A motion planning method for winter jujube harvesting robotic arm based on optimized Informed-RRT∗ algorithm

Anxiang Huang , Chenhao Yu , Junzhe Feng , Xing Tong , Ayanori Yorozu , Akihisa Ohya , Yaohua Hu

PII: S2772-3755(24)00336-8
DOI: https://doi.org/10.1016/j.atech.2024.100732
Reference: ATECH 100732

To appear in: *Smart Agricultural Technology*

Received date: 12 October 2024
Revised date: 12 December 2024
Accepted date: 16 December 2024

Please cite this article as: Anxiang Huang , Chenhao Yu , Junzhe Feng , Xing Tong , Ayanori Yorozu , Akihisa Ohya , Yaohua Hu , A motion planning method for winter jujube harvesting robotic arm based on optimized Informed-RRT∗ algorithm, *Smart Agricultural Technology* (2024), doi: https://doi.org/10.1016/j.atech.2024.100732

Highlights

- Optimized Informed-RRT* algorithm reduces path length and planning time in 2D space.
- The algorithm not only has high efficient performance but also reduces nodes in 3D space.
- The algorithm maintains stability and smooth movement in dynamic environments.

A motion planning method for winter jujube harvesting robotic arm based on optimized Informed-RRT* algorithm

Anxiang Huang [a], Chenhao Yu [a], Junzhe Feng [a], Xing Tong [a], Ayanori Yorozu [b], Akihisa Ohya [b], Yaohua Hu [a, b, c, *]

[a]*College of Optical, Mechanical and Electrical Engineering, Zhejiang A&F University, Hangzhou 311300, China*

[b]*Institute of Systems and Information Engineering, University of Tsukuba, Tsukuba 305-8573, Japan*

[c]*Key Laboratory of Agricultural Equipment for Hilly and Mountainous Areas in Southeastern China(Co-construction by Ministry and Province), Ministry of Agriculture and Rural Affairs, Hangzhou 311300, China*

*****Corresponding author: Yaohua Hu (Email: huyaohua@zafu.edu.cn)*

Abstract

Winter jujube is a fruit that is rich in nutritional value and has a delicious taste. Since the ripe winter jujubes are tender and easy to be damaged, during the picking process the mechanical arm needs to maintain smooth and stable movement to avoid the impact of vibration or oscillation on the picking task. However, existing motion planning algorithms may not guarantee the smoothness and stability of the mechanical arm's movement. The research discovered a motion planning method based on the optimized Informed-RRT* algorithm. By adding target bias, adaptive step size, and pruning strategies, the optimization of search paths was achieved, reducing unnecessary movement of the mechanical arm during operation. This can ensure high picking success rate and low damage rate. Through comparative experiments, the optimized Informed-RRT* algorithm has good performance in the two-dimensional and three-dimensional spaces. Within the specified time, the planned paths by the optimized Informed-RRT* algorithm are shorter and smoother, significantly improving the efficiency of the mechanical arm. Additionally, this research deploys the optimized Informed-RRT* algorithm to the Robot Operating System (ROS) and conducts three-dimensional modeling of the picking environment through Moveit! to obtain real-time environmental information and obstacle detection. This allows for

effective avoidance of obstacles while ensuring the optimal path. To ensure the safety of the mechanical arm's movement, this research monitors the position changes of each joint in real-time. The results indicated that during the movement process, the angular velocity and angular acceleration of the mechanical arm exhibit smooth and continuous trends, demonstrating good dynamic stability and control performance during movement and further proving the effectiveness of the optimized Informed-RRT* algorithm.

Keywords: Winter jujube; Motion planning; optimized Informed-RRT* algorithm; Performance; Robot Operating System (ROS)

1. Introduction

Winter jujube, also known as fresh jujube, is a highly nutritious fruit. Not only does it have a sweet flavour, but it is also rich in vitamins, minerals, and other health-promoting ingredients. The planting area of jujube trees and the production of jujube trees both occupy more than 95% of the global share, and the picking of jujube trees is labor-intensive work, and the picking cycle is relatively long(Yu et al., 2023). With the rising cost of labor and the increasing shortage of rural labor, the traditional manual picking method is not only inefficient but also costly and has gradually failed to meet the needs of the winter jujube industry (Yu et al., 2024).

Although the existing winter jujube picking machines provide the convenience of automated picking, they still have a lot of room for improvement in terms of the precision of identifying fruits, and the fineness of operation is not enough to avoid damage to fruits or branches. Therefore, the introduction of automated or intelligent picking technology is an effective way to improve picking efficiency (Guo et al., 2024). Robotic arm picking is an efficient intelligent picking method, and the robotic arm motion trajectory planning algorithm is one of the strategies to achieve efficient and low-loss picking(Ye et al., 2023).

Currently, robotic arm motion trajectory control algorithms are broadly classified into traditional methods (Dai et al.,

2022), graph search methods (Thakur et al., 2023), random sampling methods (Etikan and Bala, 2017), and so on. The traditional method is represented by the artificial potential field method as a representative algorithm, which was proposed by Khatib in 1986(Khatib, 1986). However, the method has some limitations, such as the problem of unreachable targets that may result when there are obstacles near the target point. To address this phenomenon, researchers have proposed many improvement strategies. Alshammrei et al. proposed an adaptive artificial potential field for free-flying space robots, addressing local minima and energy efficiency with optimized fields and escape forces, demonstrating improved obstacle avoidance and energy-saving performance (Alshammrei et al., 2022). Zhong ; lembono et al (Zhong et al., 2021; Lembono et al., 2021) For high-dimensional spatial motion planning for multi-degree-of-freedom manipulators, sampling-based methods have been widely used. Among them, the fast exploratory random tree is the most widely used method. Kavraki and Latombe proposed fast exploratory random tree does not need to map obstacles from the task space to the configuration space, and it has the characteristics of fast exploration and complete probability, which is suitable for path planning in high-dimensional space (Kavraki and Latombe, 1994). Although graph search and random sampling methods, despite being able to handle planning problems in high-dimensional spaces, have high computational complexity and may suffer from inefficiencies when dealing with dynamic environments.

Graph theoretic methods such as the A* algorithm and Dijkstra's algorithm (Magzhan and Jani, 2013) have the advantages of global optimality, flexibility, and predictability in motion planning compared to artificial potential field methods. Gu et al. develops an improved Dijkstra algorithm for collision-free mobile robot navigation. By modeling the environment as a digraph, the algorithm calculates the shortest path offline. Using ultrasonic sensors to detect obstacles, the robot updates the diagraph and recalculates paths dynamically, ensuring efficient and practical navigation in obstacle-rich environments (Gu et al., 2024). Ji et al (Ji et al., 2014) used the improved Dijkstra algorithm to find the initial walking path for an apple picking robot to find the initial walking path, and used the improved ant colony algorithm to optimize

the initial path. The experimental results show that the proposed method is simple and easy to implement, and the robot manipulator can avoid tree branches and successfully pick apples in a short time. Although graph theoretic algorithms such as A* and Dijkstra have the above advantages in motion planning (Kang et al., 2024), they also have some limitations, such as when dealing with large-scale maps or dynamic environments, the amount of computation may be larger, and real-time performance will be affected.

In 3D space, sampling-based motion planning algorithms such as Probabilistic Road Methods (PRM) (Kavraki et al., 1996) and Rapidly-exploring Random Tree (RRT) (LaValle and Kuffner, 2001) can effectively avoid gradient explosion and other problems. Pohan et al (Pohan et al., 2021) proposed a path planning algorithm using a hybrid algorithm of RRT and Ant Colony System (ACS), through which the optimal path can be generated from the previous suboptimal path information, which can reach the optimal value with a faster convergence rate. To improve the overall search speed of the algorithm, Kim et al. proposed the RRT*-connect algorithm to significantly improve the search efficiency by expanding two trees in both directions (Kim et al., 2019). Zhang et al. and Huh et al.(Zhang et al., 2020; Huh et al., 2018) proposed that in the path planning of redundant manipulators, not only the collision-free movement of the manipulator but also the manipulability of the manipulator during the movement need to be considered. An excellent configuration can avoid single or bad motion of the manipulator, thus improving the quality of the manipulator's motion. Alam et al.(Alam et al., 2023) proposed an energy-efficient industrial robot motion planning method (FC-RRT*), which reduces the energy consumed by the robot joints by 1.6% to 16.5% compared to the traditional RRT* algorithm by generating nodes in a predetermined direction and then calculating the energy consumption using the round point method. Ichnowski et al.(Ichnowski et al., 2020) proposed a multi-core serverless Lambda computational motion planning algorithm that exploits multi-core parallelism and parallel computing power to solve complex problems efficiently and quickly. Wang et al.(Wang et al., 2022) proposed a novel algorithm based on Circular Arc Filleting (CAF-RRT*), which reduces the cost

of initial planning paths by means of a path optimization strategy based on the triangular rule, and the proposed algorithm reduces the length of paths by about 2% compared to other RRT algorithms, which ensures the smoothness and security of the paths. Lacevic and Osmankovic (Lacevic and Osmankovic, 2020) proposed a path-planning method based on a geometric structure called generalized BUR, which is able to quickly capture a wide range of free C-space, thus accelerating exploration. Chen et al. (Chen et al., 2019) in the Gm-RRT algorithm, adopted the method of generating random points to improve the computational efficiency of path planning and introduced the adaptive target virtual gravity method to ensure the balance between the target position and the surrounding space. The above algorithms improve the efficiency in dealing with the multi-degree-of-freedom robotic arm planning problem, but they require more complex computational resources and algorithmic optimization to adapt to dynamically changing environments and to ensure real-time algorithmic performance.

The winter jujubes are small target fruits, and the objective of the research is to design an effective algorithm that enables the robotic arm to complete the specified task in an optimal path planning manner, i would suggest a research hypothesis:

(1) To improve Informed-RRT* algorithm to search the shorter distance and less time;

(2) To search high-efficiency Informed-RRT* algorithm to simplify the paths;

(3) To deploy optimized Informed-RRT* algorithm into ROS to validate the effectiveness of the algorithm to improve the quality and efficiency of the operation of the robotic arm in picking winter jujube.

2. Materials and methods

In this study, the hardware system construction of the identification and localisation of winter jujube and picking robot was carried out in the preliminary stage (Yu et al., 2024b), and this book firstly describes the groundwork done in the preliminary stage and then describes in detail several strategies to improve the Informed-RRT* algorithm. These

improvements are based on the original Informed-RRT* algorithm, with the addition of a goal bias method to increase the algorithm's goal orientation during the search process, thus improving the efficiency of path planning; and an adaptive step-size method to improve the flexibility and adaptability of the algorithm. A pruning method is added to efficiently prune unnecessary paths during the search process and optimize the path quality.

2.1.Materials

This study focuses on images of winter jujube trees in a complex field environment, and the sample images were collected at the winter jujube experimental demonstration station of Northwest A&F University. The jujube trees at the experimental station have a plant spacing of 4 metres, a row spacing of 2 metres, and an average crown diameter of 2 metres.

An Intel RealSense D435i camera was used in the study to capture images in the jujube orchard. The distance between the camera lens and the jujubes was kept at 50-100 cm during image acquisition to ensure that the background was close to the mechanical picking environment. To increase the diversity of sample images, the study was conducted under six different environmental conditions including bright light, dark light, single target, multi-target, branch and leaf occlusion, and fruit damage. A total of 1134 images of jujube were collected, To standardize the input for the image recognition and path planning system, and all samples were resized RGB images with a resolution of 640 × 480. images of the jujube fruit under different environments were shown in Fig. 1.

（a）

（b）

（c）

（d）

（e）

（f）

Fig.1. Winter jujubes in different situations. （a）bright environment（b）darkness environment（c）single target（d）multi-target（e）fruit obscured（f）fruit damage

The physical object of the jujube picking system is shown in Fig. 2, including the Realsense D435i depth camera, the Aubo C5 six-degree-of-freedom robotic arm (manufactured by Zhejiang Aubo Robotics Co., Ltd., China, with a maximum payload of 5 kg, a self-weight of 24 kg, a repeat positioning accuracy of ±0.1 mm, and a working radius of 886.5 mm), the end-effector, the control box, and the computer.

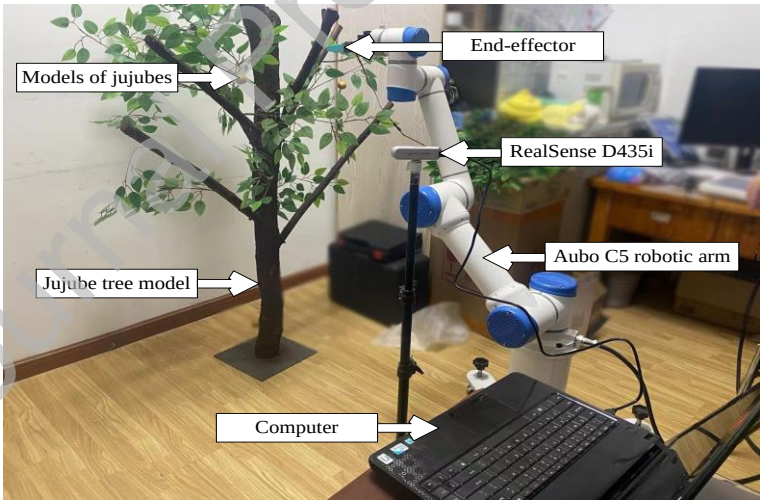


Fig.2. Physical drawing of the hardware of the picking robot

*2.2 Informed-RRT**

The RRT* algorithm samples the free space uniformly, leading to the generation of numerous redundant branches in the search tree and requiring significant time to find the globally optimal path. To accelerate convergence and obtain the optimal path, Gammell et al. proposed the Informed-RRT* algorithm (Gammell et al., 2014). This algorithm uses the

RRT* search tree to find the initial bootstrap path and then generates an elliptical sampling region defined by the start point, goal point, and path length, effectively accelerating convergence to the optimal path.

The Informed-RRT* algorithm plans the first path using the same sampling method as RRT*. Once the first path is planned, the algorithm constructs an elliptical region based on the path length and space. This elliptical region becomes the new sampling space, where the algorithm randomly samples to find a better path solution.

The standard elliptic equation is shown in Equation 1:

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

Where: a is the length of the long axis, b is the length of the short axis, the focus coordinates are (±c,0), and $a^2+b^2=c^2$. In the Informed-RRT* algorithm, the start point x_{init} and the target point x_{goal} are used as ellipse foci， Let a be half of the initial length of the path C_{best}, that is, $a=\frac{C_{best}}{2}$, and also let $c=\frac{C_{min}}{2}$, that is, $b=\frac{\sqrt{C_{best}^2+C_{min}^2}}{2}$. The elliptical sampling region is shown in Fig. 3.

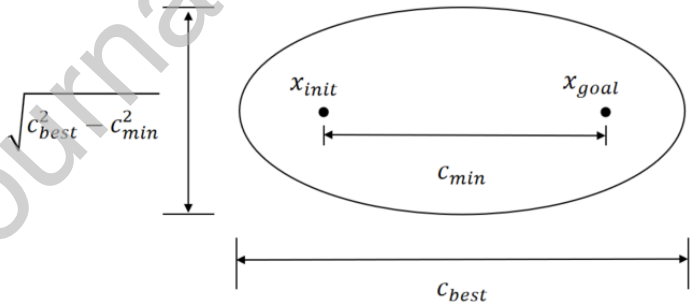


Fig.3. Informed-RRT* algorithm elliptic modelling

In each subsequent iteration, if a shorter path is found, that shortest path is used as the new optimal path length C_{best} and the sampling ellipsoid is updated. As the number of cycles of the algorithm increases, the length of the optimal path decreases, the long axis of the ellipsoid decreases, and the sampling space decreases. As a result, it can converge to the optimal solution faster, which greatly improves the search efficiency.

*2.3 optimized Methods for Informed-RRT**

2.3.1 Target biasing methods for optimizing time and space efficiency

Informed-RRT* is improved in terms of sampling space compared to the RRT* algorithm, but the search for paths in elliptic space is similar to RRT*. In this study, by adding the goal bias method, its purpose is to guide the search process toward a better quality solution, reduce the search space and search time, and accelerate the convergence to the optimal solution (Zhang et al., 2024). In addition, adding goal bias can also help the algorithm jump out of the local optimal solution and avoid falling into the local optimal solution problem, thus increasing the robustness and reliability of path planning.

First, set a target bias probability p（$0 \le p \le 1$）. Then, a random number i ranging from 0-1 is generated. determine the magnitude of the random value i with respect to the target bias probability p. If $i \ge p$, a random sample point is used as x_{rand}, and conversely, if $i < p$, the target point x_{goal} is used as x_{rand} to guide the expansion of the random tree to be able to grow in the direction of the planning endpoint. The sampling point generation expression is shown in Equation 2,3. The target bias principle is shown in Fig. 4.

$$x_{rand} = \begin{cases} x_{goal} & i<p \\ x_{random\_sample} & i \ge p \end{cases} \tag{2}$$

$$x_{random\_sample} = min(|x_{rand1} - x_{goal}|, |x_{rand2} - x_{goal}|, 0|x_{rand3} - x_{goal}|, \ldots, |x_{randn} - x_{goal}|). \tag{3}$$

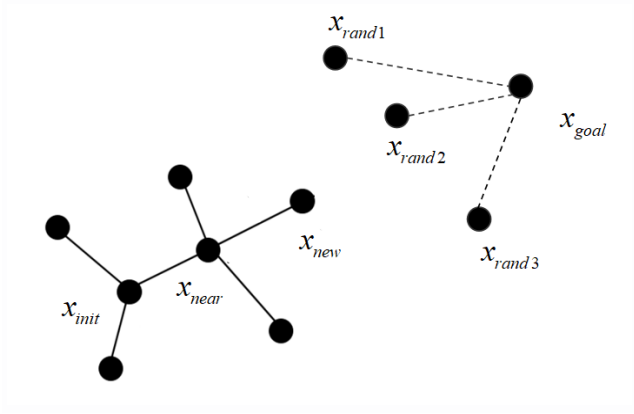


Fig.4. Target Bias Schematic

The simplified pseudo-code for the target bias function was shown in Fig.5:

```
Sample(goal):
1.  if random() < goal_bias:
2.      return x_goal
3.  else:
4.      return x_random_sample()
```

Fig.5. Simplified pseudo-code for the target bias function

2.3.2 Adaptive step-size methods for optimising search efficiency

In the Informed-RRT* algorithm, if the set expansion step size is too small, the growth of the search tree may be significantly slowed down, reducing the algorithm's efficiency. Conversely, if the step size is too large, it may be difficult to grow new nodes in maps with many obstacles, which also significantly reduces the algorithm's efficiency (Aleti and Moser, 2016). To address the impact of such extreme situations on the algorithm's efficiency, this paper introduces an adaptive step size approach.

The adaptive step size method designed in this paper sets the straight line distance between x_{near} and x_{new} as the initial step size l_{step_size}, if the connecting line between x_{new} and x_{near} does not encounter obstacles in the process of expanding, then it is considered that there is no risk of encountering obstacles under the current step size, and the step size of the next step is increased to further increase the expansion speed; if the connecting line encounters obstacles, then it is If the connecting line encounters obstacles, the obstacles was considered to have an impact on the subsequent path planning, and the step length of the next step is reduced to avoid invalid sampling. The step length expansion expression is shown in Equation 4. The principle of adaptive step size is shown in Fig.6.

$$l_{new\_step\_size}\begin{cases} (1+i)l_{step\_size}\ , & x_{new}\ outside\ the\ obstacles \\ (1-i)l_{step\_size}\ , & x_{new}\ In\ the\ obstacles \end{cases} \tag{4}$$

Where i is a randomly generated rational number between 0 and 1.

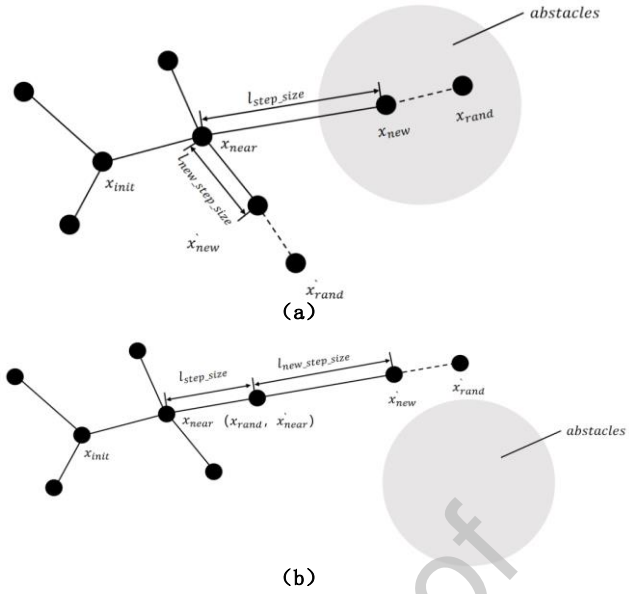


Fig.6. Adaptive Step Schematic（a）x_{new} Adaptive Expansion within Obstacles （b）x_{new} Adaptive expansion outside of obstacles

The simplified pseudo-code of the adaptive step function was shown in Fig. 7:

```
UpdateStepSize(x_near,x_new):
1.  l_step_size = distance(x_near, x_new):
2.  if ObstacleFree(x_near, x_new):
3.      l_new_node.step_size = small_step_size
4.  else:
5.      l_new_node.step_size = large_step_size
```

Fig.7. Simplified pseudo-code for the adaptive step function

2.3.3 Pruning methods for simplified paths

The Informed-RRT* algorithm generates numerous nodes in each iteration and adds them to the search tree. nodes are critical points in the search tree where choices are made, and each node can potentially lead to multiple paths. As the number of iterations increases, the search tree expands and accumulates many redundant nodes and edges. These redundant elements increase the time complexity of the path search and may lead to poor-quality generated paths (Lozano

and Medaglia, 2013). To address this issue, we introduce a pruning method to effectively reduce the size of the search tree by deleting or merging some nodes and edges, thereby improving search efficiency.

In the pruning method designed in this paper, all the nodes was put into the set D in order, with the starting point x_{init} as the root node, and the nodes was connected sequentially in order. If the connecting line does not collide with obstacles, then continue to detect whether the connecting line between the next node and the root node collides, if the connecting line collides with obstacles, then set the previous node as the root node, and continue to repeat the above steps until the root node is connected to the target point x_{goal}. The principle of pruning method is shown in Fig. 8.

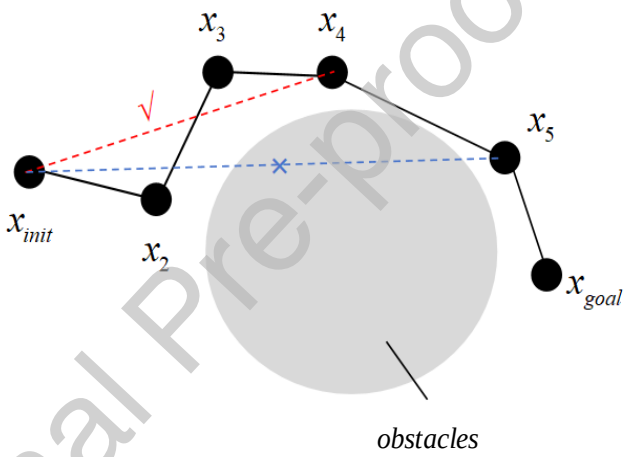


Fig.8. Pruning Strategy Schematic

The simplified pseudo-code for the pruning function was shown in Fig.9.

```
Prune(tree, x_new):
1.  x_near = Near(tree, x_new)
2.  for each x_node in x_near:
3.      if CostToGo(xnode) + CostBetween(node, x_new) < CostToGo(x_new):
4.          tree.RemoveEdge(x_node, x_father)
5.          tree.AddEdge(x_node, x_new)
6.          x_father = x_node
```

Fig.9. Simplified Pseudo-Code for Pruning Functions

3. Result and discussion

3.1 Simulation results of Informed-RRT algorithm*

To verify the effectiveness of the optimized Informed-RRT* algorithm, this study constructs 2D and 3D simulation environments of 500 cm x 500 cm and 500 cm x 500 cm x 500 cm in Matlab 2021b, respectively, as shown in Fig. 10(a) and Fig. 11(a). In the 2D space, the red point at (1, 1) is the start point, the green point at (490, 490) was the target point, and the black rectangles and circles was obstacles. In the 3D space, the red point at (1, 1, 1) was the start point, the green point at (490, 490, 490) was the target point, and the spheres was obstacles. This paper compares the path search effectiveness of the RRT, RRT Connect, RRT*, Informed-RRT*, and optimized Informed-RRT* algorithms in both two-dimensional and three-dimensional spaces. The simulation results were shown in Fig. 10(b-f) and Fig. 11(b-f), where the blue path indicates the search path of the algorithm.

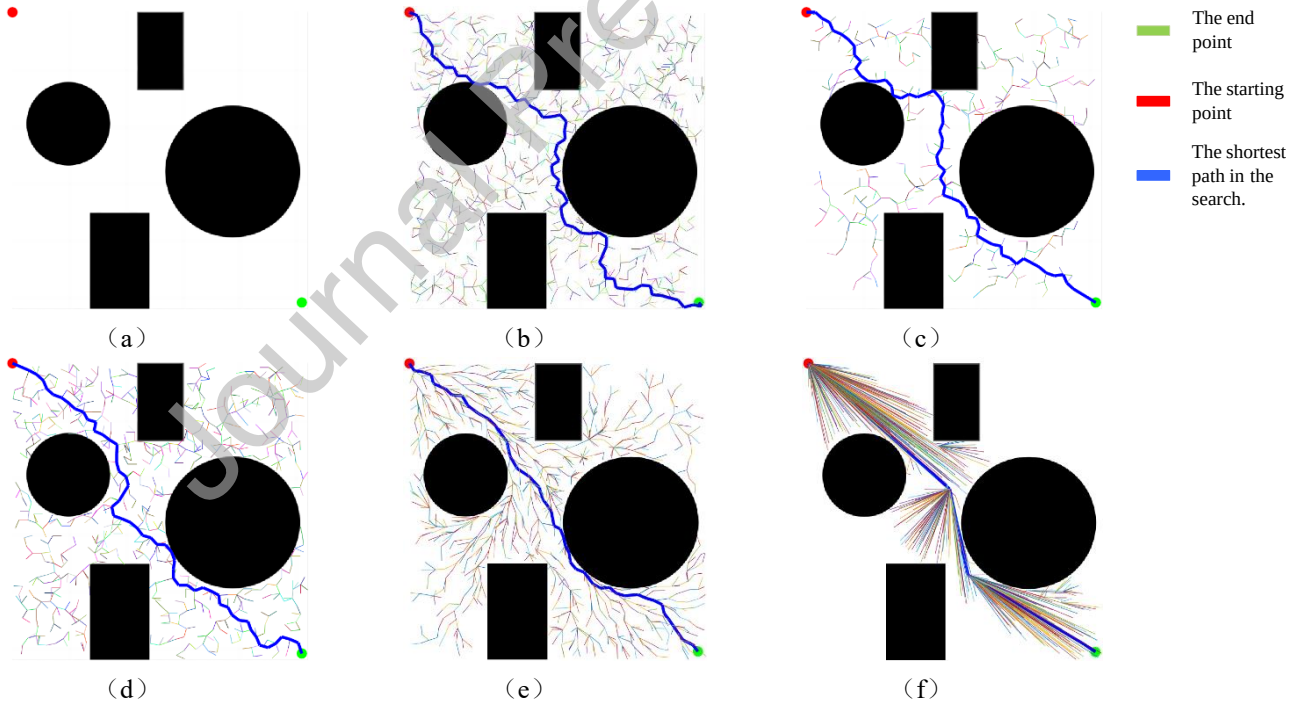


Fig.10. 2D simulation environment and path planning effect of each algorithm in 2D space（a）2D space（b）RRT algorithm（c）RRT Connect Algorithm（d）RRT* algorithm（e）Informed-RRT* algorithm（f）Optimized Informed-RRT* algorithm

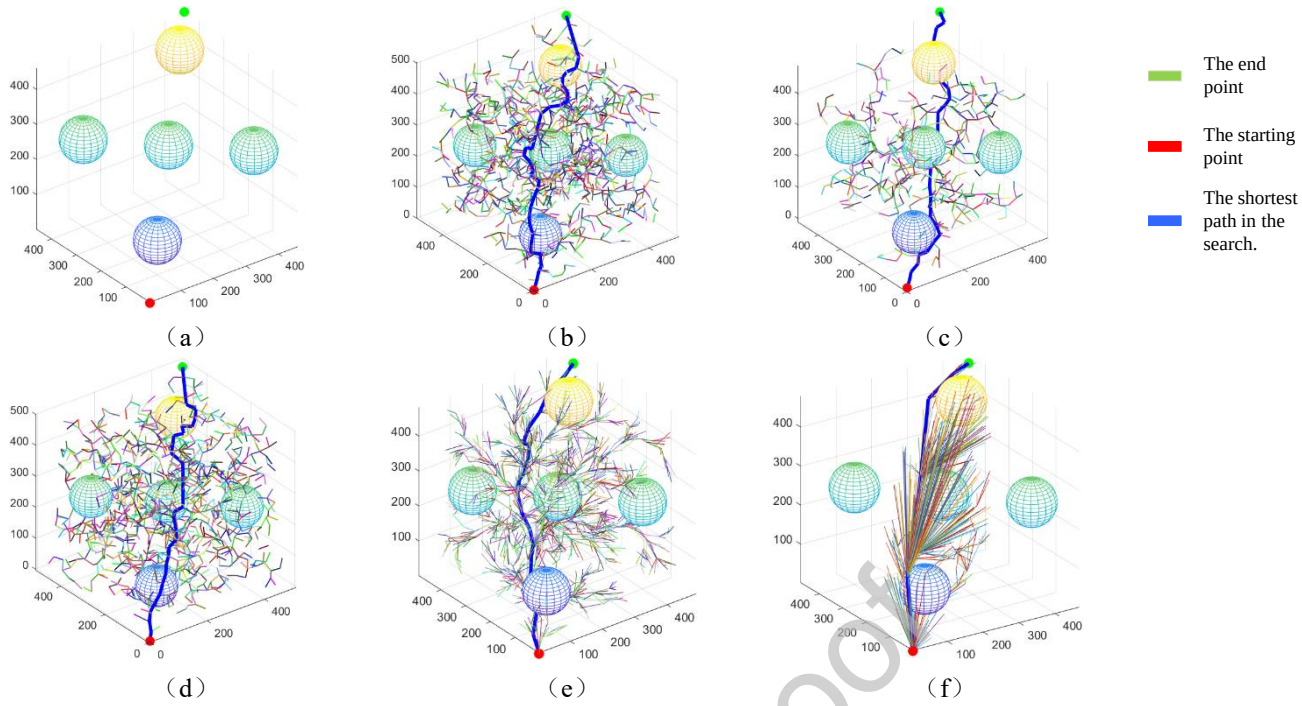


Fig.11. 3D simulation environment and path planning effect of each algorithm in 3D space（a）3D space（b）RRT algorithm（c）RRT Connect Algorithm（d）RRT* algorithm（e）Informed-RRT* algorithm（f）Optimized Informed-RRT* algorithm

To minimize the impact of algorithmic randomness on path planning results, each algorithm was repeated 20 times, with a maximum of 2000 iterations per trial. The success rate was defined as the probability of successfully reaching the target point from the starting point across 20 repetitions. The experimental data were presented in Table 1.

Table 1. Comparison of simulation results of each algorithm in 2D and 3D space

Simulation environment	Algorithm	Average path planning time (s)	Average sampling node	Average path length (m)	Success rate (%)
2D space	RRT	3.21	930	11.03	100.00
	RRT Connect	1.76	412	10.34	100.00
	RRT*	2.62	852	9.68	100.00
	Informed-RRT*	5.91	1,086	8.37	100.00
	Optimized Informed-RRT*	5.03	458	7.23	100.00
3D space	RRT	7.82	3,369	9.56	80.00
	RRT Connect	5.33	732	8.97	90.00
	RRT*	6.55	3,567	8.72	95.00
	Informed-RRT*	9.21	1,933	9.09	90.00
	Optimized Informed-RRT*	8.06	391	8.36	100.00

From the 2D space data in Table 1, it was evident that the optimized Informed-RRT* algorithm presented in this

paper has fewer average sampling points and the shortest average path length in two-dimensional space. As shown in Fig. 10, the optimized Informed-RRT* algorithm plans a smoother path. However, in terms of planning time, the RRT Connect algorithm has the shortest average planning time, while the average planning time for the optimized Informed-RRT* algorithm was 5.03 seconds. This was because the optimized Informed-RRT* algorithm does not exit the loop after the initial path planning but continues to optimize the search route until the maximum number of iterations is reached. In contrast, the RRT, RRT Connect, and RRT* algorithms stop planning and exit the loop after finding the route. As a result, the optimized Informed-RRT* algorithm has a longer planning time compared to the RRT, RRT Connect, and RRT* algorithms. To address this in practical deployments, the maximum planning time can be set to prevent the optimized Informed-RRT* algorithm from taking too long.

From the 3D space data in Table 1, it was evident that the optimized Informed-RRT* algorithm has the shortest average planning path and significantly fewer average sampling nodes compared to other algorithms. Notably, the average number of nodes was reduced by 88.39% compared to the RRT algorithm. As shown in Fig. 11, the optimized Informed-RRT* algorithm also plans the smoothest search paths. This improvement was attributed to the introduction of the target bias method, which directs the algorithm to search more toward the target point rather than exploring the space randomly. This targeted search approach makes it easier to find shorter paths.

Additionally, the adaptive step size method allows the algorithm to adjust the step size according to the environment during the search. In areas where the path was relatively flat, the step size can be increased to speed up the search, while in more complex or dense regions, the step size can be reduced to improve path accuracy. Combined with the pruning method, the algorithm further optimizes the path by removing unnecessary nodes, which may be local minima or branches that do not lead directly to the goal. This reduces the number of folds or detours in the path, resulting in a shorter and smoother path.

3.2 Obstacle avoidance path planning test of winter jujube picking robotic arm

3.2.1 Setting up the test environment

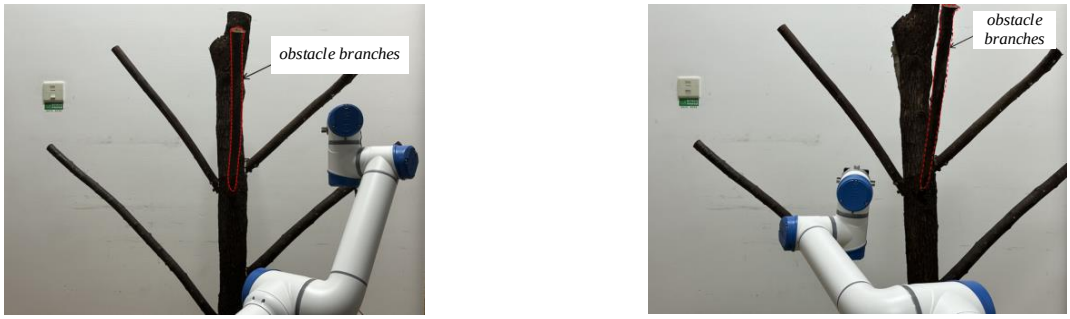
In this paper, an optimized Informed-RRT* algorithm was proposed and validated in a Matlab environment. The results demonstrate that the optimized Informed-RRT* algorithm significantly reduces the running time, the number of spatial nodes, and sampling points. To further verify the effectiveness of the optimized algorithm in a real picking environment, two scenarios were constructed for this experiment: a simple environment without leafy branches and a complex environment with leafy branches, as shown in Fig. 12. By comparing the obstacle avoidance and path planning performance of the robotic arm in these two different environments, the study aims to enhance the path planning and obstacle avoidance capabilities of the robotic arm in complex environments.

（a） （b）

Fig.12. Experimental scene（a）Simple environment（b）Complex environment

3.2.2 Simulation tests in a simple environment

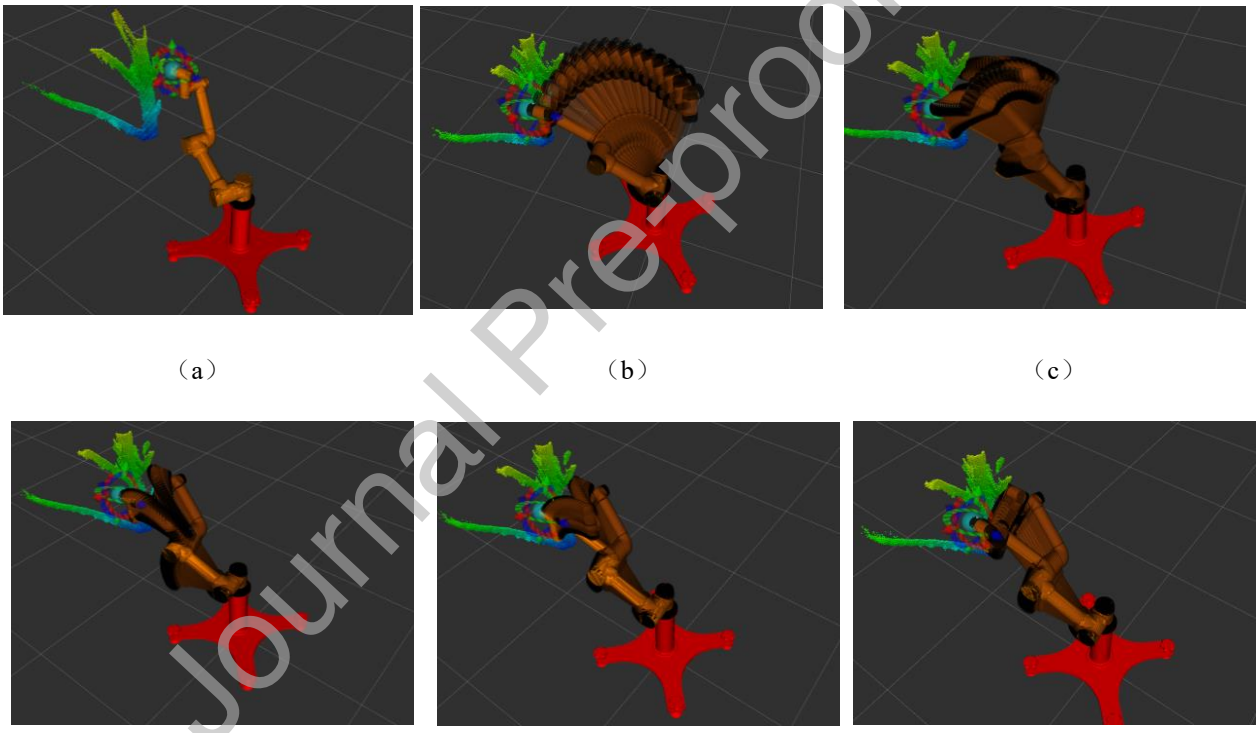
In the environment shown in Fig. 12(a), the motion planning of the robotic arm for jujube picking compares four mainstream path planning algorithms, as well as the optimized Informed-RRT* algorithm. The robotic arm needs to move from Fig. 13(a) to Fig. 13(b). There was a tree branch as an obstacle in the linear space between these two locations, which the robotic arm must avoid to reach the target location.



（a） （b）

Fig.13. Manipulator position in simple environments（a）Initial position（b）Target position

In this experiment, five algorithms were tested under the same initial conditions, where the maximum time for planning was set to 3 s. The simulation results were shown in Fig. 14.



（a） （b） （c）

（d） （e） （f）

Fig.14. Initial position of manipulator and motion planning of manipulator based on various algorithms in simple environments（a）Initial position（b）RRT（c）RRT*（d）RRT Connect（e）Informed-RRT*（f）Optimized Informed-RRT*

In a simple environment, by comparing the obstacle avoidance paths generated by the five algorithms, it can be seen that the Informed-RRT* series of algorithms was superior to the other three algorithms in that it generates shorter planning

paths and the movement of the robotic arm was more fluent. The reason for this was that the Informed-RRT* algorithm continuously optimizes the path during the search process, and by introducing the optimality guarantee mechanism, a path closer to the optimal solution can be obtained. To systematically study the characteristics of the five planning algorithms and avoid the chance of the algorithms, we carried out 20 repetitions of motion planning tests in a simple environment and recorded the results in Table 2.

Table 2. Comparison of simulation experiment results under simple obstacle environment

Programming algorithm	Average sampling points	Average planning time (s)	Success rate (%)
RRT	680	2.07	90.00
RRT Connect	565	1.37	95.00
RRT*	623	1.95	95.00
Informed-RRT*	739	1.83	100.00
Optimized Informed-RRT*	460	1.72	100.00

From Table 2, it can be seen that RRT, RRT*, RRT-Connect, Informed-RRT*, and optimized Informed-RRT* planning success rates are 90.00%, 95.00%, 95.00%, 100.00%, and 100.00% respectively. In terms of the average number of sampling points, the optimized Informed-RRT* algorithm has the least number of average sampling points and shorter planning time. In summary, in simple environments, the optimized Informed-RRT* algorithm of this study has a significant improvement in planning performance compared to some mainstream robotic arm path planning algorithms.

3.2.3 Simulation tests in complex environments

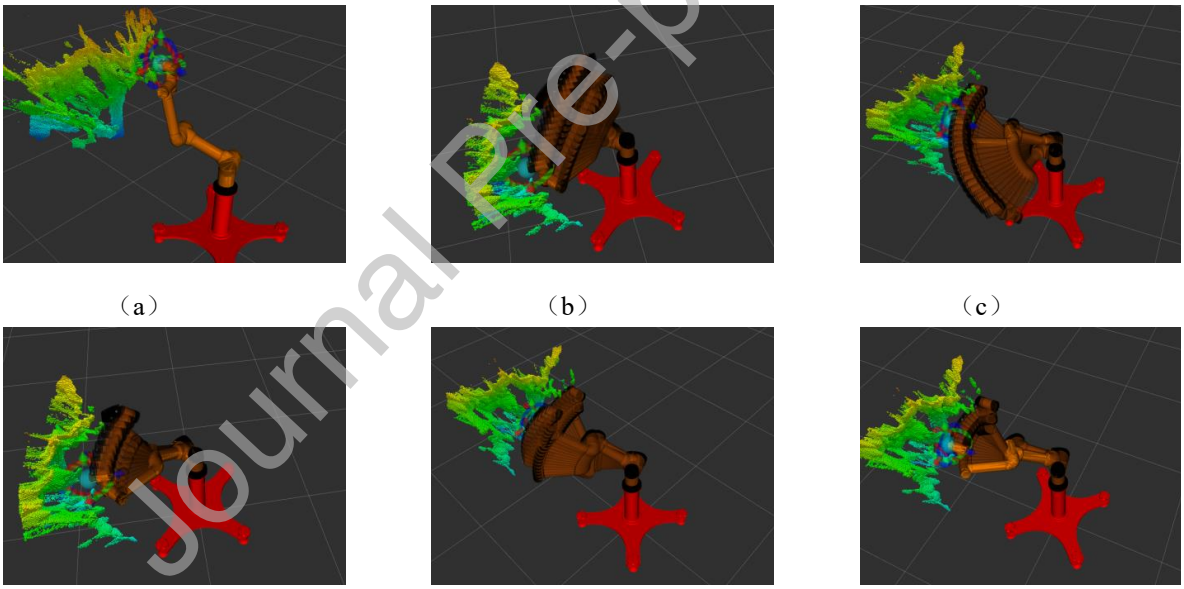
To further simulate the real picking environment, this study adds a planning scenario with the presence of complex obstacles in the Rviz visualisation environment, where the robotic arm needs to move from Fig. 15(a) to 15(b). In complex environments, obstacles are composed of elements like branches, leaves, and fine twigs, which together create intricate spaces that increase the challenge of path planning. In the linear space of these two positions, there were not only obstacle branches, but also leaves and fine branches interfering, which requires the obstacle avoidance algorithm of the picking robotic arm to be able to plan the picking path more efficiently. The initial and target positions of the robotic arm were

shown in Fig. 15.

（a） （b）

Fig.15. Manipulator position in complex environments （a）Initial position，（b）Target position

All five algorithms in this experiment were carried out under the same initial conditions, the maximum planning time was set to 3 s. The simulation results were shown in Fig. 15.



（a） （b） （c）

（d） （e） （f）

Fig.16. Initial position of manipulator and motion planning of manipulator based on various algorithms in complex environments（a）Initial position（b）RRT（c）RRT*（d）RRT connect（e）Informed-RRT*（f）Optimized Informed-RRT*

By comparing the simulation results of the five algorithms under complex environments, it can be found that the optimized Informed RRT algorithm performs significantly better than the other algorithms in terms of path length and

path curvature. To better analyze the performance of different algorithms and to avoid the influence of algorithmic randomness on the test results, this study conducted 20 repetitions of motion planning tests for each algorithm in a complex scenario. The test results were shown in Table 3.

Table 3. Comparison of simulation experiment results under complex obstacle environment

Programming algorithm	Average sampling points	Average planning time (s)	Success rate (%)
RRT	1682	4.53	60.00
RRT Connect	1278	2.66	80.00
RRT*	1603	4.81	85.00
Informed-RRT*	1530	4.44	85.00
Optimized Informed-RRT*	762	3.15	90.00

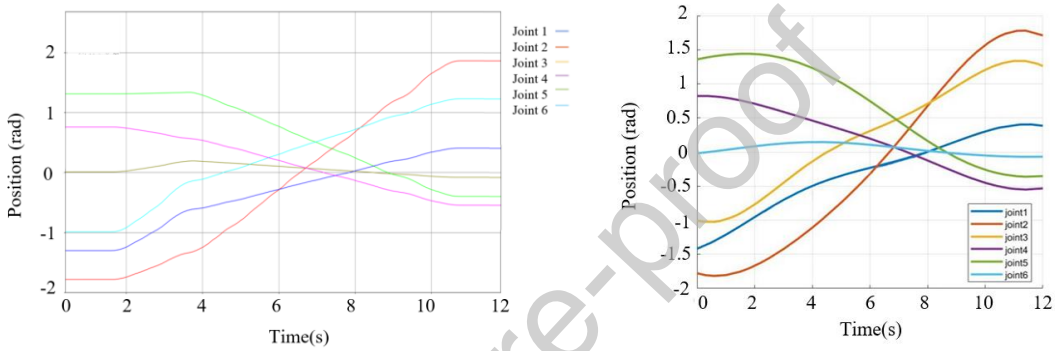
As can be seen from Table 3, The fewer sampling points, the shorter the planning time and the better the effective search ability of the algorithm.the optimized Informed-RRT* algorithm has a path planning success rate of up to 90% in complex environments, and its number of sampling points is significantly less than that of other algorithms. The average planning time, as one of the important indicators for evaluating the performance of real-time picking robots, is 3.15 s in the optimized Informed-RRT* algorithm, which is 30.46% less compared to the RRT algorithm. This is because the optimized Informed-RRT* algorithm can guide the search to converge quickly to the target area by adding target bias and adaptive step size methods, thus reducing the search time and the number of sampling points; at the same time, the addition of the pruning method can prune out the ineffective or redundant branches in time, thus obtaining a better-quality path and reducing the length of the path. In summary, when facing complex environments, the optimized Informed-RRT* algorithm proposed in this study has obvious advantages in planning performance compared with mainstream path planning algorithms.

3.2.4 Algorithm Validation on a Robotic Arm Motion Platform

According to the simulation results in Sections 3.2.2 and 3.2.3, the optimized Informed-RRT* algorithm proposed in this paper performs the best across various indicators. However, in the workspace of a robotic arm, safety also needs to be fully considered, especially during movement. Therefore, when applying the optimized Informed-RRT* algorithm

to a real robotic arm, its operational stability becomes a crucial index.

To verify that the optimized Informed-RRT* algorithm can support the stable operation of a picking robotic arm, this study monitored the positional transformations of the joints in real-time during the robotic arm's movement. The test scene was set in a complex picking environment, with the initial and target positions of the robotic arm shown in Fig. 15. By subscribing to the joint_states topic of the robotic arm, the positional change curves of each joint over time were plotted, as shown in Fig. 17 (a).



(a) (b)

Fig.17. Joint change curve (a) Variation curve of each joint position with time (b) Fitted curves for each joint position over time

In this study, 30 sets of position change data for the six joints of the robotic arm were separately collected. The displacement as a function of time was analyzed using Fourier curve fitting to understand the regularity and performance characteristics of the robotic arm during motion. Fig.17(b) presents the fitted curves for each joint.

Table 4. Evaluation indexes of fitted curves for each joint position over time

Joint Name	R-square	SSE	RMSE
Joint 1	0.9968	0.03348	0.03735
Joint 2	0.9991	0.05146	0.04449
Joint 3	0.9975	0.05333	0.04714
Joint 4	0.9989	0.07937	0.01747
Joint 5	0.9974	0.03672	0.03758
Joint 6	0.9840	0.02671	0.01014
Average	0.9956	0.04685	0.03236

As shown in Table 4, the average coefficient of determination for the six fitted curves was 0.9956, which is close to

1, indicating that the fitted model can explain the data variability well. The average sum-of-squares error (SSE) was 0.04685, which is close to 0, suggesting that the total error of the fitted model relative to the actual data was relatively small. SSE represents the sum of the squared differences between the observed values and the values predicted by the model, and a smaller SSE indicates a better fit of the model to the actual data. Additionally, the average root-mean-square error (RMSE) was 0.03236, which was also very small, further suggesting that the prediction error of the fitted model was minimal. The degree of freedom in this context refers to the number of independent data points used in fitting the model minus the number of parameters estimated, which influences the variance and error estimations. In summary, the resulting curve fit was excellent, and the fitted model closely matches the actual data, effectively explaining the data variability and exhibiting high prediction accuracy. The position-time fitting functions for the six joints are shown in equations (5) to (10).

$$f(x_1)=-0.3857-0.7161\times cos(0.3639\times x_1)-0.3394\times sin(0.3639\times x_1) -0.2156\times cos(0.7278\times x_1)+0.06449\times sin(0.7278\times x_1) \quad (5)$$

$$f(x_2)=0.09935-1.852\times cos(0.2643\times x_2)-0.3378\times sin(0.2643\times x_2) \quad (6)$$

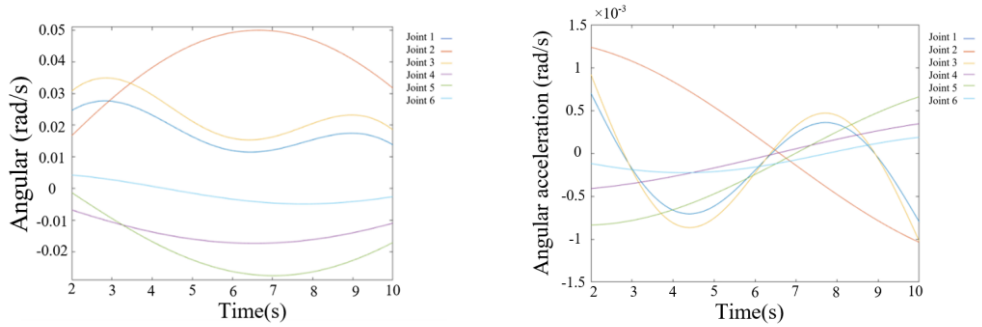
$$f(x_3)=-0.1907-0.9173\times cos(0.3634\times x_3)-0.4588\times sin(0.3634\times x_3) -0.2759\times cos(0.7268\times x_3)+0.08302\times sin(0.7268\times x_3) \quad (7)$$

$$f(x_4)=0.08774+0.6691\times cos(0.2568\times x_4)-0.08311\times sin(0.2568\times x_4) \quad (8)$$

$$f(x_5)=0.4751+0.7634\times cos(0.303\times x_5)-0.4915\times sin(0.303\times x_5) \quad (9)$$

$$f(x_6)=0.03239-0.04299\times cos(0.4537\times x_6)-0.09883\times sin(0.4537\times x_6) \quad (10)$$

According to Fig. 17(a), each joint of the robotic arm starts moving before 2 seconds and stops moving after 10 seconds. The joint displacement time function of the robotic arm was analyzed by taking the first-order and second-order derivatives within the interval of 2 to 10 seconds. Therefore, this study focuses on the motion of the robotic arm from 2 to 10 seconds, solving the first-order and second-order derivatives of the joint displacement time function within this interval. This analysis yields the curves of angular velocity and angular acceleration of the joints with respect to time, as shown in Fig. 18.



（a） （b）

Fig.18. Joint change curve（a）Curve of angular velocity of each joint with time（b）Curve of angular acceleration versus time for each joint

Observing Fig.18, it was evident that the angular velocity and angular acceleration of the robotic arm exhibit a smooth and continuous trend throughout the movement, with no significant sudden changes. This indicates that the robotic arm moves at a uniform and steady speed while performing tasks. These results verify that the optimized Informed-RRT* algorithm ensures the accuracy and stability of the robotic arm's movements, while also enhancing its movement efficiency and operational lifespan.

4. Conclusion

This study demonstrates that the optimized Informed-RRT* algorithm significantly improves the efficiency and effectiveness of robotic path planning performance, addressing key challenges in winter jujube harvesting. The main findings are as follow:

(1) In the 2D simulation environment, the optimized Informed-RRT* algorithm demonstrated superior performance compared to traditional algorithms, such as RRT, RRT Connect, and RRT*. It achieved the shortest average path length (7.23 m) while significantly reducing the number of sampling nodes. Although the algorithm's average planning time (5.03 s) was slightly longer due to continuous optimization, the smoother and more efficient paths ensured a 100% success rate.

(2) In the 3D simulation environment, the optimized Informed-RRT* algorithm maintained its advantage, outperforming all tested algorithms in terms of path length and efficiency. The algorithm achieved a success rate of 100% and reduced the average sampling nodes by 88.39% compared to RRT, indicating its capability to handle high-dimensional spaces effectively. Adaptive step size and target bias strategies played a key role in focusing the search and reducing computational complexity.

(3) In dynamic environments with dense obstacles like branches and leaves, The optimized Informed-RRT* algorithm, with pruning strategies, achieved a 90% success rate and reduced planning time by 30.46% in dynamic environments. Tests on a robotic arm confirmed its stability, making it practical for efficient obstacle avoidance in dynamic agricultural settings.

These findings indicate that the optimized Informed-RRT* algorithm not only enhances the planning efficiency and accuracy of robotic arms in simulated environments but also ensures stability and reliability in real-world applications. This research provides a robust framework for addressing motion planning challenges in agricultural robotics and other fields requiring high-precision and efficient path planning. Future work will explore further optimizations to reduce planning time and extend the algorithm’s applicability to larger-scale dynamic environments.

CRediT authorship contribution statement

Anxiang Huang: Methodology, Software, Writing - original draft, Visualization. **Chenhao Yu:** Writing - review & editing, formal analysis. **Junzhe Feng:** Investigation, Validation. **Xing Tong:** Investigation, Validation. **Ayanori Yorozu:** Validation, Writing - review & editing. **Akihisa Ohya:** Writing - review & editing. **Yaohua Hu:** Writing - review & editing, Funding acquisition, Supervision.

Conflicts of Competing Interest

The authors declare that they have no known competing financial interests or personal

relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Talent Start-up Project of Zhejiang A&F University Scientific Research Development Foundation No. 2021LFR066, the National Natural Science Foundation of China No. 32171894, and the China Scholarship Council.

References

[1] C. Yu, Y. Qiao, J. Feng, T. Guo, W. Luo, J. Guo, Y. Hu Optimization of Vibration Parameters for Red Jujube Trees with Different Diameters. Forests, 14(2023), 1287, https://doi.org/10.3390/f14071287.

[2] C. Yu, J. Feng, Z. Zheng, J. Guo, Y. Hu A lightweight SOD-YOLOv5n model-based winter jujube detection and counting method deployed on Android. Comput. Electron. Agric. 218(2024a), 108701, https://doi.org/10.1016/j.compag.2024.108701.

[3] J. Guo, Z. Yang, M. Karkee, Q. Jiang, X. Feng, Y. He Technology progress in mechanical harvest of fresh market strawberries. Comput. Electron. Agric. 226(2024), 109468, https://doi.org/10.1016/j.compag.2024.109468.

[4] L. Ye, F. Wu, X. Zou, J. Li Path planning for mobile robots in unstructured orchard environments: An improved kinematically constrained bi-directional RRT approach. Comput. Electron. Agric. 215(2023), 108453, https://doi.org/10.1016/j.compag.2023.108453.

[5] Y. Dai, C. Xiang, Y. Zhang, Y. Jiang, W. Qu, Q. Zhang A review of spatial robotic arm trajectory planning. Aerospace, 9(2022), 361, https://doi.org/10.3390/aerospace9070361.

[6] A. Thakur, S. Venu, M. Gurusamy An extensive review on agricultural robots with a focus on their perception systems. Comput. Electron. Agric. 212(2023), 108146,

https://doi.org/10.1016/j.compag.2023.108146.

[7] I. Etikan, K. Bala Sampling and sampling methods. Biom. biostat. int. j. 5(2017), 00149, DOI:10.15406/bbij.2017.05.00149.

[8] O. Khatib Real-time obstacle avoidance for manipulators and mobile robots. Int J Rob Res. 5(1986), 90-98, https://doi.org/10.1177/027836498600500106.

[9] S. Alshammrei, S. Boubaker, L. Kolsi Improved Dijkstra algorithm for mobile robot path planning and obstacle avoidance. Comput. Mater. Contin, 72(2022), 5939-5954, DOI: 10.32604/cmc.2022.028165.

[10] J. Zhong, T. Wang, L. Cheng Collision-free path planning for welding manipulator via hybrid algorithm of deep reinforcement learning and inverse kinematics. Complex intell. systems. (2021), 1-14, https://doi.org/10.1007/s40747-021-00366-1.

[11] T. S. Lembono, E. Pignat, J. Jankowski, S. Calinon Learning constrained distributions of robot configurations with generative adversarial network. IEEE robot. autom. lett. 6(2021), 4233-4240, DOI:10.1109/LRA.2021.3068671.

[12] L. Kavraki, J.-C. Latombe. Randomized preprocessing of configuration for fast path planning. Proceedings of the 1994 IEEE International Conference on Robotics and Automation, 1994. IEEE, 2138-2145, DOI:10.1109/ROBOT.1994.350966.

[13] K. Magzhan, H. M. Jani A review and evaluations of shortest path algorithms. Int. J. Sci. Technol. Res, 2(2013), 99-104.

[14] X. Gu, L. Liu, L. Wang, Z. Yu, Y. Guo Energy-optimal adaptive artificial potential field method for path planning of free-flying space robots. J Franklin Inst. 361(2024), 978-993, https://doi.org/10.1016/j.jfranklin.2023.12.039.

[15] W. Ji, J. L. Li, D. A. Zhao, Y. Jun Obstacle avoidance path planning for harvesting robot manipulator based on MAKLINK graph and improved ant colony algorithm. Appl. Mech. Mater. 530(2014), 1063-1067, https://doi.org/10.4028/www.scientific.net/AMM.530-531.1063.

[16] M. Kang, Q. Chen, Z. Fan, C. Yu, Y. Wang, X. Yu A RRT based path planning scheme for multi-DOF robots in unstructured environments. Comput. Electron. Agric. 218(2024), 108707, https://doi.org/10.1016/j.compag.2024.108707.

[17] L. E. Kavraki, P. Svestka, J.-C. Latombe, M. H. Overmars Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE trans. robot. autom. 12(1996), 566-580, DOI:10.1109/70.508439.

[18] S. M. LaValle, J. J. Kuffner Rapidly-exploring random trees: Progress and prospects: Steven m. lavalle, iowa state university, a james j. kuffner, jr., university of tokyo, tokyo, japan. Algorithmic and computational robotics, (2001), 303-307.

[19] M. A. R. Pohan, B. R. Trilaksono, S. P. Santosa, A. S. Rohman Path planning algorithm using the hybridization of the rapidly-exploring random tree and ant Colony systems. IEEE Access, 9(2021),153599-153615, DOI:10.1109/ACCESS.2021.3127635.

[20] H. Kim, H. Seo, J. Kim, H. J. Kim. Sampling-based motion planning for aerial pick-and-place. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019. IEEE, 7402-7408, DOI:10.1109/IROS40897.2019.8967922.

[21] H. Zhang, Q. Sheng, Y. Sun, X. Sheng, Z. Xiong, X. Zhu A novel coordinated motion planner based on capability map for autonomous mobile manipulator. Rob Auton Syst. 129(2020), 103554, https://doi.org/10.1016/j.robot.2020.103554.

[22] J. Huh, B. Lee, D. D. Lee. Constrained sampling-based planning for grasping and manipulation. 2018 IEEE International Conference on Robotics and Automation (ICRA), 2018. IEEE, 223-230, DOI:10.1109/ICRA.2018.8461265.

[23] M. M. Alam, T. Nishi, Z. Liu, T. Fujiwara A Novel Sampling-Based Optimal Motion Planning Algorithm for Energy-Efficient Robotic Pick and Place. Energies, 16(2023), 6910, https://doi.org/10.3390/en16196910.

[24] J. Ichnowski, W. Lee, V. Murta, S. Paradis, R. Alterovitz, J. E. Gonzalez, I. Stoica, K. Goldberg. Fog robotics algorithms for distributed motion planning using lambda serverless computing. 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020. IEEE, 4232-4238, DOI:10.1109/ICRA40945.2020.9196651.

[25] B. Wang, D. Ju, F. Xu, C. Feng, G. Xun CAF-RRT*: A 2D path planning algorithm based on circular arc fillet method. IEEE Access, 10(2022), 127168-127181, DOI:10.1109/ACCESS.2022.3226465.

[26] B. Lacevic, D. Osmankovic. Improved C-space exploration and path planning for robotic manipulators using distance information. 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020. IEEE, 1176-1182, DOI:10.1109/ICRA40945.2020.9196920.

[27] Z. Chen, Y. Yang, X. Xu, A. Rodic. Path planning of redundant series manipulators based on improved RRT algorithms. 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2019. IEEE, 553-557, DOI:10.1109/ROBIO49542.2019.8961770.

[28] C. Yu, X. Shi, W. Luo, J. Feng, Z. Zheng, A. Yorozu, Y. Hu, J. Guo MLG-YOLO: A Model for Real-Time Accurate Detection and Localization of Winter Jujube in Complex Structured Orchard Environments. Plant Phenomics, 6(2024b), 0258, DOI: 10.34133/plantphenomics.0258.

[29] J. D. Gammell, S. S. Srinivasa, T. D. Barfoot. 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, 2014. IEEE, 2997-3004, DOI:10.1109/IROS.2014.6942976.

[30] A. Mahanti, A. Bagchi AND/OR graph heuristic search methods. Journal of the ACM (JACM), 32(1985), 28-51, https://doi.org/10.1145/2455.2459.

[31] R. Zhang, H. Guo, D. Andriukaitis, Y. Li, G. Królczyk, Z. Li Intelligent path planning by an improved RRT algorithm with dual grid map. Alex. Eng. J. 88(2024), 91-104, https://doi.org/10.1016/j.aej.2023.12.044.

[32] A. Aleti, I. Moser A systematic literature review of adaptive parameter control methods for evolutionary algorithms. ACM Computing Surveys (CSUR), 49(2016), 1-35, https://doi.org/10.1145/2996355.

[33] L. Lozano, A. L. Medaglia On an exact method for the constrained shortest path problem. Comput Oper Res. 40(2013), 378-384, https://doi.org/10.1016/j.cor.2012.07.008.

Ethical Statement

In all our team's work and research, we strive to uphold honesty, integrity, and professionalism. We respect the rights, dignity, and privacy of all individuals involved, ensuring their well-being remains a priority. Our team will make every effort to avoid any potential harm and will properly acknowledge the contributions of others, ensuring that plagiarism is avoided and transparency is maintained when presenting results.

We are committed to minimizing any negative impacts arising from our processes and ensuring that the methods we

employ are reasonable and ethically sound. Additionally, we will adhere to relevant laws, regulations, and institutional guidelines, and, where necessary, obtain informed consent from participants. Confidentiality will be maintained unless disclosure is required by law or in exceptional circumstances.

We recognize the responsibility that comes with communicating our results and will aim to represent our findings fairly and accurately, avoiding any actions that may mislead or negatively impact stakeholders, the public, or the environment.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.