study constructs a three-dimensional simulation environment within Unity3D software for path planning of a robotic arm. The three-dimensional obstacle environment is depicted in Figure 11. Similar to the two-dimensional simulation algorithms, a comparison and analysis of data results are conducted for the RRT, RRT*, Informed RRT*, and the improved RRT algorithms within this constructed three-dimensional simulation environment.

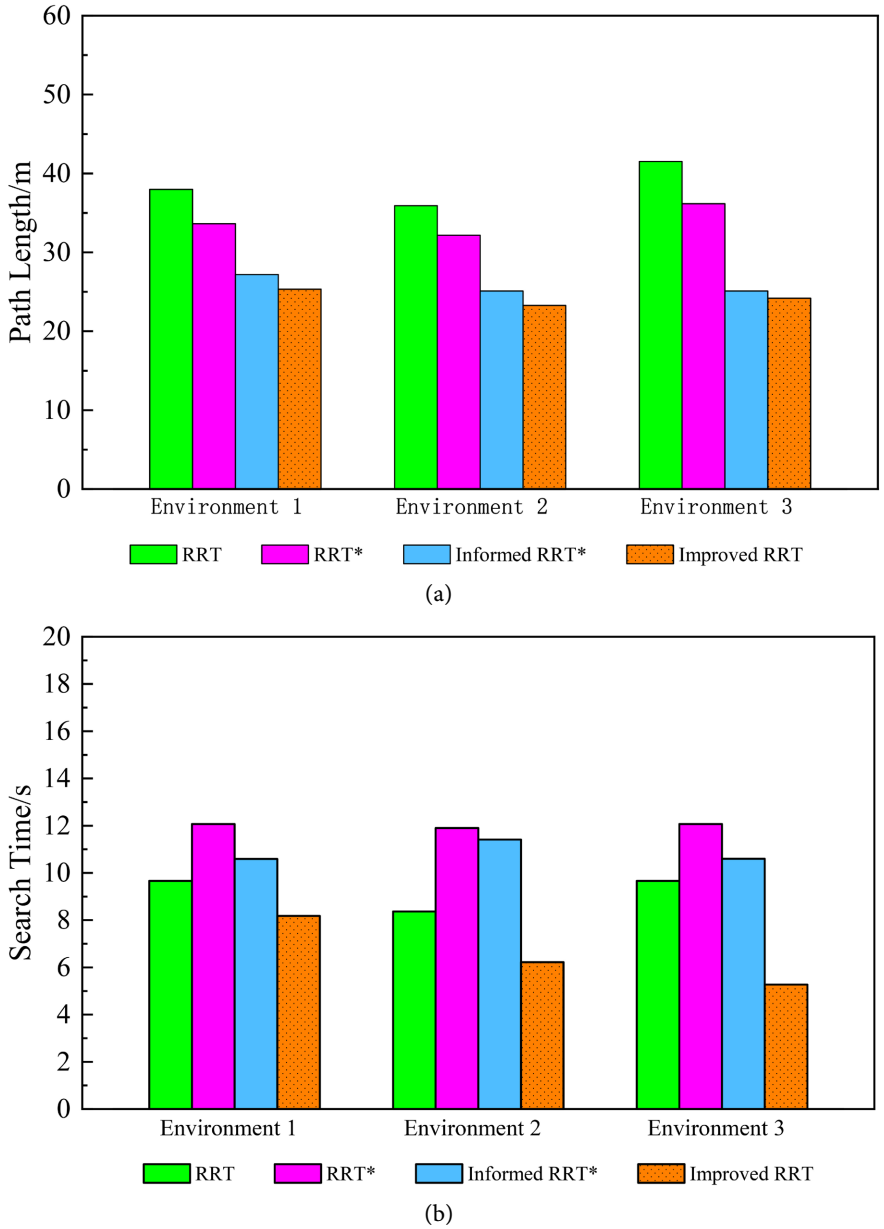


Figure 9. (a) Path lengths of the four algorithms in different environments, (b) Search time of the four algorithms in different environments.

The initial point is marked in red with coordinates (−5.94, 6.46, 1.1), while the target point is indicated in green, located at (5.81, −7.02, −2.4). The maximum number of sampling nodes is set to N = 5000, with a search radius of 2 and a step size of 1. Each algorithm is subjected to 50 path calculations to determine their path lengths, search times, and success rates, with the averages being calculated. The path taken by one instance of the algorithm run is shown in **Figure 11**. The recorded data are presented in **Table 6**.

Integrating the data from **Table 6** with **Figure 12**, the enhanced RRT algorithm demonstrates superior performance in the three-dimensional simulation environment regarding path length, search time, and success rate. The algorithm

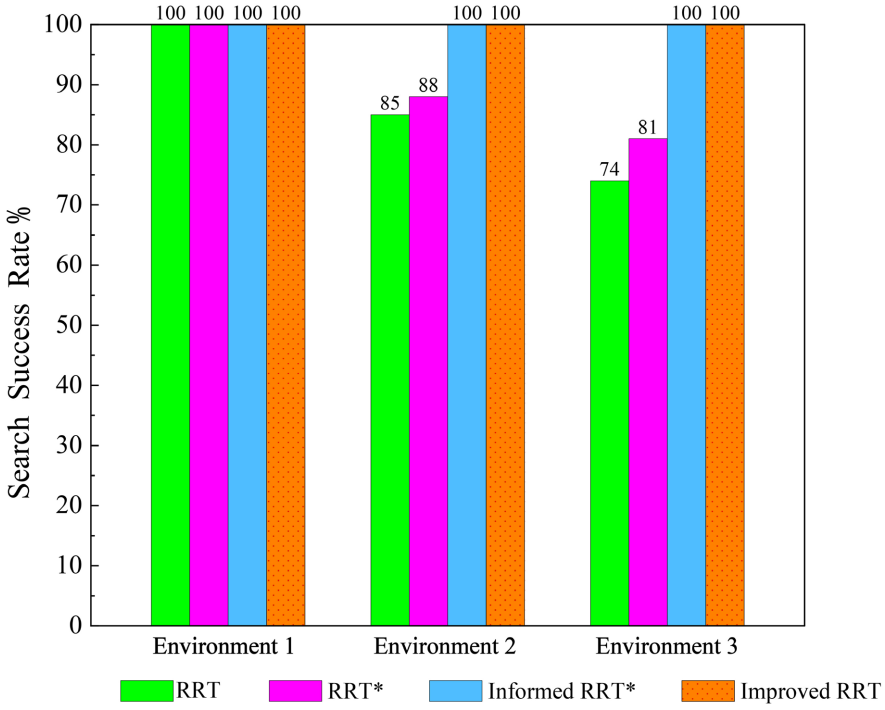


Figure 10. Success rates of four algorithms in different environments.

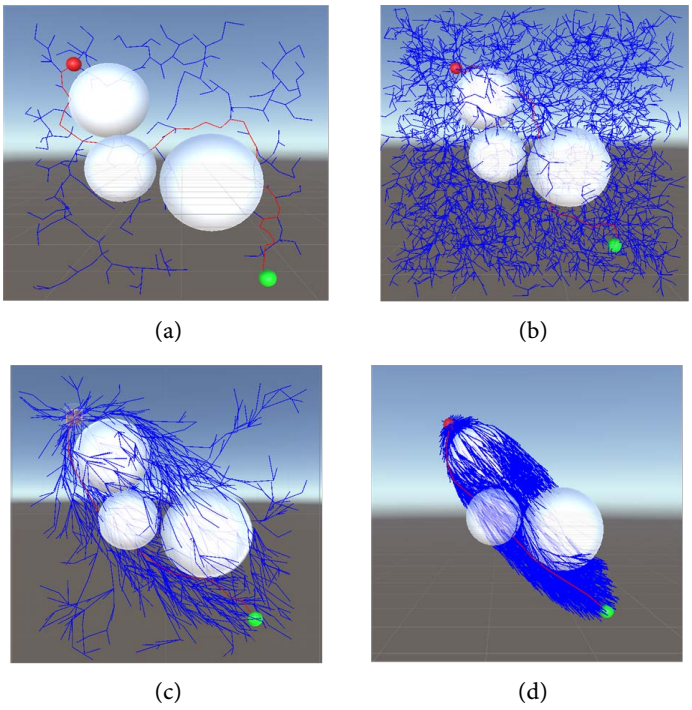
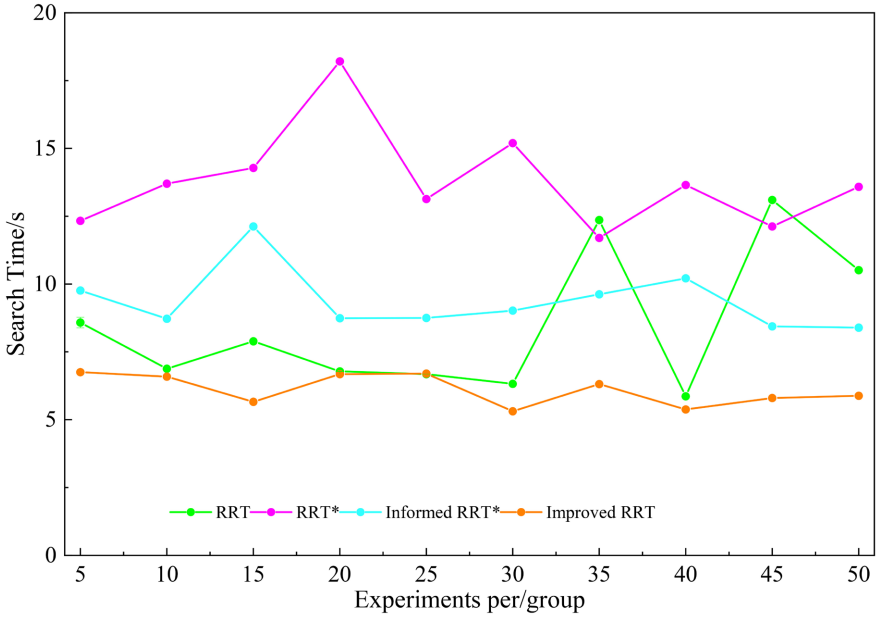


Figure 11. Path planning in three-dimensional environment: (a) RRT algorithm, (b) RRT* algorithm, (c) Informed RRT* algorithm, (d) Improved RRT algorithm.

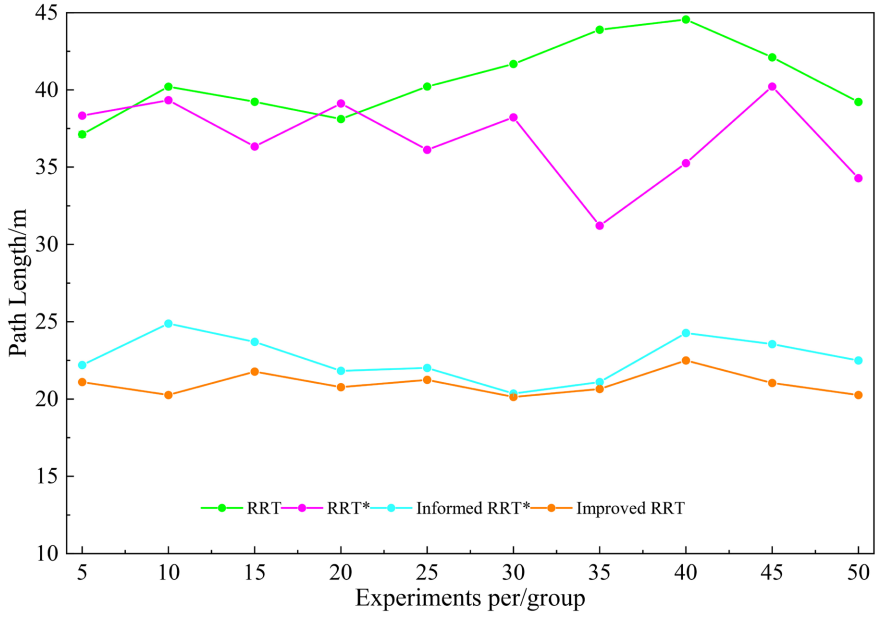
achieves a reduction in path length by 52.28%, 47.56%, and 11.14%; and a reduction in search time by 27.72%, 55.65%, and 34.75% compared to its counterparts. The RRT and RRT* algorithms do not achieve a 100% success rate, falling short

Table 6. Comparative performance of algorithms in a three-dimensional environment.

Algorithms	Step Size	Path Length/m	Search Time/s	Search Success Rate
RRT	1	42.11	8.49	90%
RRT*	1	38.33	13.80	91%
Informed RRT*	1	22.62	9.38	100%
Improved RRT	1	20.10	6.12	100%



(a)



(b)

Figure 12. (a) Search time of four algorithms in a three-dimensional environment, (b) Path length of four algorithms in a three-dimensional environment.

of the requirements for three-dimensional or higher-dimensional search needs. While the Informed RRT* algorithm does reach a 100% success rate, its path length and search time are not as advantageous when compared to the improved algorithm.

5. Conclusion

This chapter addresses issues such as node redundancy, poor path quality, and decreased success rates in complex environments encountered by the RRT, RRT*, and Informed RRT* algorithms during the sampling process, by proposing an improved RRT path planning algorithm. It begins with a description of the basic principles of the RRT, RRT*, and Informed RRT* algorithms, followed by a detailed presentation of the strategies employed by the improved RRT algorithm, emphasizing adaptive pruning optimization and dynamic elliptical search strategies. Particularly, building on the Informed RRT* algorithm, it introduces a synchronized pruning of the dynamic elliptical search tree to eliminate branches with costs higher than the current optimal path, effectively reducing the search space and enhancing search efficiency. This targeted path optimization enables the improved RRT algorithm to outperform the Informed RRT* algorithm in speed and reliability, especially in complex environments. To verify the practical effects of the improved algorithm, 50 sets of experiments in two and three-dimensional spaces were conducted. The results in two-dimensional environments show significant improvements in path length and search time, with path lengths reduced by an average of 35.77%, 28.48%, and 7.77% compared to the other three algorithms, and search times decreased by 28.82%, 45.43%, and 39.51%, respectively.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Karaman, S., Walter, M.R., Perez, A., *et al.* (2011) Anytime Motion Planning Using the RRT. 2011 *IEEE International Conference on Robotics and Automation*, Shanghai, 9-13 May 2011, 1478-1483. https://doi.org/10.1109/ICRA.2011.5980479

[2] Qureshi, A.H. and Ayaz, Y. (2015) Intelligent Bidirectional Rapidly-Exploring Random Trees for Optimal Motion Planning in Complex Cluttered Environments. *Robotics and Autonomous Systems*, **68**, 1-11. https://doi.org/10.1016/j.robot.2015.02.007

[3] Kuffner, J. and La Valle, S.M. (2000) 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.*00*CH*37065), San Francisco, 24-28 April 2000, 995-1001. https://doi.org/10.1109/ROBOT.2000.844730

[4] Kuwata, Y., Teo, J., Fiore, G., *et al.* (2009) Real-Time Motion Planning with Applications to Autonomous Urban Driving. *IEEE Transactions on Control Systems*

Technology, **17**, 1105-1118. https://doi.org/10.1109/TCST.2008.2012116

[5] Karaman, S. and Frazzoli, E. (2011) Sampling-Based Algorithms for Optimal Motion Planning. *The International Journal of Robotics Research*, **30**, 846-894. https://doi.org/10.1177/0278364911406761

[6] Gammell, J.D., Srinivasa, S. and Barfoot, T.D. (2014) Informed RRT: Optimal Sampling-Based Path Planning Focused via Direct Sampling of an Admissible Ellipsoidal Heuristic. 2014 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, 14-18 September 2014, 2997-3004. https://doi.org/10.1109/IROS.2014.6942976

[7] Islam, F., Nasir, J., Malik, U., *et al.* (2012) RRT*-Smart: Rapid Convergence Implementation of RRT* towards Optimal Solution. 2012 *IEEE International Conference on Mechatronics and Automation*, Chengdu, 5-8 August 2012, 1651-1656. https://doi.org/10.1109/ICMA.2012.6284384

[8] Armstrong, D. and Jonasson, A. (2021) AM-RRT: Informed Sampling-Based Planning with Assisting Metric. 2021 *IEEE International Conference on Robotics and Automation* (*ICRA*), Xi'an, 30 May-5 June 2021, 10093-10099. https://doi.org/10.1109/ICRA48506.2021.9561604

[9] Connell, D. and La, H.M. (2017) Dynamic Path Planning and Replanning for Mobile Robots Using RRT. 2017 *IEEE International Conference on Systems, Man, and Cybernetics* (*SMC*), Banff, 5-8 October 2017, 1429-1434. https://doi.org/10.1109/SMC.2017.8122814

[10] Otte, M. and Frazzoli, E. (2016) RRT^{X}: Asymptotically Optimal Single-Query Sampling-Based Motion Planning with Quick Replanning. *The International Journal of Robotics Research*, **35**, 797-822. https://doi.org/10.1177/0278364915594679

[11] Adiyatov, O. and Varol, H.A. (2017) A Novel RRT—Based Algorithm for Motion Planning in Dynamic Environments. 2017 *IEEE International Conference on Mechatronics and Automation* (*ICMA*), Takamatsu, 6-9 August 2017, 1416-1421. https://doi.org/10.1109/ICMA.2017.8016024

[12] Ge, J., Sun, F. and Liu, C. (2016) RRT-GD: An Efficient Rapidly-Exploring Random Tree Approach with Goal Directionality for Redundant Manipulator Path Planning. 2016 *IEEE International Conference on Robotics and Biomimetics* (*ROBIO*), Qingdao, 3-7 December 2016, 1983-1988. https://doi.org/10.1109/ROBIO.2016.7866620

[13] Yang, H., Li, L. and Gao, Z. (2017) Obstacle Avoidance Path Planning of Hybrid Harvesting Manipulator Based on Joint Configuration Space. *Transactions of the Chinese Society of Agricultural Engineering*, **33**, 55-62.

[14] Liu, C., Han, J. and An, K. (2017) Dynamic Path Planning Based on an Improved RRT Algorithm for RoboCup Robot. *ROBOT*, **39**, 8-15. https://doi.org/10.13973/j.cnki.robot.2017.0008

[15] Liao, B., Wan, F., Hua, Y., *et al.* (2021) F-RRT*: An Improved Path Planning Algorithm with Improved Initial Solution and Convergence Rate. *Expert Systems with Applications*, **184**, Article 115457. https://doi.org/10.1016/j.eswa.2021.115457

[16] Wu, Z., Meng, Z., Zhao, W., *et al.* (2021) Fast-RRT: A RRT-Based Optimal Path Finding Method. *Applied Sciences*, **11**, Article 11777. https://doi.org/10.3390/app112411777

[17] Wang, J., Li, B. and Meng, M.Q.H. (2021) Kinematic Constrained Bi-Directional RRT with Efficient Branch Pruning for Robot Path Planning. *Expert Systems with Applications*, **170**, Article 114541. https://doi.org/10.1016/j.eswa.2020.114541

[18] Zhang, Y., Wang, H., Yin, M., *et al.* (2023) Bi-AM-RRT*: A Fast and Efficient Sam-

pling-Based Motion Planning Algorithm in Dynamic Environments. *IEEE Transactions on Intelligent Vehicles*, **9**, 1282-1293. https://doi.org/10.1109/TIV.2023.3307283

[19] Kashyap, A.K. and Parhi, D.R. (2022) Implementation of Intelligent Navigational Techniques for Inter-Collision Avoidance of Multiple Humanoid Robots in Complex Environment. *Applied Soft Computing*, **124**, Article 109001. https://doi.org/10.1016/j.asoc.2022.109001

[20] Kashyap, A.K. and Parhi, D.R. (2022) Stable Locomotion of Humanoid Robots on Uneven Terrain Employing Enhanced DAYANI Arc Contour Intelligent Algorithm. *Journal of Autonomous Vehicles and Systems*, **2**, Article 041002. https://doi.org/10.1115/1.4063055

[21] Kashyap, A.K. and Parhi, D.R. (2023) Dynamic Walking of Multi-Humanoid Robots Using BFGS Quasi-Newton Method Aided Artificial Potential Field Approach for Uneven Terrain. *Soft Computing*, **27**, 5893-5910. https://doi.org/10.1007/s00500-022-07606-7

[22] Zhang, H., Lin, W. and Chen, A. (2018) Path Planning for the Mobile Robot: A Review. *Symmetry*, **10**, Article 450. https://doi.org/10.3390/sym10100450