symmetry

MDPI

Article

Autonomous Obstacle Avoidance Path Planning for Grasping Manipulator Based on Elite Smoothing Ant Colony Algorithm

Xiaoling Meng [1,2] and Xijing Zhu [1,2,*]

[1] Shanxi Key Laboratory of Advanced Manufacturing Technology, North University of China, Taiyuan 030051, China
[2] School of Mechanical Engineering, North University of China, Taiyuan 030051, China
* Correspondence: zxj161501@nuc.edu.cn

Abstract: Assembly robots have become the core equipment of high-precision flexible automatic assembly systems with a small working range. Among different fields of robot technology, path planning is one of the most important branches. In the present study, an elite smoothing ant colony algorithm (ESACO) is proposed for spatial obstacle avoidance path planning of the grasping manipulator. In this regard, the state transition probability and pheromone update strategies are improved to enhance the search capability of path planning symmetry and the convergence of the algorithm. Then a segmented B-spline curve is presented to eliminate path folding points and generate a smooth path. Finally, a manipulator control system based on the Arduino Uno microcontroller is designed to drive the manipulator according to the planned trajectory. The experimental results show that the performance of the ESACO algorithm in different scenarios has symmetry advantages, and the manipulator can efficiently complete the simulation trajectory with high accuracy. The proposed algorithm provides a feasible scheme for the efficient planning of manipulators in equipment manufacturing workshops.

Keywords: elite smoothing ant colony algorithm; grasping manipulator; autonomous obstacle avoidance; global path planning

check for updates

Citation: Meng, X.; Zhu, X. Autonomous Obstacle Avoidance Path Planning for Grasping Manipulator Based on Elite Smoothing Ant Colony Algorithm. *Symmetry* **2022**, *14*, 1843. https://doi.org/10.3390/sym14091843

Academic Editors: Wenyin Gong and Libao Deng

Received: 9 August 2022
Accepted: 31 August 2022
Published: 5 September 2022

1. Introduction

With the rapid development of emerging industries and the increasing demands of society, combining the mobile manipulator with conventional machining and assembly is highly demanded to improve the automation level of operations [1,2]. Path planning is the basis for manipulators to complete path control safely and reliably [3,4]. The main objective of obstacle avoidance is to find an optimal and collision-free path from the starting point to the target point in a given environment [5,6]. It is worth noting that the smoothness of the path affects the energy consumption, operating efficiency, and trajectory tracking time of the mechanical arm, and a non-smooth path restricts robot movement and may cause unplanned slowdowns [7,8]. Therefore, the ideal obstacle avoidance path planning for space manipulators is of significant importance.

Currently, intelligent algorithms such as genetic algorithm (GA) [9,10], particle swarm algorithm (PSO) [11], ant colony optimization (ACO) [12,13], neural network [14], and rapidly exploring random trees (RRT) [15,16] have been widely applied in different applications in the manipulator path planning. Among these algorithms, the ant colony algorithm has superior characteristics such as strong robustness, good positive feedback mechanism, and inherent parallel mechanism [4,17]. Accordingly, this algorithm has been widely applied in path planning. Despite several significant features, the conventional ACO has disadvantages such as slow convergence speed, long path planning time, and low utilization of cyclic information [18,19]. In order to resolve these shortcomings, various methods have been proposed to improve the pheromone update and path search strategy and solve the obstacle avoidance path problem. In this regard, Miao et al. [20] proposed an improved adaptive ant colony algorithm (IAACO) with the obstacle exclusion factor and adaptive adjustment factor to realize a comprehensive global optimization of the robot path planning. Sangeetha et al. [21] proposed a fuzzy gain-based dynamic ant colony optimization (FG-DACO) for dynamic path planning to effectively plan collision-free and smooth paths. Jiao et al. [22] proposed an intelligent wheelchair path planning method using an adaptive state transfer strategy and an adaptive information update strategy to improve the traditional ant colony algorithm. Furthermore, Michalis Mavrovouniotis et al. [23] proposed a modal ant colony algorithm to improve the addressing efficiency. Jin et al. [24] proposed a fusion algorithm based on the improved ant colony algorithm-rolling window method that can reach the specified target area quickly and safely in complex dynamic environments.

A review of the literature indicates that numerous investigations have been carried out in terms of path smoothening. In this regard, Xiong et al. [25] combined a Voronoi-based scheme with ant colony optimization (ACO) and found collision-free optimal trajectories for multiple autonomous marine vehicles (AMVs). Then they measured oceanic parameters using a modified heuristic function. Liu et al. [12] proposed an optimization method combining pheromone diffusion and geometric local optimization to solve the convergence speed problem in the ant colony algorithm, and effectively shortened the search space of ants. Yang et al. [26] proposed an efficient double-layer ant colony algorithm (DL-ACO) to navigate the robot autonomously and designed a segmented B-sample curve with smoothing the paths only at the corners. Accordingly, a complete robot navigation path planning scheme was formed. Zhang et al. [27] constructed a nearly shortest path, applied the 2D path smoothing method to solve the 3D path smoothing problem, and provided a new velocity planning method to find the time-optimal path.

The performed literature survey indicates that although many algorithms have been proposed for finding collision-free smooth paths, further investigations are required especially in the field of intelligent algorithms with logical reasoning. To address these issues, the following contributions are made in this paper. Generally, these contributions can be categorized as efficient planning of a safe and smooth path in the presence of obstacles.

- An elite ant colony (EACO) algorithm is proposed, in which the decisive factor is introduced into the state transition equation to offset the error caused by the positive feedback system, and the attenuation factor is added to the pheromone update strategy to prevent the algorithm from falling into local optimum.
- Combining the B-spline curve with the improved EACO, the elite smoothing ant colony (ESACO) algorithm is proposed, where the smoothing strategy is applied to reduce unnecessary traversals in the search process and generate efficient paths.
- The performance of the three algorithms before and after the improvement is compared, and the results show that the ESACO algorithm generates shorter and smoother routes with higher convergence and reliability.
- Physical experiments are conducted using RobotStudio software, and the results are verified.

This paper is organized as follows: Section 2 presents the definition of the problem and describes the environment. Section 3 is focused on the algorithm design, improvement of the algorithm, and the B-sample smoothing method. Section 4 compares and analyzes the performance of different algorithms. Section 5 presents the results of the numerical simulations. Finally, the results and main achievements are summarized in Section 6.

2. Problem Statement and Environment Description

2.1. Problem Definition and Formulation

Generally, robots assemble components in a complex environment with assembling parameters such as time windows, size of equipment parts, and assembly accuracy [28]. Moreover, different production plans may affect the equipment assembly efficiency [29]. In the equipment manufacturing workshop, each manipulator is an independent assembly unit, and the end-effector and connecting rod of the manipulator also avoid obstacles during movement. To better describe the path planning for mechanical arm obstacle avoidance, the actual problem can be simplified using the following hypotheses:

- Assume that there are three static obstacles in the working environment;
- The starting and ending positions of the mechanical arm movement are known;
- The dimensions of the parts are within the clamping range of the mechanical arm;
- The equipment parts have been arranged in order in advance.

In this section, the path length, the maximum grasping range, the consumed energy to complete the task, and the maximum working radius of the manipulator are considered the modeling objectives. The affecting variables in the modeling are described in Table 1.

Table 1. Parameters description.

Parameters	Description
$L(x)$	The total length of the path
i, j	The points visited by ants, where $i = \{1, 2, \ldots, I\}$ and $j = \{1, 2, \ldots, J\}$
o	The number of obstacles where $o = \{0, 1, \ldots, O\}$
d_{ij}	The distance between points i and j
d_{io}	The distance between point i and the nearest obstacle o
a	The number of equipment parts where $a = \{0, 1, \ldots, A\}$
ϕ_a	The size of equipment parts
Φ_{min}, Φ_{max}	Minimum and maximum range of end grippers
υ_{ij}^{a}	Energy consumption of the mechanical arm a to move from point i to j
E_{max}	Maximum energy consumption of the mechanical arm
q_a	The weight of equipped parts
W_{max}	The maximum load of the mechanical arm
R_{max}	The maximum radius of operation of the mechanical arm
F	The position of mechanical arm
x_{iF}, x_{jF}	The variable of points i and j on the path to the mechanical arm
d_{iF}, d_{jF}	The distance between points i and j on the path to the mechanical arm

Based on these parameters, the objective function can be expressed in the form below:

$$\min L(x) = \sum_{i=1}^{I} \sum_{j=1}^{J} x_{ij} d_{ij} \tag{1}$$

This function is subjected to the following constraints:

$$\max\left\{\sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{o=1}^{O} x_{ij} d_{io}\right\} \tag{2}$$

$$\sum_{a=1}^{A} \phi_a \geq \Phi_{min} \tag{3}$$

$$\sum_{a=1}^{A} \phi_a \leq \Phi_{max} \tag{4}$$

$$\sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{a=1}^{A} \upsilon_{ij}^{a} \leq E_{max} \tag{5}$$

$$\sum_{a=1}^{A} q_a \leq W_{max} \tag{6}$$

$$\sum_{i=1}^{I} x_{iF} d_{iF} \leq R_{max} \tag{7}$$

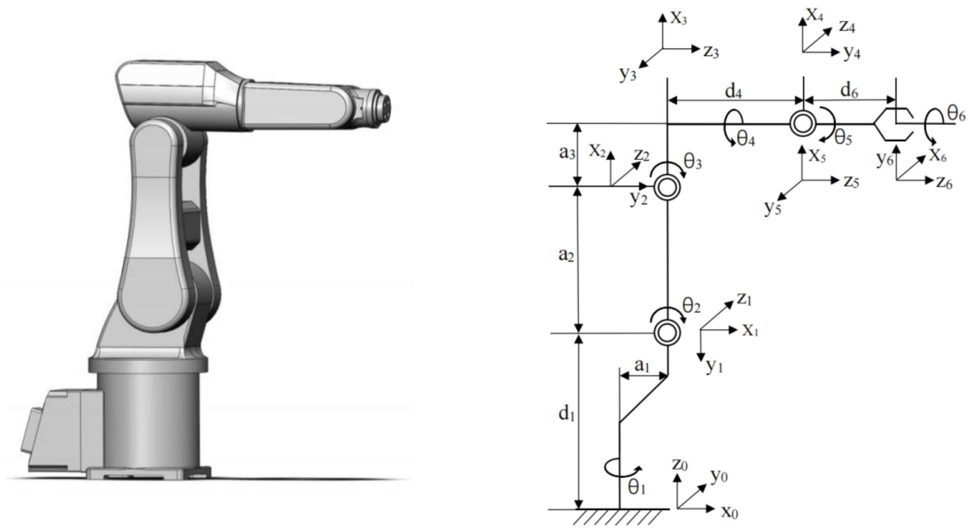
$$\sum_{j=1}^{J} x_{jF} d_{jF} \leq R_{max} \tag{8}$$

$$x_{ij} = \begin{cases} 1, \text{for a path from i to j} \\ 0, \text{others} \end{cases} \tag{9}$$

where Equation (1) indicates the minimization of the distance to the target. Equation (2) indicates the maximization of the distance between the mechanical arm and the obstacles. Equations (3) and (4) denote the size of the parts within the maximum and minimum range of the end gripper, respectively. Equation (5) indicates the maximum energy consumption to complete the assembly process. Equation (6) indicates that the weight of the part cannot exceed the maximum load of the manipulator. Equations (7) and (8) denote that the planned path is within the maximum operating radius of the mechanical arm. Equation (9) refers to the path that can be included or dropped depending on its existence.

2.2. Manipulator and Environment Description

In the present study, a 6-DOF robot is used to investigate path planning during equipment assembly [30]. Figure 1a illustrates the configuration of the manipulator, which can be simplified as a six-link mechanism, which is connected by joint pairs and moves the end-effector through the rotation of six joints. Figure 1b shows the base coordinate system of the mechanical arm. Furthermore, Tables 2 and 3 show the technical parameters of the mechanical arm and the D-H parameters of the system, respectively.



(**a**) Configuration of the manipulator (**b**) Degrees of freedom of joints of the manipulator

Figure 1. Configuration of the manipulator and DOF of its joints.

Table 2. Technical parameters of the mechanical arm.

Numbers	Indicators	Parameters
1	Maximum loads	3 kg
2	Repeat positioning accuracy	±0.01 mm
3	Maximum working radius	580 mm
4	Weight of the machine	25 kg
5	Arm loads	0.3 kg
6	Maximum speed of grabbing 1 kg items	6.2 m/s
7	Maximum acceleration for grabbing 1 kg items	28 m/s^2

Table 3. D-H parameters of the system.

Linkage i	θ_i/(°)	d_i/(mm)	a_i/(mm)	α_i/(°)	Joint Range/(°)
1	q1	335	40	90	−170 to 170
2	q2	0	280	0	−70 to 120
3	q3	0	70	90	−110 to 70
4	q4	313	0	−90	−180 to 180
5	q5	0	0	90	−120 to 120
6	q6	81	0	0	−360 to 360

In Figure 1b, θ_i and d_i indicate the joint angle and the offset of linkages, respectively. Moreover, a_i and α_i denote the length of linkages and the twist angle, respectively.

The transformation matrix of the manipulator ${}^{0}_{6}T$ represents the position correlation between the end linkage posture (n, o, a, p) and the joint coordinate system, which can be obtained by multiplying successively the transformation matrix of each linkage ${}^{i-1}_{i}T(i = 1, 2, \ldots, 6)$ in the form below:

$$ {}^{0}_{6}T = {}^{0}_{1}T(\theta_1){}^{1}_{2}T(\theta_2)\cdots{}^{5}_{6}T(\theta_6) = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10} $$

where $n = [n_x, n_y, n_z]^T$, $o = [o_x, o_y, o_z]^T$, $a = [a_x, a_y, a_z]^T$ denote the normal vector, orientation vector and approach vector of the coordinate system at the end of the manipulator, respectively. $p = [p_x, p_y, p_z]^T$ denotes the position vector of the end of the manipulator [31].

To reduce the spatial complexity, the cylindrical envelope method and the axis-aligned bounding box (AABB) method [32] are used to simplify the manipulator and the obstacle, respectively. The main idea of these methods is to transform the collision between the manipulator and the obstacle into an interference problem between the spatial cylinder and the rectangular enclosing box, thereby reducing the computational expenses of interference detection and improving the efficiency of path planning. Figure 2 shows the schematic of the established model, where r_1 and r_2 are the radii of the two cylinders.

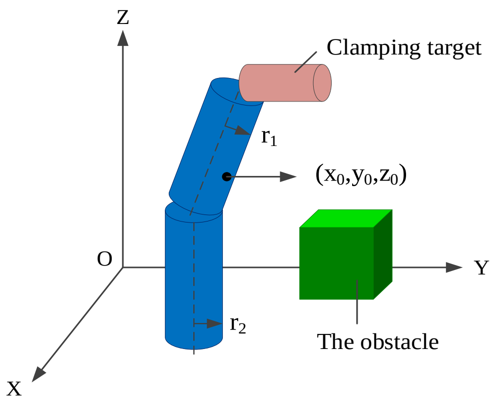


Figure 2. Bounding model of manipulator and obstacle.

When an arbitrary point (x_0, y_0, z_0) on the cylinder satisfies the following conditions, then a collision occurs and vice versa: $x_{min} \le x_0 \le x_{max}$, $y_{min} \le y_0 \le y_{max}$, $z_{min} \le z_0 \le z_{max}$. Where the points $(x_{min}, y_{min}, z_{min})$ and $(x_{max}, y_{max}, z_{max})$ represent the minimum and maximum coordinates of the bounding box projected on the X-, Y-, and Z-axes, respectively.

3. Algorithm Design

3.1. Ant Colony Algorithm

The ant colony algorithm is an evolutionary algorithm proposed by M. Dorigo in the 1990s [33]. This algorithm is based on the behavior of ants that always search for the shortest path between the food source and the nest when foraging. The ant $k(k = 1, 2, \ldots\ldots, m)$ selects new path and transition probability according to pheromone concentration [34].

The mathematical model of the ant colony algorithm can be established as follows: Suppose the number of ants is m, the distance between node i and node j is d_{ij}, and $\eta_{ij}(t) = 1/d_{ij}$ is the expected heuristic function. The concentration of the pheromone of the node (i, j) at time t is $\tau_{ij}(t)$, and the initial pheromone concentration is $\tau_{ij}(0) = \tau(0)$. $\Delta\tau_{ij}$ represents the increment of pheromone from point i to point j. Moreover, $p^k_{ij}(t)$ denotes the probability that ant k moves from node i to node j, and it can be mathematically expressed as follows:

$$p_{ij}^{k}(t) = \begin{cases} \frac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum\limits_{s\in allowed_k}\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}, & s \in allowed_k \\ 0, & others \end{cases} \quad (11)$$

where α is the pheromone factor indicating the relative importance of the pheromone traces, β denotes the expected heuristic factor reflecting the strength of the deterministic factor, and $allowed_k$ is the matrix that ant k can access.

The pheromone concentration on the path is updated once for each cycle completed by the ant, and $\rho(0 < \rho \leq 1)$ is the volatile factor of pheromone. The pheromone concentration is updated as follows:

$$\begin{cases} \tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} \\ \Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \end{cases}, 0 < \rho < 1 \quad (12)$$

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k}, & (i,j) \in T \\ 0, & (i,j) \notin T \end{cases} \quad (13)$$

where Q, T, and L_k denote the pheromone intensity, the path constructed by ant k, and the path length, respectively.

3.2. Elite Smoothing Ant Colony Optimization

The conventional ant colony algorithm has a promising performance in solving path optimization problems. However, it also has some drawbacks. In order to generate collision-free paths in complex spaces, an elite smoothing ant colony (ESACO) algorithm is proposed. In this algorithm, it is intended to improve the state transfer probability and pheromone update model and optimize the trajectory using B-sample curves.

3.2.1. Improvement of State Transition Probability

In the ant colony algorithm, the ants calculate the next visited points according to the state transition probability equation. In this regard, a decision factor ω is defined, which reflects the importance of eliminating unnecessary search in probabilistic selection by ants. When $\omega_0 \leq \omega$, the probability of ants selecting the next node is only determined by the inter-node path length, when $\omega_0 > \omega$, this probability depends on the pheromone concentration and the inter-node path length $(0 \leq \omega_0 \leq 1)$. The improved probability equation can be expressed as follows:

$$p_{ij}^{k}(t) = \begin{cases} \frac{\eta_{ij}^{\beta}(t)}{\sum\limits_{s\in allowed_k}\eta_{ij}^{\beta}(t)} & s \in allowed_k, \omega_0 \leq \omega \\ \frac{\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum\limits_{s\in allowed_k}\tau_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)} & s \in allowed_k, \omega_0 > \omega \\ 0 & s \notin allowed_k \end{cases} \quad (14)$$

The improved state transfer formula allows the ants to have multiple probabilistic methods to select the next point, which counteracts the error caused by the positive feedback system.

3.2.2. Optimization of the Pheromone Update Strategy

As time passes, the pheromone concentration on the optimal route increases gradually, while the concentration on other routes will decrease. Therefore, its characteristics cannot be fully played out. In the present study, the attenuation coefficient θ is introduced to weaken the pheromone increment in the path finding, which reflects whether the global pheromone feedback can make the algorithm search for the optimal solution at a reasonable evolutionary rate. The modified expression can be expressed as follows:

$$\begin{cases} \tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + (1-\rho)\Delta\tau_{ij} - \frac{\theta^2}{L_k}, 0 < \rho < 1 \\ \Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \end{cases} \tag{15}$$

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_k}, & (i,j) \in T \\ 0, & (i,j) \notin T \end{cases} \tag{16}$$

The modified pheromone update equation gives the ants more reference information, makes the algorithm less likely to fall into the local optima, and solves the problem of low utilization of the cyclic information.

3.2.3. Trajectory Optimization

The segmented B-spline curve has the advantages of continuity, locality, convexity, and reasonable fitting in motion planning [35,36]. In this section, it is intended to use the B-spline curve to smooth the avoidance path of the manipulator. This can be mathematically expressed as follows:

$$C(u) = \sum_{i=0}^{n} P_i N_{i,p}(u), u \in [0,1] \tag{17}$$

where P_i and $N_{i,p}(u)$ denote the i-th point and the B-spline based function defined by the following DeBoor–Cox recursive equation, respectively [37,38].

$$N_{i,0}(u) = \begin{cases} 1, u_i \le u \le u_{i+1} \\ 0, \text{ others} \end{cases} \tag{18}$$

$$N_{i,p}(u) = \begin{cases} \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \end{cases} \tag{19}$$

Equations (17) and (18) define the recursive algorithm of base functions, where u_i is called the knot value. The first-order basis function is calculated by the corresponding knot vector $[u_i, u_{i+1}]$ and then recursively substituted into Equation (19) to calculate the high-order basis function from 2 to p.

3.3. The Global Path Planning Process of the ESACO Algorithm

The steps of the improved algorithm for path planning are as follows:

Step 1. Establish the workspace of the manipulator and initialize the parameters. Then, determine the starting and target position.

Step 2. Set all ants on the starting point. Select the transition probability of the next path node according to Equation (14).

Step 3. Calculate the distance of each ant in the path node matrix.

Step 4. Calculate the fitness values of all ants and select the best ant as the next set of solutions.

Step 5. Perform the global update of the pheromone concentration according to Equations (15) and (16), and enter the next loop iteration.

Step 6. Determine whether the termination condition is satisfied. If it is satisfied, output the optimal solution, or return to Step 2.

Step 7. Smoothing the optimal path by Equations (17)–(19).

Figure 3 shows the flowchart of the manipulator avoidance path planning based on the ESACO algorithm.

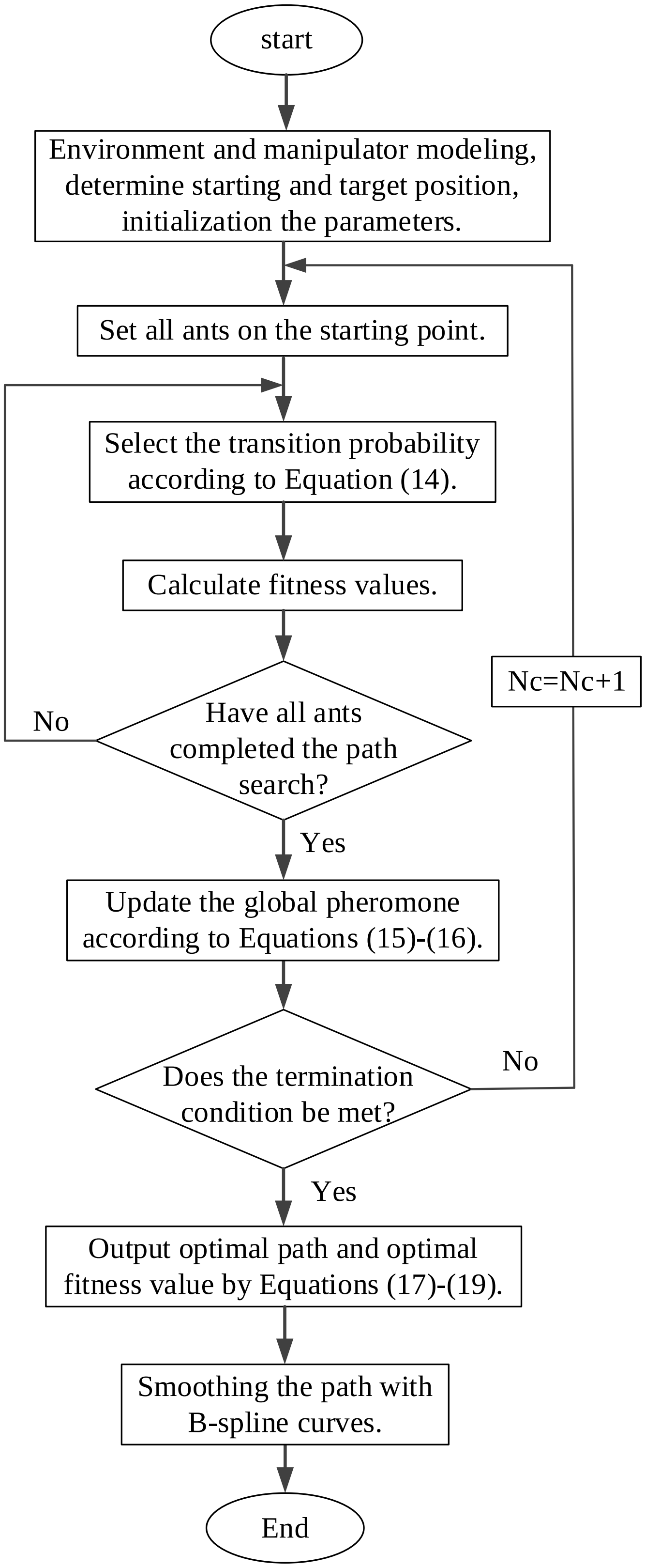


Figure 3. Flowchart of the improved ESACO algorithm.

The pseudo-code of the ESACO-based path planning process is shown in Algorithm 1.

Algorithm 1 Pseudo-code for ESACO-based path planning

```
procedure ESACO
  Build environment model;
  Set the size and location of obstacles, starting point S and ending point E;
  Initialize the number of ants m, the maximum number of iterations N_max, weights α, β, ρ, Q, and the new parameters ω, θ;
  for N = 1 to N_max do
   Put all ants into the S
    while ant k does not reach E do
     allowed_k ← the set of reachable grids for k
    Select the next grid by Equation (14)
    end while
    if all ants have arrived E then
    L_k ← path length of ant k
    L_best ← the best path in this iteration
    Update the global pheromone by Equations (15) and (16)
     if the fitness is optimal then
      best-fitness ← minimum fitness value
      Best-Fitness ← record the change of fitness values
     end if
    end if
  end for
  Output optimal path and optimal fitness values
  Smoothing the optimal path by Equations (17)–(19)
end procedure
```

4. Simulations and Analysis

Three cubic obstacles with different sizes are established to test the obstacle avoidance performance of the algorithm, the size of which is (4 cm × 4 cm × 8 cm), (4.5 cm × 4.5 cm × 5.5 cm), and (6 cm × 6 cm × 6 cm). The coordinates of the starting and the target point are (1 cm, 10 cm, 1 cm) and (21 cm, 8 cm, 12 cm). By substituting the parameters of Table 1 into Equation (10), the unique position and pose of the end-effector with respect to the base coordinate system are obtained as follows:

$$T_{start} = \begin{bmatrix} -0.816 & 0.437 & 0.377 & 0.319 \\ -0.541 & -0.807 & -0.236 & -0.229 \\ 0.201 & -0.397 & 0.896 & -0.431 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_{end} = \begin{bmatrix} 0.885 & 0.414 & 0.215 & -0.631 \\ 0.466 & -0.778 & -0.421 & -0.506 \\ -0.007 & 0.472 & -0.881 & -0.273 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4.1. Simulation Results and Performance Comparison

In the present study, the elite formulation and smoothing method are used to optimize the algorithm twice. Therefore, the improved algorithms are called EACO and ESACO. In order to ensure the accuracy of the experiment, the parameters of the three algorithms are set uniformly. By referring to the parameter range in other citations, considering other researchers' settings, and trying a variety of combination simulations, the parameter combination with the best experimental results is finally determined as: $m = 30$, $N = 50$, $\alpha = 1$, $\beta = 3$, $\rho = 0.3$, $Q = 100$, and the newly set parameters are taken as follows: $\omega = 0.3$, $\theta = 6$. In addition, considering the randomness of the algorithm, the three algorithms before and after the improvement are carried out 15 times for path planning with obstacles in the scene to test the effectiveness of the algorithm. The following indicators are used to evaluate the symmetry performance of ACO, EACO, and ESACO. Table 4 presents the recorded results.

(1) Path length/cm: The path length directly affects the total energy consumption of the mechanical arm and the best performance of the algorithm.
(2) Running time/s: The running time of the program is a prominent indicator of the efficiency of the algorithm.
(3) Collision detection: The collision relationship between the path and the obstacles determines whether the mechanical arm can continue moving. If there is a collision, the path planning will be invalid. If there is no collision, the action can be followed up.

Table 4. Four kinds of the performance record table.

Number	ACO			EACO			ESACO		
	(1)	(2)	(3)	(1)	(2)	(3)	(1)	(2)	(3)
1	77.578	5.88	1	59.715	4.94	0	52.604	3.68	0
2	71.360	5.97	1	59.417	5.00	1	51.640	3.74	0
3	66.419	6.04	1	56.471	5.08	1	46.402	3.85	1
4	65.941	5.80	1	64.170	5.16	1	51.686	3.93	1
5	65.251	6.04	1	60.892	5.22	0	52.141	3.91	0
6	53.940	6.15	0	57.029	5.00	0	50.338	3.79	0
7	67.345	5.70	0	57.299	4.70	0	49.065	3.73	0
8	65.769	5.89	0	63.915	4.93	0	54.120	3.66	0
9	58.946	5.77	0	61.429	5.21	1	52.596	3.76	1
10	67.966	5.93	1	58.223	4.81	1	48.880	3.80	0
11	56.137	5.87	0	60.055	4.87	0	52.369	3.93	0
12	69.015	5.84	1	63.581	5.11	0	56.266	3.93	0
13	67.211	6.12	1	55.239	4.83	1	46.398	3.92	1
14	60.441	5.88	0	49.596	4.86	0	42.020	3.83	0
15	56.629	5.57	0	62.153	5.19	0	50.596	4.01	0

Figure 4 shows the path length trend diagram. It is observed that the overall trend of each improved path is lower than the previous one. Table 5 shows the performance comparison of different algorithms. It is observed that the shortest path length is reduced from the original 53.940 cm to 49.596 cm to 42.020 cm, and the two improved shortest path lengths are 8% and 22.1% lower than the original one. The average path length is reduced from 64.663 cm to 59.279 cm to 50.475 cm, and the average path lengths after the two improvements are 8.3% and 21.9% lower than the initial one. The standard deviation is used to analyze the stability of the data with values of 6.133, 3.726, and 3.409, respectively. The results indicate that the quality of the optimized paths is more stable than that before optimization.

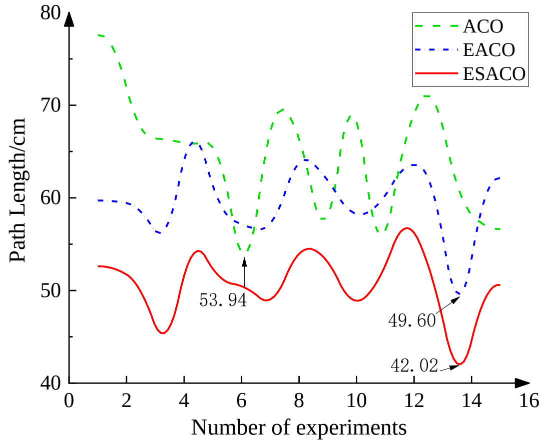


Figure 4. Path length trend of different algorithms.

Table 5. Comparison of the performance of ACO, EACO, and ESACO algorithms.

Method	Path Length			Running Time	
	Min. Path Length/cm	Average Path Length/cm	Standard Deviation	Average Running Time/s	Standard Deviation
ACO	53.940	64.663	6.133	5.897	0.150
EACO	49.596	59.279	3.726	4.994	0.158
ESACO	42.020	50.475	3.409	3.831	0.101

Figure 5 shows the trend of the running time, indicating shorter runtimes after each improvement. Table 5 shows that the average running time is reduced from 5.897 s to 4.994 s to 3.831 s, and the optimized values are 15.3% and 35% shorter than the original one. The standard deviations are 0.150, 0.158, and 0.101, respectively. The obtained results show that the lowest value can be achieved using the ESACO algorithm.

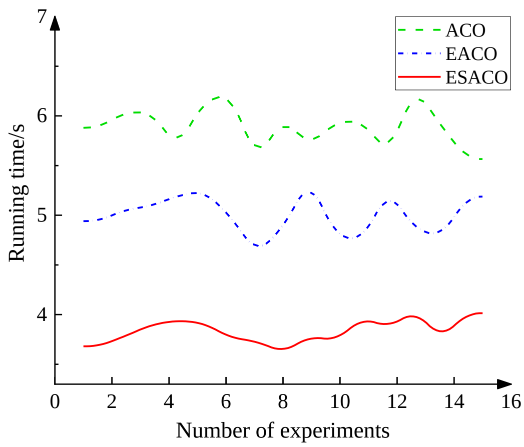


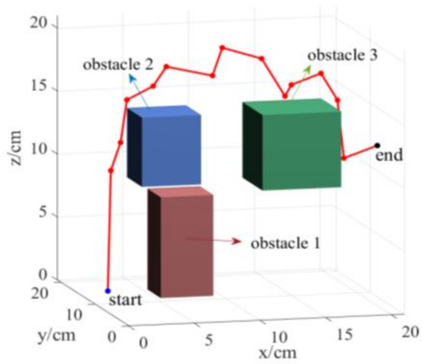
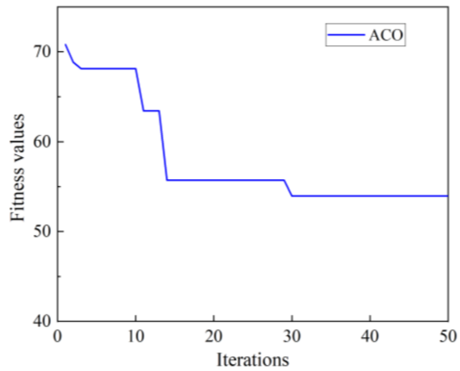
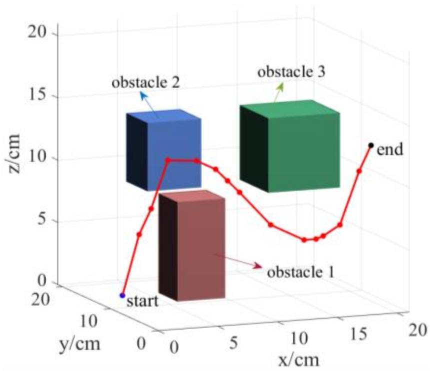
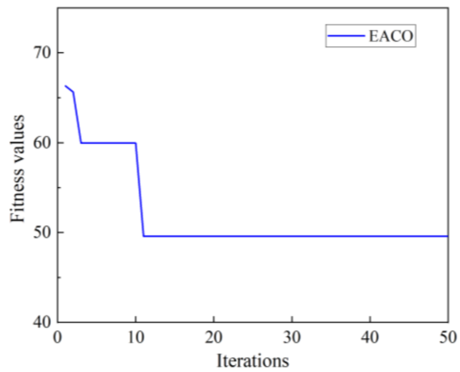
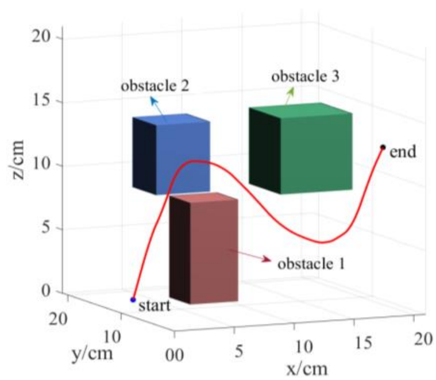
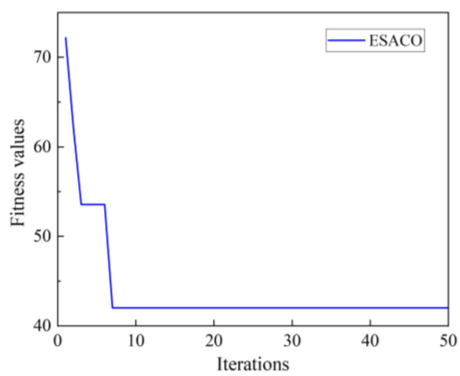
Figure 5. Running time trend of different algorithms.

Furthermore, by analyzing Table 4 with and without collisions (0 represents no collision, 1 represents collision), the original ACO has 7 collision-free times, EACO has 9 collision-free times, and ESACO has 11 collision-free times in the environment with obstacles, which further illustrates the feasibility and superiority of the improved algorithm in global obstacle avoidance.

4.2. Analysis of the Best Simulation Results of Three Algorithms

The path length is the most immediate and effective indicator to test algorithm improvement and mechanical arm obstacle avoidance. The performance comparison in Table 4 shows that the optimal path planned by ACO is 53.940 cm in Group 6, and the optimal paths planned by EACO and ESACO are 49.596 cm and 42.020 cm respectively, both in Group 14. Table 6 illustrates the optimal obstacle avoidance paths and iterative convergence curves for the three algorithms. The comparative path plot shows that the path obtained by ESACO is smoother than ACO and ESACO, where the path of ACO has reciprocal points that increase the path length, and the path of EACO is close to the optimal value; however, it is not smooth enough.

Table 6. Optimal obstacle avoidance paths and iterative convergence curves based on three algorithms.

Method	Optimal Obstacle Avoidance Path	Convergence Curve
ACO		
EACO		
ESACO		

The iterative convergence plots reveal that the fitness values of all three algorithms decrease as the number of iterations increases, and the curves show a horizontal trend when a certain optimal value is reached. Further analysis shows that the initial search fitness of the ACO algorithm is less than 72, the best path is obtained at 30 iterations, and the curve decreases slowly. The initial search fitness of the EACO algorithm is less than 70, while it decreases slightly faster than that of the ACO and achieves the best fitness in 11 iterations. Moreover, the initial fitness of the ESACO algorithm is greater than 72, and the correlation curve tends to decrease rapidly, reaching the best fitness value in 7 iterations. The results show that the first two algorithms tend to converge prematurely and fall into local optimum, while the improved ESACO algorithm converges faster and iterates better.

5. System Design and Experimental Discussions

5.1. System Design of the Gripping Mechanical Arm

Since the advent of embedded technology, this scheme has been widely used in manipulator control systems. It should be indicated that a dedicated microcomputer is an essential and inseparable component of this technology. Currently, Raspberry Pi, Arduino, and FPGA are the commonly used parts in the manipulator-embedded technology. Raspberry Pi [39] is a complete core processing chip with strong comprehensive performance. It has I/O ports that can expand external applications but cannot easily expand peripheral hardware. On the other hand, Arduino [40] is a software and hardware development platform that simplifies programming. It has a very strong, low-price, and expandable chip, which is widely used in the control development of various devices. Moreover, FPGA [41] is an integrated circuit that contains programmable logic elements. FPGA has an embedded programmable unit and a very flexible function.

In this section, the Arduino Uno microcontroller is utilized to generate the dynamic system of the manipulator. The control system of the grasping manipulator includes a key module, a power module, a display module, and an Arduino control module. The Arduino control board acts on the stepper motor drive module, which drives the steering gear of the manipulator to form a closed control system. Finally, the multi-angle-free and accurate movement of the manipulator is realized through system debugging. The structure diagram of the control system is presented in Figure 6.

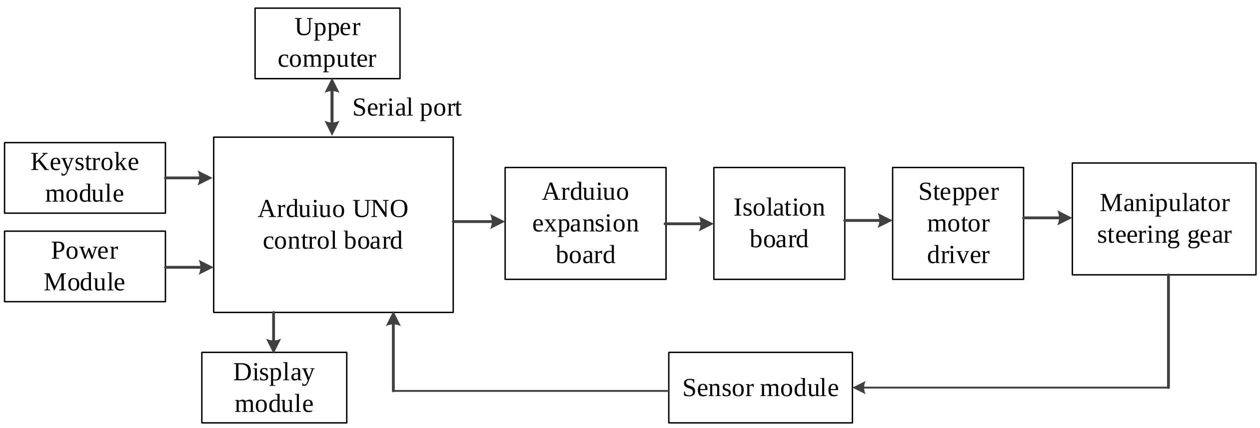


Figure 6. Block diagram of the mechanical arm control system.

5.2. Experimental Verification

The experimental system is implemented on Windows 10 operating system using RobotStudio 6.04 software, which is a computer application for robot off-line programming. It uses graphical programming and debugging of the robot system to operate the robot. It is unique in that it can simulate and optimize existing robot programs, and models according to the real environment. In the experiment, the information of initial pose, target pose, and obstacles are introduced into the algorithm module, then the mechanical arm performs obstacle avoidance path planning and collision detection. Finally, the control motor drives the manipulator to follow the planned trajectory. Figure 7 shows the working process of obstacle avoidance path planning for the grasping manipulator.

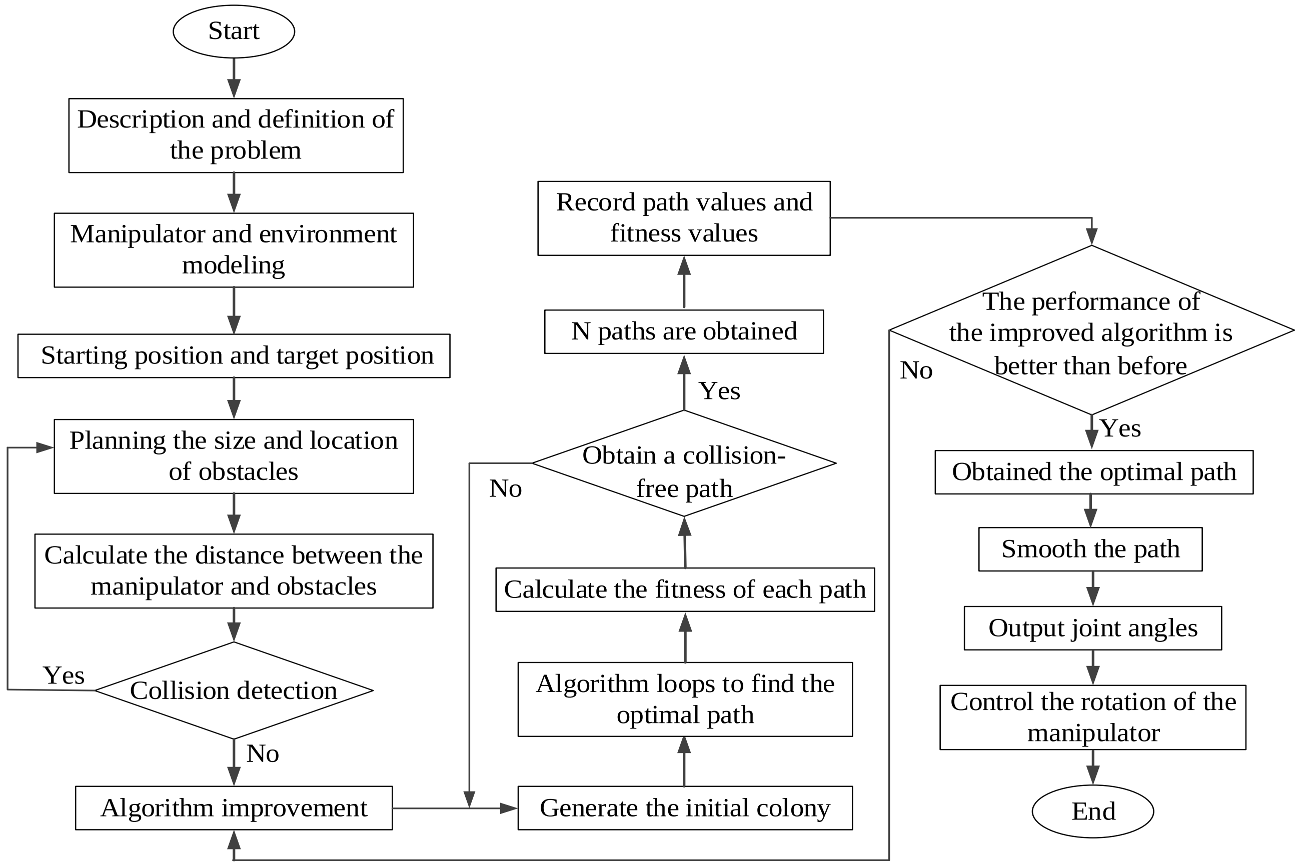
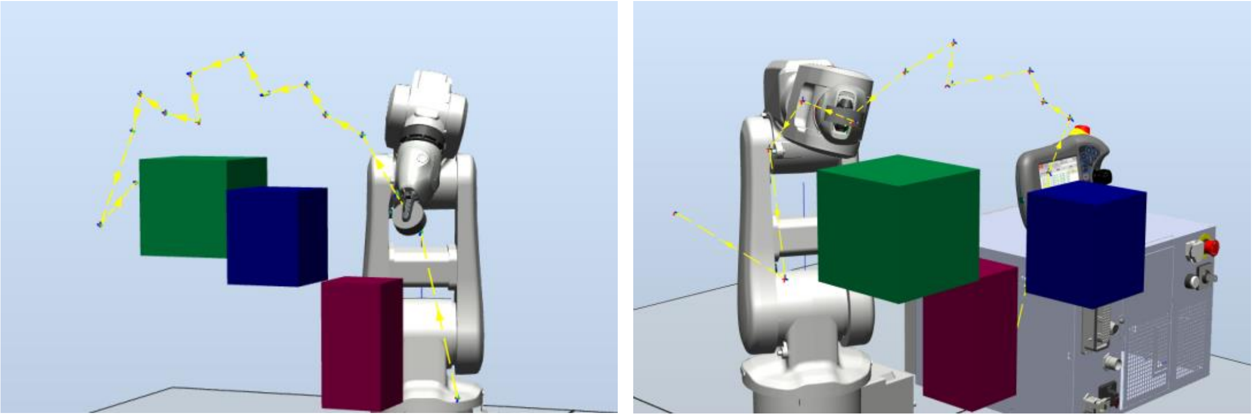


Figure 7. Workflow diagram of the grasping manipulator.

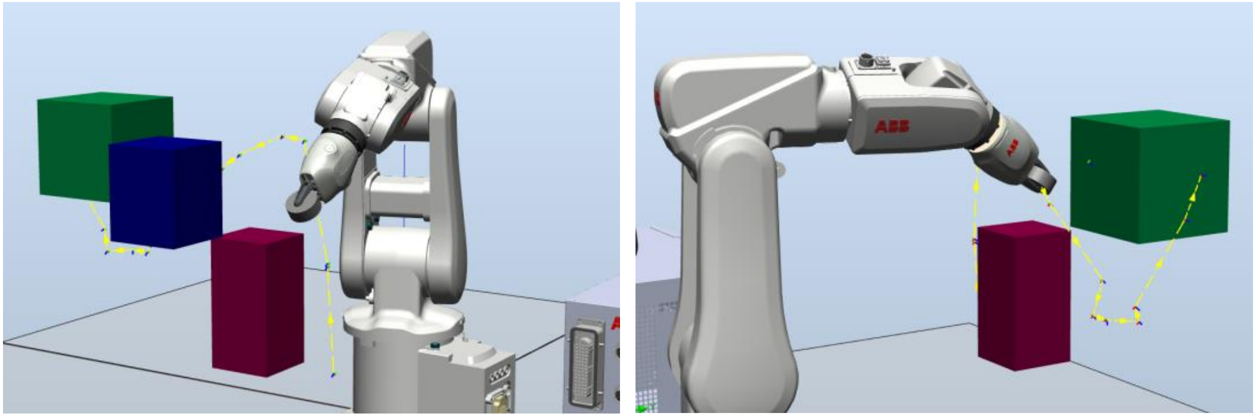
Figure 8 shows the experimental procedure using different algorithms. In the experiments, the grasping manipulators safely avoided the obstacles and successfully guided the end-effector to the target point. Ten repeated experiments were carried out for each algorithm, in which the connecting rod and the end gripper of the manipulator avoided obstacles, the planned paths are within the maximum operating range of the mechanical arm, and the trajectory of each experiment is basically consistent with the simulation trajectory.

Inverse kinematics is the process of solving for all joint variables $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ given the pose of the end of the manipulator. Figure 9 shows the variation of each joint angle during obstacle avoidance for the three algorithms. It is observed that the variation of each joint in the ACO and EACO algorithms is larger than that in the ESACO algorithm, which indicates the relatively stable motion of the mechanical arm in the ESACO algorithm.

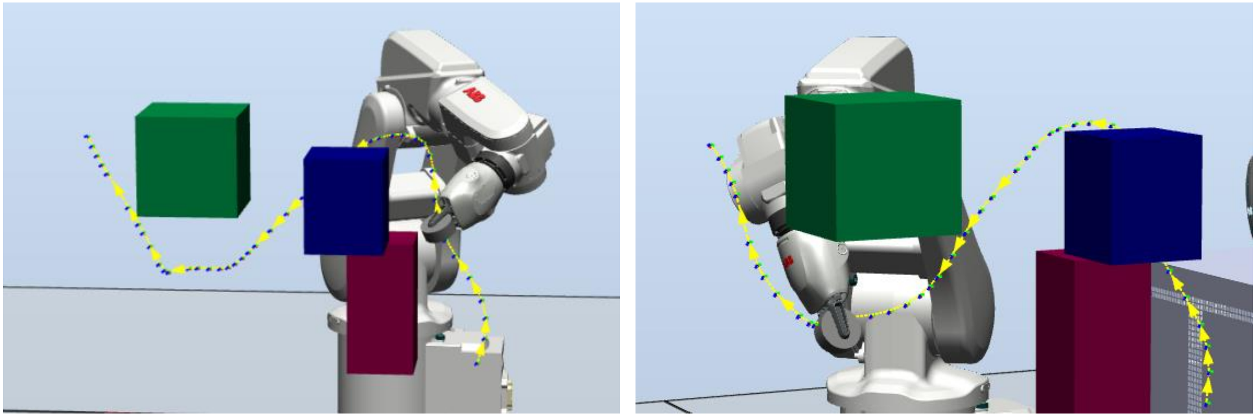
Figure 10 shows the total energy variation of the grasping mechanical arm, indicating that the manipulator completes the path in 3.576 s with total motor energy of 524.998 J when using the ACO algorithm, 2.784 s with total motor energy of 350.803 J when using the EACO algorithm, and 2.664 s with total motor energy of 319.184 J when using the ESACO algorithm. This shows that among the studied algorithms, ESACO can complete the path in the shortest time and the least energy consumption during the assembly process with good symmetry.



(**a**) Obstacle avoidance path planned using the ACO algorithm

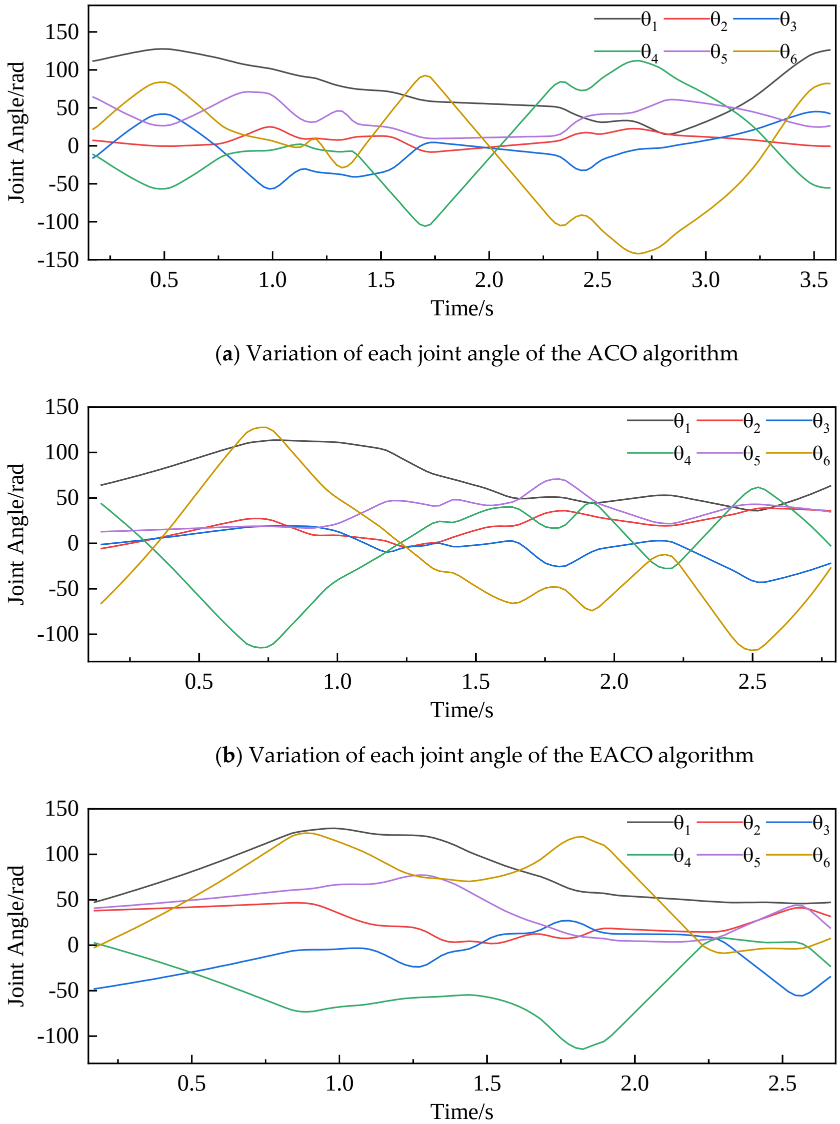


(**b**) Obstacle avoidance path planned using the EACO algorithm



(**c**) Obstacle avoidance path planned using the ESACO algorithm

Figure 8. Obstacle avoidance paths using ACO, EACO, and ESACO algorithms.



(**a**) Variation of each joint angle of the ACO algorithm

(**b**) Variation of each joint angle of the EACO algorithm

(**c**) Variation of each joint angle of the ESACO algorithm

Figure 9. Variation of each joint angle for the three algorithms.

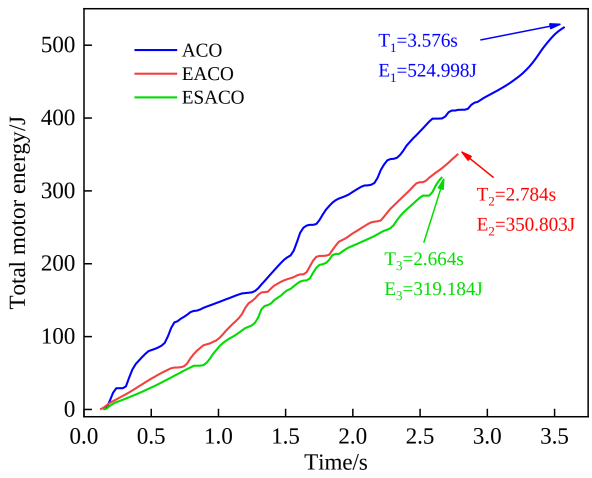
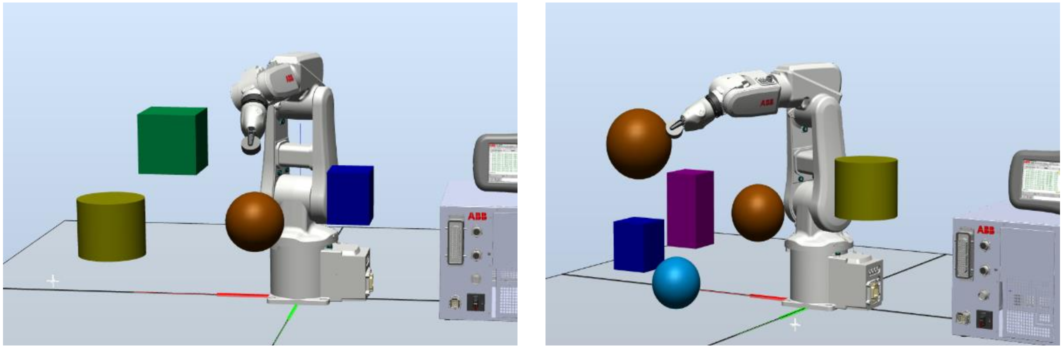


Figure 10. Total energy change of grasping mechanical arm.

5.3. Experiments in Different Scenarios

In order to further verify the effectiveness of the proposed algorithm for obstacle avoidance path planning of the grasping manipulator, two scenarios with different shapes and different obstacles are set up. Then each algorithm is tested 15 times. Figure 11a,b shows the scenario with four and six obstacles, respectively. Table 7 shows the experimental results for different scenarios.



(**a**) Scenario 1: Four obstacles (**b**) Scenario 2: Six obstacles

Figure 11. Scenarios with different shapes and different number of obstacles.

Table 7. Simulation results in different scenarios.

	Method	Min. Path Length/cm	Average Running Time/s	Successful Times	Total Energy/J
Scenario 1	ACO	67.354	8.012	11	726.298
	EACO	52.681	6.584	12	668.594
	ESACO	49.323	5.746	14	596.261
Scenario 2	ACO	89.252	8.623	12	823.154
	EACO	80.541	7.258	13	769.329
	ESACO	68.422	6.567	14	630.468

It is found that applying the ESACO algorithm reduces the shortest path length and search time in Scenario 1 by 27% and 28%, respectively. In this case, the successful times of experiments are 14, and the total energy consumption of the robotic arm is reduced by 20%. Moreover, the path length and search time in Scenario 2 under the ESACO algorithm was reduced by 24% and 24%, respectively. In this scenario, the number of successful experiments was 14, and the total energy consumption of the robotic arm was reduced by 23%. The results show that the ESACO algorithm can effectively speed up the search efficiency in different scenarios of obstacles, and the manipulator has high accuracy and strong reliability during the experiments.

Modern optimization techniques such as genetic algorithm (GA), particle swarm optimization (PSO), gray wolf optimization (GWO), and whale optimization algorithm (WOA) are selected to compare with the algorithms proposed in this paper, and their advantages and disadvantages are evaluated by optimization performance and robustness. The optimization performance is calculated by Equation (20), and the closer the value is to 0, the better the optimization performance. Robustness is calculated by Equation (21), which measures how close the algorithm is to the optimal solution, and the smaller the value, the higher the robustness.

$$E_m = (L_a - L_{min})/L_a \times 100\% \quad (20)$$

where L_{min} is the optimal path value for the simulation, and L_a is the average value of the path obtained by running 15 times.

$$E_f = (N_e \times T)/N_{max} \times 100\% \tag{21}$$

where N_e is the number of iterations when the path is optimal, T is the running time, and N_{max} is the maximum number of iterations.

Simulation results for maps of Scenario 1 and Scenario 2 are shown in Table 8 below. It can be seen that the ESACO algorithm and EACO algorithm have better optimization performance than GA, PSO, GWO, and WAO. ESACO algorithm has the best performance and can obtain the global optimal solution. The robustness of the ESACO algorithm is also better than other algorithms, indicating that the algorithm has the best stability.

Table 8. Simulation results in different scenarios.

Method	Scenario 1		Scenario 2	
	Optimization Performance	Robustness	Optimization Performance	Robustness
GA	0.169	2.943	0.203	2.477
PSO	0.198	1.781	0.192	1.449
GWO	0.122	1.154	0.216	2.396
WOA	0.167	1.352	0.178	2.134
EACO	0.017	1.069	0.141	1.022
ESACO	0.094	0.536	0.109	1.006

6. Conclusions

Based on the performed analyses, the main achievement and conclusions of the present study can be summarized as follows.

(1) In this study, an elite smoothing ant colony algorithm is proposed to plan the path of the grasping manipulator. Firstly, the probability transfer formula and pheromone update strategy are improved to enhance the reliability of the algorithm and the flexibility of the path. Then, the B-sample curve is designed to eliminate the fold points and generate collision-free smooth paths. The proposed algorithm improves the path quality and planning efficiency.

(2) The different metrics of the three algorithms are analyzed and compared. The results show that in the simple environment, the shortest path length is optimized by 22.1%, the running time is shortened by 35%, and the collision-free number is increased by 4 times. Similarly, in different obstacle scenarios, the path length and running time of the ESACO algorithm are increased by more than 20%, and the number of successful experiments is 14 times, indicating the feasibility and symmetry of the ESACO algorithm.

(3) The experimental validation of the obstacle avoidance path of the grasping manipulator is carried out. The results show that the ESACO algorithm can guide the manipulator to avoid obstacles to reach the target point with minimal time and energy consumption, which provides reliable support for the grasping mechanical arm to rapidly assemble equipment in complex environments. In future work, the motion control of the manipulator during the path tracking process will be investigated in detail to achieve higher accuracy.

Author Contributions: Conceptualization, X.M.; methodology, X.M.; software, X.M.; validation, X.M.; data curation, X.M.; writing—original draft preparation, X.M.; writing—review and editing, X.M.; visualization, X.Z.; supervision, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (grant no. 51975540).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: This research was supported by the Shanxi Key Laboratory of Advanced Manufacturing Technology.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Guo, X.; Peng, G.; Meng, Y. A modified Q-learning algorithm for robot path planning in a digital twin assembly system. *Int. J. Adv. Manuf. Technol.* **2022**, *119*, 3951–3961. [CrossRef]
2. Liu, Y.; Er, M.J.; Guo, C. Online time-optimal path and trajectory planning for robotic multipoint assembly. *Assem. Autom.* **2021**, *41*, 601–611. [CrossRef]
3. Kobayashi, M.; Motoi, N. Local Path Planning: Dynamic Window Approach with Virtual Manipulators Considering Dynamic Obstacles. *IEEE Access* **2022**, *10*, 17018–17029. [CrossRef]
4. Wang, H.; Chen, W. Multi-Robot Path Planning with Due Times. *IEEE Robot. Autom. Lett.* **2022**, *7*, 4829–4836. [CrossRef]
5. Chai, X.; Gao, F.; Qi, C.K.; Pan, Y.; Xu, Y.L.; Zhao, Y. Obstacle avoidance for a hexapod robot in unknown environment. *Sci. China Technol. Sci.* **2017**, *60*, 818–831. [CrossRef]
6. Wang, X.; Zhou, X.; Xia, Z.; Gu, X. A survey of welding robot intelligent path optimization. *J. Manuf. Process.* **2021**, *63*, 14–23. [CrossRef]
7. Xu, L.; Cao, M.; Song, B. A new approach to smooth path planning of mobile robot based on quartic Bezier transition curve and improved PSO algorithm. *Neurocomputing* **2022**, *473*, 98–106. [CrossRef]
8. Patle, B.K.; Babu, L.G.; Pandey, A.; Parhi, D.R.K.; Jagadeesh, A. A review: On path planning strategies for navigation of mobile robot. *Def. Technol.* **2019**, *15*, 582–606. [CrossRef]
9. Li, Z.; Dankelman, J. Path planning for endovascular catheterization under curvature constraints via two-phase searching approach. *Int. J. Comput. Assist. Radiol. Surg.* **2021**, *16*, 619–627. [CrossRef]
10. Wang, W.-C.; Ng, C.-Y.; Chen, R. Vision-Aided Path Planning Using Low-Cost Gene Encoding for a Mobile Robot. *Intell. Autom. Soft Comput.* **2022**, *32*, 991–1006. [CrossRef]
11. Zhang, L.; Zhang, Y.J.; Li, Y.F. Mobile Robot Path Planning Based on Improved Localized Particle Swarm Optimization. *IEEE Sens. J.* **2021**, *21*, 6962–6972. [CrossRef]
12. Liu, J.H.; Yang, J.G.; Liu, H.P.; Tian, X.J.; Gao, M. An improved ant colony algorithm for robot path planning. *Soft Comput.* **2017**, *21*, 5829–5839. [CrossRef]
13. Yang, L.; Fu, L.; Li, P.; Mao, J.; Guo, N.; Du, L. LF-ACO: An effective formation path planning for multi-mobile robot. *Math. Biosci. Eng.* **2022**, *19*, 225–252. [CrossRef]
14. Luo, M.; Hou, X.R.; Yang, S.X. A Multi-Scale Map Method Based on Bioinspired Neural Network Algorithm for Robot Path Planning. *IEEE Access* **2019**, *7*, 142682–142691. [CrossRef]
15. Ma, N.; Wang, J.; Liu, J.; Meng, M.Q.H. Conditional Generative Adversarial Networks for Optimal Path Planning. *IEEE Trans. Cogn. Dev. Syst.* **2022**, *14*, 662–671. [CrossRef]
16. Qi, J.; Yang, H.; Sun, H. MOD-RRT*: A Sampling-Based Algorithm for Robot Path Planning in Dynamic Environment. *IEEE Trans. Ind. Electron.* **2021**, *68*, 7244–7251. [CrossRef]
17. Lyu, D.; Chen, Z.; Cai, Z.; Piao, S. Robot path planning by leveraging the graph-encoded Floyd algorithm. *Future Gener. Comput. Syst.* **2021**, *122*, 204–208. [CrossRef]
18. Deng, W.; Zhao, H.M.; Zou, L.; Li, G.Y.; Yang, X.H.; Wu, D.Q. A novel collaborative optimization algorithm in solving complex optimization problems. *Soft Comput.* **2017**, *21*, 4387–4398. [CrossRef]
19. Yu, J.; You, X.M.; Liu, S. Ant colony algorithm based on magnetic neighborhood and filtering recommendation. *Soft Comput.* **2021**, *25*, 8035–8050. [CrossRef]
20. Miao, C.W.; Chen, G.Z.; Yan, C.L.; Wu, Y.Y. Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm. *Comput. Ind. Eng.* **2021**, *156*, 107230. [CrossRef]
21. Sangeetha, V.; Krishankumar, R.; Ravichandran, K.S.; Cavallaro, F.; Kar, S.; Pamucar, D.; Mardani, A. A Fuzzy Gain-Based Dynamic Ant Colony Optimization for Path Planning in Dynamic Environments. *Symmetry* **2021**, *13*, 280. [CrossRef]
22. Jiao, Z.Q.; Ma, K.; Rong, Y.L.; Wang, P.; Zhang, H.K.; Wang, S.H. A path planning method using adaptive polymorphic ant colony algorithm for smart wheelchairs. *J. Comput. Sci.* **2018**, *25*, 50–57. [CrossRef]
23. Mavrovouniotis, M.; Muller, F.M.; Yang, S.X. Ant Colony Optimization with Local Search for Dynamic Traveling Salesman Problems. *IEEE Trans. Cybern.* **2017**, *47*, 1743–1756. [CrossRef] [PubMed]
24. Jin, Q.; Tang, C.; Cai, W. Research on Dynamic Path Planning Based on the Fusion Algorithm of Improved Ant Colony Optimization and Rolling Window Method. *IEEE Access* **2022**, *10*, 28322–28332. [CrossRef]
25. Xiong, C.K.; Chen, D.F.; Lu, D.; Zeng, Z.; Lian, L. Path planning of multiple autonomous marine vehicles for adaptive sampling using Voronoi-based ant colony optimization. *Robot. Auton. Syst.* **2019**, *115*, 90–103. [CrossRef]
26. Yang, H.; Qi, J.; Miao, Y.C.; Sun, H.X.; Li, J.H. A New Robot Navigation Algorithm Based on a Double-Layer Ant Algorithm and Trajectory Optimization. *IEEE Trans. Ind. Electron.* **2019**, *66*, 8557–8566. [CrossRef]

27. Zhang, H.; Yang, S.W. Smooth path and velocity planning under 3D path constraints for car-like vehicles. *Robot. Auton. Syst.* **2018**, *107*, 87–99. [CrossRef]
28. Rath, A.K.; Parhi, D.R.; Das, H.C.; Muni, M.K.; Kumar, P.B. Analysis and use of fuzzy intelligent technique for navigation of humanoid robot in obstacle prone zone. *Def. Technol.* **2018**, *14*, 677–682. [CrossRef]
29. Sun, Z.; Shao, Z.F.; Li, H. An eikonal equation based path planning method using polygon decomposition and curve evolution. *Def. Technol.* **2020**, *16*, 1001–1018. [CrossRef]
30. Patle, B.K.; Pandey, A.; Jagadeesh, A.; Parhi, D.R. Path planning in uncertain environment by using firefly algorithm. *Def. Technol.* **2018**, *14*, 691–701. [CrossRef]
31. Chen, S.; Li, Z.; Lin, Y.; Wang, F.; Cao, Q. Automatic ultrasound scanning robotic system with optical waveguide-based force measurement. *Int. J. Comput. Assist. Radiol. Surg.* **2021**, *16*, 1015–1025. [CrossRef]
32. Dai, P.; Yao, S.; Li, Z.; Zhang, S.; Cao, X. ACE: Anchor-Free Corner Evolution for Real-Time Arbitrarily-Oriented Object Detection. *IEEE Trans. Image Process.* **2022**, *31*, 4076–4089. [CrossRef]
33. Qi, X.; Gan, Z.; Liu, C.; Xu, Z.; Zhang, X.; Li, W. Collective intelligence evolution using ant colony optimization and neural networks. *Neural Comput. Appl.* **2021**, *33*, 12721–12735. [CrossRef]
34. Luo, Q.; Wang, H.B.; Zheng, Y.; He, J.C. Research on path planning of mobile robot based on improved ant colony algorithm. *Neural Comput. Appl.* **2020**, *32*, 1555–1566. [CrossRef]
35. Majeed, A.; Abbas, M.; Sittar, A.A.; Misro, M.Y.; Kamran, M. Airplane designing using Quadratic Trigonometric B-spline with shape parameters. *AIMS Math.* **2021**, *6*, 7669–7683. [CrossRef]
36. Elbanhawi, M.; Simic, M.; Jazar, R. Improved manoeuvring of autonomous passenger vehicles: Simulations and field results. *J. Vib. Control* **2017**, *23*, 1954–1983. [CrossRef]
37. Cao, X.M.; Zou, X.J.; Jia, C.Y.; Chen, M.Y.; Zeng, Z.Q. RRT-based path planning for an intelligent litchi-picking manipulator. *Comput. Electron. Agric.* **2019**, *156*, 105–118. [CrossRef]
38. Liao, S.L.; Zhu, R.M.; Wu, N.Q.; Shaikh, T.A.; Sharaf, M.; Mostafa, A.M. Path planning for moving target tracking by fixed-wing UAV. *Def. Technol.* **2020**, *16*, 811–824. [CrossRef]
39. Urrea, C.; Kern, J. Design and implementation of a wireless control system applied to a 3-DoF redundant robot using Raspberry Pi interface and User Datagram Protocol. *Comput. Electr. Eng.* **2021**, *95*, 107424. [CrossRef]
40. Siewe, R.T.; Domguia, U.S.; Woafo, P. Generation of pulse-like and bursting-like oscillations from nonlinear systems using embedded technologies and applications to excite mechanical arms. *Commun. Nonlinear Sci. Numer. Simul.* **2019**, *69*, 343–359. [CrossRef]
41. Ohkawa, T.; Yamashina, K.; Kimura, H.; Ootsu, K.; Yokota, T. FPGA Components for Integrating FPGAs into Robot Systems. *IEICE Trans. Inf. Syst.* **2018**, *E101D*, 363–375. [CrossRef]