


RESEARCH ARTICLE

# Path planning for spot welding robots based on improved ant colony algorithm

Yuesheng Tan\*, Jie Ouyang , Zhuo Zhang, Yinglun Lao and Pengju Wen

School of Technology, Beijing Forestry University, Beijing 100083, China

\*Corresponding author. E-mail: [tanyuesheng@163.com](mailto:tanyuesheng@163.com)

**Received:** 13 March 2022; **Revised:** 12 May 2022; **Accepted:** 14 July 2022; **First published online:** 15 August 2022

**Keywords:** path planning, ant colony algorithm, particle swarm algorithm, spot welding robot

## Abstract

A welding path can be planned effectively for spot welding robots using the ant colony algorithm, but the initial parameters of the ant colony algorithm are usually selected through human experience, resulting in an unreasonable planned path. This paper combines the ant colony algorithm with the particle swarm algorithm and uses the particle swarm algorithm to train the initial parameters of the ant colony algorithm to plan an optimal path. Firstly, a mathematical model for spot welding path planning is established using the ant colony algorithm. Then, the particle swarm algorithm is introduced into the ant colony algorithm to find the optimal combination of parameters by treating the initial parameters  $\alpha$  and  $\beta$  of the ant colony algorithm and as two-dimensional coordinates in the particle swarm algorithm. Finally, the simulation analysis was carried out using MATLAB to obtain the paths of the improved ant colony algorithm for six different sets of parameters with an average path length of 10,357.7509 mm, but the average path length obtained by conventional algorithm was 10,830.8394 mm. Convergence analysis of the improved ant colony algorithm showed that the average number of iterations was 17. Therefore, the improved ant colony algorithm has higher solution quality and converges faster.

## 1. Introduction

Robots are widely used in welding production. Although this has improved productivity, there are some common problems, one of which is the poor path planning of spot welding robots on welding workstations [1]. In practice, the path planning for spot welding robots is developed by the relevant process personnel based on their own experience, and the robot is given task instructions using online teaching which is tedious. Therefore, it is important to use algorithms to plan a reasonable welding path. The mainstream path planning algorithms include artificial potential field method, ant colony algorithm, RRT algorithm, and particle swarm algorithm [2, 3, 4, 5]. Path planning for robot motion has been studied both at home and abroad. Christian Wurll presents results from the Karlsruhe University on path planning for a six-degree-of-freedom robot, which discretely stratifies Cartesian space to calculate distances and uses the A\* search algorithm to achieve finding the shortest distance between multiple weld points, but the setting of its valuation function can seriously affect the quality and efficiency of the solution [6]. Pei Chengsong proposed an improved cuckoo search algorithm based on compact and parallel techniques. This algorithm can better deal with the problem of 3D path planning, but there are still problems such as insufficient accuracy and long paths [7]. Liu Lisang proposed a new fusion algorithm of jump-A\* algorithm and dynamic window approach to meet the performance requirements of global optimization and path smoothness, but the algorithm takes more time [8]. Ma Haonan used improved artificial potential field method to overcome the shortcoming of local minima and realized the path planning of the dual-manipulator system, but the algorithm cannot overcome the issue of unreachable target points [9]. Zhang Zhen used the RRT algorithm based on constraint sampling to plan the path of the robot arm, which improved the search efficiency and smoothed the trajectory, but the algorithm cannot

obtain the shortest path [10]. XW Wang analyzed and compared the characteristics of the genetic algorithm and the particle swarm algorithm, and the simulation results showed that both algorithms could be used to solve small-scale path planning in welding tasks. Finally, A combination of the genetic algorithm and the particle swarm algorithm was proposed, which had the effect of avoiding the algorithm from falling into a local optimum [11]. Wang Jiahai analyzed the actual welding process and compared the convergence speed and optimization accuracy of simulated annealing, the ant colony algorithm (ACS), and the genetic algorithm and proved that the ACS is more effective than the others [12]. Lin Juguang used the traditional ant colony algorithm to plan the welding path of the body-in-white base plate, and the simulation results in DELMIA showed that the optimized path was more efficient than the traditional path, but in-depth study of the ant colony algorithm was not carried out [13]. It is found that combining other algorithms with the ant colony algorithm or introducing adaptive strategies can improve the adaptability and effectiveness of the ant colony algorithm [14, 15, 16, 17].

The ant colony algorithm shows strong superiority over other algorithms in terms of its global search capability and good robustness, but the performance of its algorithm is influenced by initial parameters [18, 19]. In this paper, particle swarm algorithm is proposed to train the initial parameters of the ant colony algorithm to obtain the optimal initial parameters for improving the solution quality and efficiency, and the influence of the initial parameters of the ant colony algorithm on the algorithm itself is analyzed. Finally, MATLAB is used to conduct simulation to verify the feasibility and superiority of the improved algorithm.

## 2. Implementation of the ant colony algorithm

The ants secrete pheromone as they walk which acts as a message transmitter, and other ants can sense the strength of the pheromone as they pass by and use it to guide themselves [20].

### 2.1. Rules for the movement of ants between solder joints

The corresponding mathematical model is developed for the spot welding path planning problem based on the ant colony algorithm. Let the number of ants in the whole ant population be  $m$  and the number of solder joints is  $n$ . The distance between solder joint  $i$  and solder joint  $j$  is  $d_{ij}(i, j = 1, 2, \dots, n)$ , and the pheromone concentration of the path between solder joint  $i$  and solder joint  $j$  is  $\tau_{ij}(t)$ . The ants are placed at different solder joints at the initial moment, and the pheromone concentration on all connected paths between solder joints remains the same  $\tau_{ij}(0) = \tau_0$ . Then, devising a probability function for the ant's choice between the different solder joints  $p_{ij}^k(t)$  denotes the probability of an ant choosing at moment  $t$  from solder joint  $i$  to solder joint  $j$ .

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}(t)]^\beta}{\sum_{s \in allow_k} [\tau_{is}(t)]^\alpha \cdot [\eta_{is}(t)]^\beta}, j \in allow_k \\ 0, j \notin allow_k \end{cases} \tag{1}$$

where  $\eta_{ij}(t)$  – The degree of expectation that ant from the solder joint  $i$  to  $j$  the weld point, as a heuristic function

$allow_k$  – The set of all solder joints to be processed,  $k = 1, 2, \dots, n$

$\alpha$  – Pheromone importance factor

$\beta$  – Significance factor of the heuristic function

$\rho$  – Degree of volatility of pheromones.

### 2.2. Pheromone updates program design

When the ant has traversed all the solder joints once, the concentration of information connecting the individual solder joints to each other is

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

where  $\Delta \tau_{ij}^k$  – The k ant adds pheromones to the solder joint i and solder joint pheromone added to the channel

$\Delta \tau_{ij}$  – All ants in the incremental pheromone release on the channel between the solder joints  
 $\Delta \tau_{ij}^k$  is calculated by the following method.

$$\Delta \tau_{ij}^k = \begin{cases} \frac{Q}{L_k}, & \text{if an ant visits solder joint j from solder joint i} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where  $Q$  – The total amount of pheromones released by the ant in one cycle  
 $L_k$  – The total length of the path traveled by the first ant.

### 2.3. Objective function and implementation steps

In the path planning of the spot welding industrial robot, the order of access to  $n$  process points, which is  $V = \{v_1, v_2, v_3, \dots, v_n\}$ , is  $T = \{t_1, t_2, t_3, \dots, t_n\}$ , where  $t_i \in V, i = (1, 2, 3, \dots, n)$ , and the corresponding three-dimensional spatial coordinates of  $t_i$  is  $(x, y, z)$ . The objective function is the distance of the spot welding robot gun movement, whose mathematical expression is

$$\min L = \sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} \quad (4)$$

where  $L$  is a closed path to access all solder joints.

Figure 1 shows the flow chart of the program structure of the basic ant colony algorithm, and the basic steps of the algorithm are as follows:

1. At the beginning of the calculation, the relevant parameters are initialized, the ant colony model (number of ants)  $m$ , the pheromone factor  $\alpha$ , the importance factor of the heuristic function  $\beta$ , the volatility factor of the information system  $\eta$ , the total number of pheromone constants  $Q$ , the maximum number of iterations  $iter\_max$ , and the initial value of the iterations  $iter = 1$ .
2. Increasing the number of iterations.
3. Initializing the taboo table and placing the ant's current location in the taboo table.
4. Number of ants  $k = k + 1$ .
5. Calculation of the transfer probability function  $p_{ij}^k(t)$ .
6. Modifying the contraindication form.
7. Continuing with the fourth iteration step if the current number of ants is less than the total number of ants, otherwise, continuing updating the pheromone.
8. The loop ends and the result of the calculation is output when the number of iterations  $iter\_max$  is reached. The forbidden table is reinitialized and the loop continues if it is less than the number of iterations.

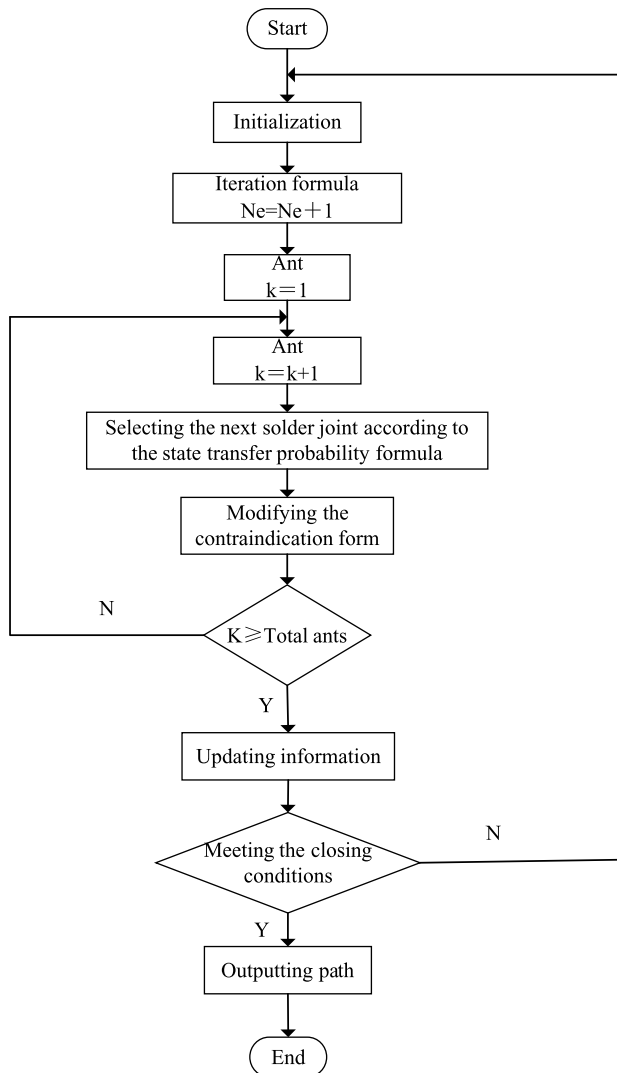


Figure 1. Ant colony algorithm flow chart.

### 3. Improvement of ant colony algorithm based on particle swarm idea

The ACO shows strong robustness and good search capability in dealing with TSP problems but has some limitations. The solution performance of the ACO mainly depends on the pheromone factor  $\alpha$  and the heuristic function factor  $\beta$ , which are usually determined through extensive experiments or manual experience. To avoid the problem of long solution time or poor solution quality due to improper parameter selection, this paper adopts a strategy of applying particle swarm ideas to optimize the  $\alpha$  and  $\beta$  parameters of the ant colony algorithm. The particle swarm algorithm can quickly approximate the optimal solution [21], which provides a reasonable basis for the selection of parameters. Thus, it improves the solution performance of the ant colony system. Figure 2 shows the flow chart of the improved ant colony algorithm, and the basic steps of the algorithm are shown as above.

#### 3.1. Particle update program

The particle swarm algorithm originates from the foraging behavior of a flock of birds [22], in which each bird is one of the particles in the process of searching for food. In a  $D$  dimensional target search space,

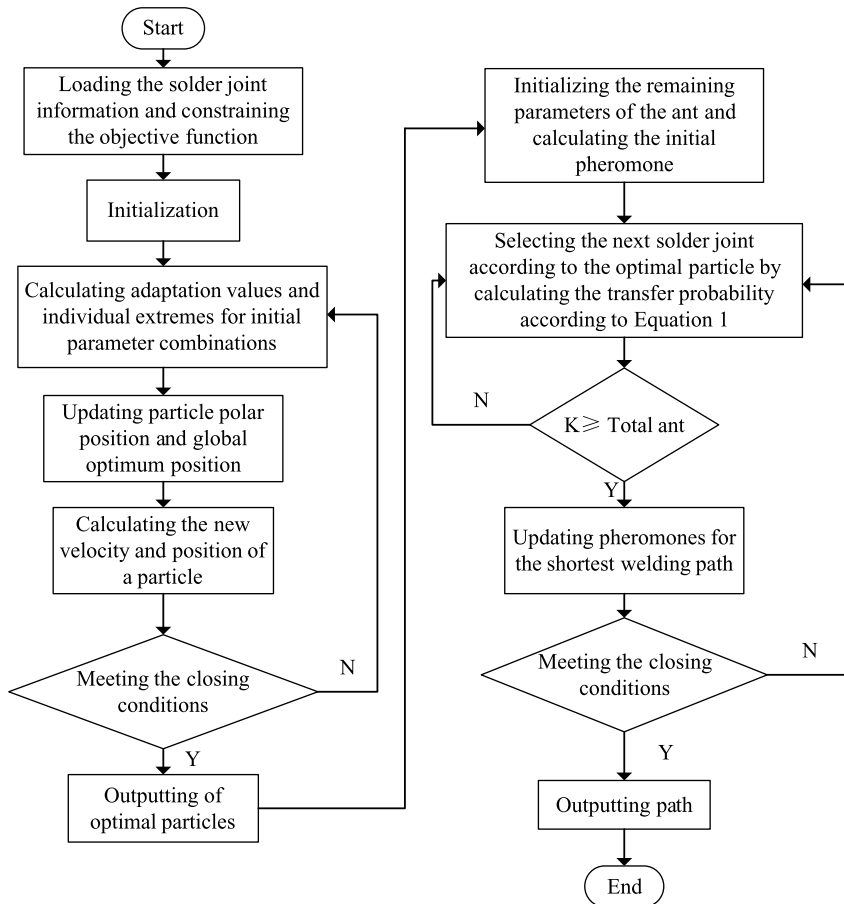


Figure 2. Algorithm flow chart.

provided that there are  $m$  particles, the coordinate position of the  $i$  particle in the search space is  $X_i = (P_i^1, P_i^2, \dots, P_i^D)$ , each particle is a potential solution of the objective function and will be brought into the requested objective function to obtain the adaptation value of each particle, and the particle's local optimal position is noted as  $P_i = (P_i^1, P_i^2, \dots, P_i^D)$ . The velocity of the  $i$  particle is  $V_i = (v_i^1, v_i^2, \dots, v_i^D)$ .

The particle is updated continuously at its location using the following equation.

$$v_i^d = \omega v_i^d + c_1 r_1 (p_i^d - x_i^d) + c_2 r_2 (p_g^d - x_i^d) \tag{5}$$

$$x_i^d = x_i^d + \lambda v_i^d \tag{6}$$

where  $i = 1, 2, \dots, m$ ;  $d = 1, 2, \dots, D$ ;  $\omega \geq 0$  is the inertia factor. The acceleration constant is  $c_1 \geq 0, c_2 \geq 0$ .  $r_1$  and  $r_2$  are a random number in the range  $[0, 1]$ .  $\lambda$  is the constraint factor for the purpose of velocity control of the particle. In addition,  $v_i^d \in [-v_{\max}^d, v_{\max}^d]$ , the maximum velocity of the particle flying  $V_i$ , is limited by the maximum velocity. The maximum velocity is indicated by  $(v_{\max}^1, v_{\max}^2, \dots, v_{\max}^D)$ . When the velocity of a particle in one dimension exceeds  $v_{\max}^d$ , the velocity of that dimension is limited at this point  $v_{\max}^d$ . The condition of termination of the iteration is that the set maximum number of iterations is reached or the particle population is searched for the optimal solution at the moment to satisfy the minimum permissible error of the objective function.

### 3.2. Improvement and implementation of ant colony algorithm

#### 3.2.1. Algorithm improvement scheme

First, the certain number of particles are initialized with the coordinates of the particles in two-dimensional space  $(\alpha, \beta)$ . Each position of a particle is a potential target solution for parameter optimization. The corresponding parameters of each particle are fed back to the ant colony algorithm. The individual particle adaptation values are counted according to the objective function to determine the degree of superiority or inferiority of the particles. Then, the positions and velocities of the particles are updated according to Eqs. (5) and (6), and the iteratively updated particles call the ant colony algorithm again to measure the adaptation values of the particles. The above steps are cycled until the number of iterations reaches 20 or when the difference between the optimal solutions of two adjacent generations of particles is less than  $\sigma$ , terminating the cycle. The optimal particle position  $P_g(\alpha, \beta)$  is the optimal initial parameter in the ant colony algorithm.

#### 3.2.2. Improved algorithm design

##### 1. Initial particle swarm parameter design

In this paper, the solution variables are two dimensional. In this two-dimensional target search space, the particle swarm is initialized by setting the initial particle coordinate values to random.

The initial position and velocity of the  $i$  particle are

$$p[i][j] = r_1 \times (x_{max} - x_{min}) + x, j \in [0, 1] \tag{7}$$

$$V[i][j] = r_2 \times x_{max}, j \in [0, 1] \tag{8}$$

where  $r_1$  and  $r_2$  are the random numbers of the variation of the range  $[0, 1]$ .

Set the values of the basic particle swarm parameters as follows:

$$\left\{ \begin{array}{l} number = 40 \\ generation = 20 \\ x_{min} = 0.5 \\ x_{max} = 10.0 \\ v_{min} = 0 \\ v_{max} = 1.0 \\ \omega = 0.8 \\ c_1 = 2 \\ c_2 = 2 \\ \sigma = 0.001 \end{array} \right.$$

where *number* is the size of the particle swarm and taken 20–40 as a general value. The 30 particles are sufficient for most problems, but the 100–200 particles are needed for special puzzles. *generation* is the number of iterations.  $c_1$  and  $c_2$  are the acceleration constants, representing the weight of the particles' own experience and social experience, respectively.  $\omega$  is the inertia factor, which defines the particle's flight range. The larger the flight range, the stronger the global search ability, unfortunately, the weaker the local search ability.  $x$  is the random variable and is determined by the direction of the dimensional space in which the particle is placed. The velocity range  $[v_{min}, v_{max}]$  and position range  $[x_{min}, x_{max}]$  of each particle place a limit on how far the particle can be transferred and reduce the spatial extent of the particle initialization.

Each position of particle represents a set of parameters for the ant colony algorithm.

$$\begin{cases} \alpha = p[i][0] \\ \beta = p[i][1] \end{cases} \quad (9)$$

## 2. Design of other initial parameters of the ant colony

For the TSP problem, it is generally accepted that the ant cycle system model is the more reasonable of the three models of the ant colony algorithm (ant quantity system, ant density system, and ant cycle system). The solder joints are given as  $n$ , the number of ants is set as  $m = n/2$  in the ant colony algorithm, and the pheromone constant  $Q = 10$  and the iterations of the ant colony algorithm are set as 30.

## 3. Feedback of initial particle values to the ACS

In the random initialization of particles brought into the ant colony algorithm, the objective function is defined according to the TSP problem-solving objective. The adaptation value of each particle is calculated according to the objective function to determine the merit of the current position of the particle. The adaptation value determines the high quality of the combination of parameters it represents.

## 4. Particle information and optimal solution information update scheme

Update the velocity and position of the particle according to the following equation.

$$v[i][j] = \omega \times V[i][j] + c_1 \times r_3 \times (pbest_i - p[i][j]) + c_2 \times r_4 \times (gbest_i - p[i][j]) \quad (10)$$

$$p[i][j] = p[i][j] + v[i][j] \quad (11)$$

In Eq. (10),  $r_3$  and  $r_4$  are random numbers in the range  $[0, 1]$ . The first term of the velocity update is the memory term of the particle, indicating that the update direction is influenced by the velocity magnitude and direction of the last location. The second part is the particle's own experience part, indicating that the particle has been to the point where the adaptation value is optimal. The third part represents the influence brought by the social experience of the particle, reflecting the mutual learning ability between the particles. The superposition of these three components causes the effect of the particles automatically finding the optimal parameters.

The velocity of the particle in two dimensions is updated according to Eq. (5). The updated particle continues to call the ant colony algorithm, which calculates the adaptation value of each particle according to the objective function in the ant colony algorithm to determine the superiority of the particle. Updating the local optimal position and the global optimal position of the particle, and then cycling this process.

## 5. Iteration termination conditions

The algorithm is terminated when the difference between the adaptation values of two consecutive generations of particles does not exceed  $\sigma$  or the number of particle swarm update iterations exceeds 20. The final two-dimensional coordinates of the particle with the best adaptation value are the optimal parameters of the ant colony algorithm.

## 4. MATLAB simulation test and result analysis

In this paper, the algorithm is programmed in MATLAB to verify the effectiveness of the parameters of the ant colony algorithm. First, extensive experiments are conducted to analyze the parameters  $\alpha$  and  $\beta$  values and combinations on the performance of the algorithm. Then, the proposed new algorithm is used to find the combination of high-quality parameters.

**Table I.** The information of side circumference solder joint.

Welded joints	X coordinate	Y coordinate	Z coordinate	Roll	Pitch	Yaw
1	1932.445	-184.187	422.383	-1.571	0.023	0.008
2	2106.053	-219.037	417.685	-1.572	0.021	-0.042
3	2150.443	-224.339	415.809	-1.571	0.022	-0.031
4	2193.201	-233.035	414.451	-1.571	0.026	-0.019
5	2232.299	-236.017	414.154	-1.57	0.036	0.013
6	2276.473	-243.598	415.115	-1.569	0.042	0.05
7	2321.724	-248.795	418.136	-1.565	0.06	0.092
8	2365.399	-254.188	423.177	-1.562	0.061	0.148
9	2415.047	-262.067	431.371	-1.561	0.047	0.196
10	2462.21	-266.027	441.792	-1.568	0.01	0.243
11	2497.213	-269.159	450.943	-1.57	0.002	0.266
12	1611.259	1882.524	1055	-1.571	0	-1.571
13	2511.243	-142.549	453.092	1.561	0.031	-0.236
14	2463.709	-130.864	439.767	1.569	0.008	-0.179
15	2413.256	-122.288	429.041	1.569	0.011	-0.124
16	2345.658	-106.63	420.489	1.571	-0.006	-0.03
17	2295.874	-96.98	417.807	1.571	-0.016	0.02
18	2244.811	-88.393	417.108	1.571	-0.017	0.053
19	2189.468	-80.155	417.599	1.571	-0.019	0.071
20	1939.782	-39.109	423.05	1.57	-0.019	0.087
21	1807.393	-657.074	444.337	1.045	0.042	0.005
22	1971.555	-749.707	420.325	1.048	0.01	-0.041
23	2364.185	-1034.6	435.799	1.565	0.038	0.154
24	1579.661	-90.201	440.998	-3.142	0	0
25	1569.073	-17.1	440.998	-3.142	0	0
26	1571.653	86.19	440.998	-3.142	0	0
27	1574.787	224.345	440.998	3.142	0	0
28	2018.475	-366.138	418.797	-1.571	0.023	0.008
29	2478.012	-269.731	672.751	-1.576	-0.016	0.312
30	2204.496	206.809	422.073	1.571	-0.019	0.071
31	2070.834	206.656	431.584	1.571	-0.019	0.071
32	1912.235	206.474	442.87	1.571	-0.019	0.071
33	1967.969	-45.327	745.761	1.57	-0.019	0.087
34	1231.772	-46.169	810.05	1.57	-0.019	0.087
35	1579.661	-90.201	839.998	-3.142	0	0
36	2171.661	-484.201	839.998	-3.142	0	0
37	1988.47	-565.719	417.646	-1.57	-0.011	-0.036
38	1982.002	-746.42	712.119	-1.57	-0.011	-0.036
39	1751.248	-152.832	946.849	-1.57	-0.011	-0.036

Table I shows the output of the simulation process for the 39 weld joints on the body-in-white side surrounds. X, Y, and Z represent the position information of the weld joints in the table. *Roll*, *Pitch*, *Yaw* are the Euler angles of the weld joint coordinate system. *Roll* represents the angle of rotation of the weld joint coordinate system around the Z-axis in the XOY plane. *Pitch* represents the angle of rotation of the YOZ around the X-axis, and *Yaw* represents the angle of rotation of the XOZ around the Y-axis in the reference coordinate system.

The distribution of solder joints in three dimensions is shown in Fig. 3.



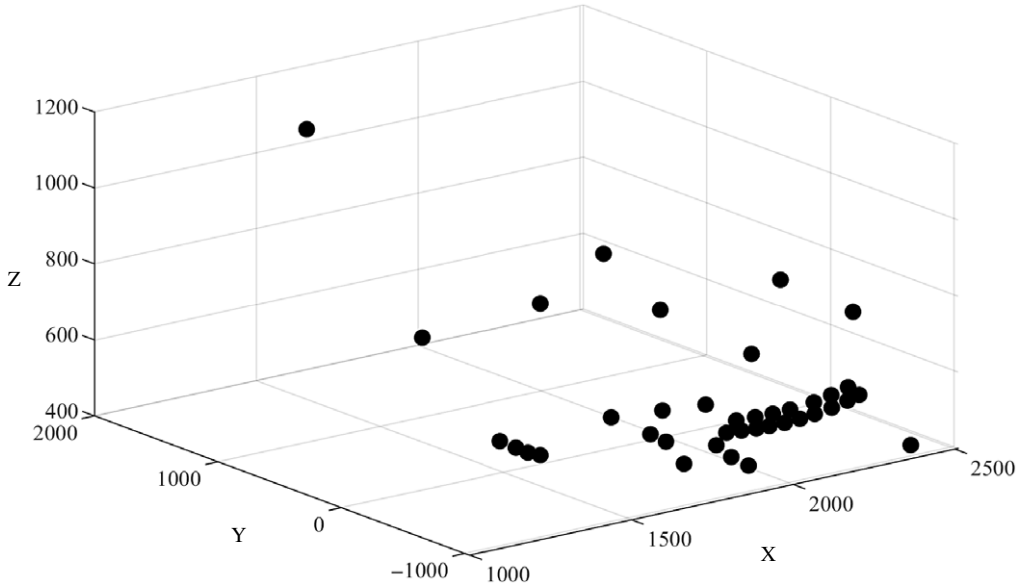


Figure 3. Spatial distribution of welding spot.

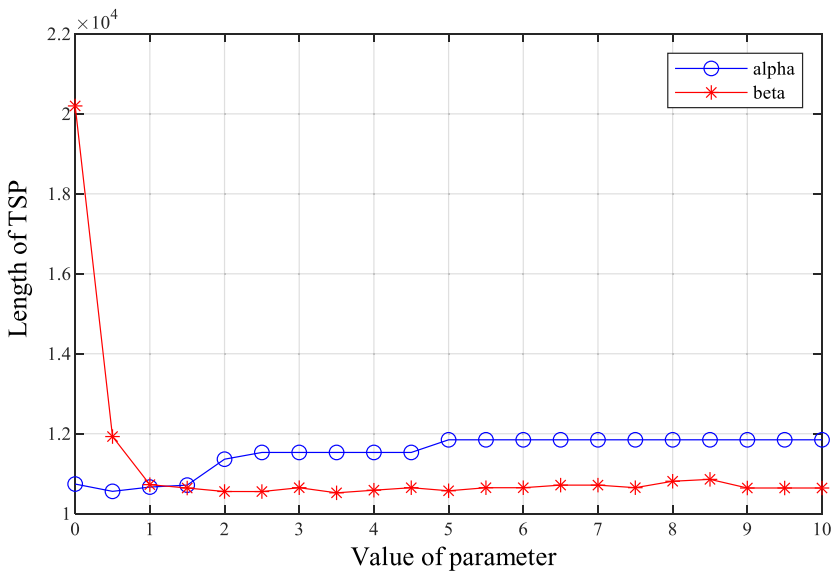


Figure 4. Relationship between  $\alpha$  and  $\beta$  values and optimal path distance.

Holding all other parameter constant,  $m = 75$ ,  $vol = 0.2$ ,  $Q = 10$  as an example, to illustrate  $\alpha$  and  $\beta$  have a huge impact on the performance of the algorithm. Fixing one of  $\alpha$  and  $\beta$  and varying the value of the other, the relationship between parameter variation and optimal path distance can be derived.

As shown in Fig. 4, it is seen that the pheromone importance degree factor  $\alpha < 1$  in the ant colony algorithm has a huge impact on the convergence and solution quality of the algorithm when the other parameters of the current algorithm are fixed. The comprehensive solution performance of the algorithm is better when  $\alpha \in [0.5, 2.5]$ . The influence of the heuristic function importance degree factor  $\beta$  is more obvious in the ant colony algorithm on the convergence speed of the algorithm. The comprehensive performance of the algorithm is better when  $\beta \in [0, 2]$ . The influence of the pheromone factor and the

**Table II.** High-quality parameter combination.

Number	$\alpha$	$\beta$	Path length
1	1.1108	5.5596	10,356.9353
2	0.9372	1.4561	10,357.0823
3	1.3746	2.5438	10,359.9847
4	0.7735	3.9751	10,356.9815
5	0.9755	9.3640	10,358.4784
6	1.0034	5.5273	10,357.0432

**Table III.** Random parameter combination.

Number	$\alpha$	$\beta$	Path length
1	6.9713	4.3702	11,550.9976
2	2.3191	4.6889	10,482.8757
3	3.3748	2.2939	10,452.6463
4	10.0000	8.7376	11,290.9743
5	7.8979	5.2707	10,642.6518
6	9.7741	6.4403	10,564.8908

heuristic function factor on the algorithm was equally evident when the ant colony size  $m$ , pheromone volatility factor  $vol$ , and constant factor  $Q$  were other values.

Table II shows the improved algorithm to find the six sets of  $\alpha$  and  $\beta$  quality combinations of particles and path lengths.

In order to compare the solution performance of the algorithm before and after optimization, the values of  $\alpha$  and  $\beta$  and their combinations are set to random, and path planning is performed with random initial parameters. Table III shows the path lengths obtained using the six sets of random initial parameters.

The average path length obtained by the six sets of initial parameters after optimization is 10,357.7509 mm, and the average distance of the six groups of path lengths obtained by the algorithm before optimization is 10,830.8394 mm. The path distance before optimization is reduced by 473.0885 mm compared with that after optimization, which indicates that the solution quality of the algorithm after optimization has been improved.

A set of parameters  $\alpha = 1.1108$ ,  $\beta = 5.5596$ , which obtained from the particle swarm training, is fed back to the ant colony algorithm, the path order as the planning results of the ant colony algorithm is obtained as follows:

12 32 31 30 19 18 17 16 15 14 13 11 10 9 8 7 6 5 4 3 2 28 37 21 22 23 38 36 29 1 20 33 39 35 34 24 25 26 27 12.

The path is a closed one, as shown in Fig. 5, with the start and end points being the robot’s operating origin, which corresponds to the actual working conditions of the robot. The minimum distance of the path is 10,356.9353 mm.

Figure 6 shows six sets of algorithm convergence curves for the improved ant colony algorithm and pre-modified ant colony algorithm for path planning under different parameter. The blue lines are optimized combination of parameters, and the red lines are random combination of parameters.

The average number of iterations for convergence of the ant colony algorithm for path planning before the improvement is 45, while the average number of iterations for convergence of the improved ant colony algorithm is 17, which is reduced by more than 50%. Importantly, the improved ant colony algorithm can obtain shorter path than traditional ant colony algorithm. The average time of traditional ant colony algorithm for path planning is 5.52 s, but the average time of improved ant colony algorithm for path planning is 4.33 s.

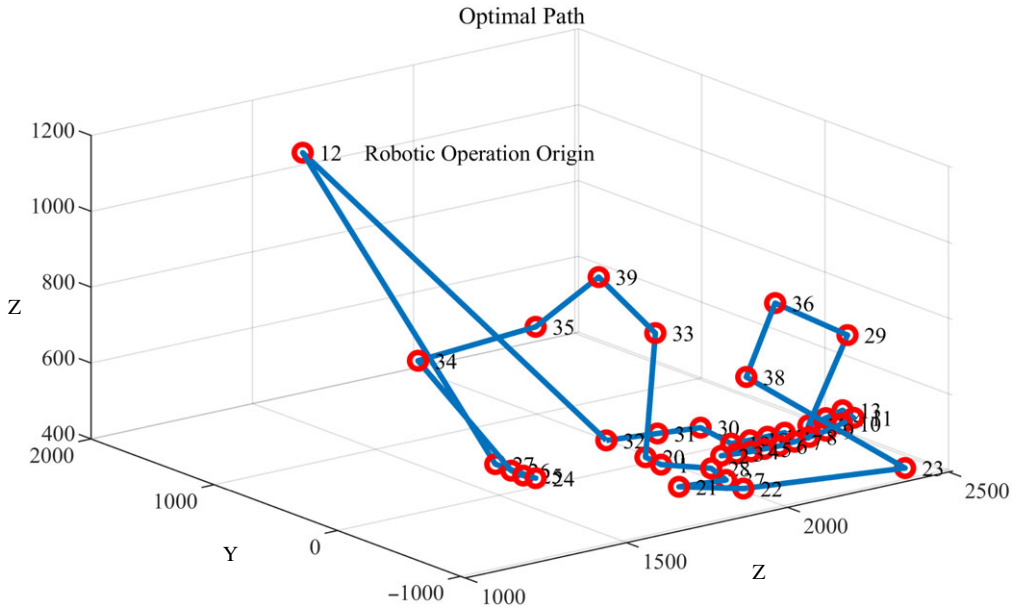


Figure 5. Path planning sequence.

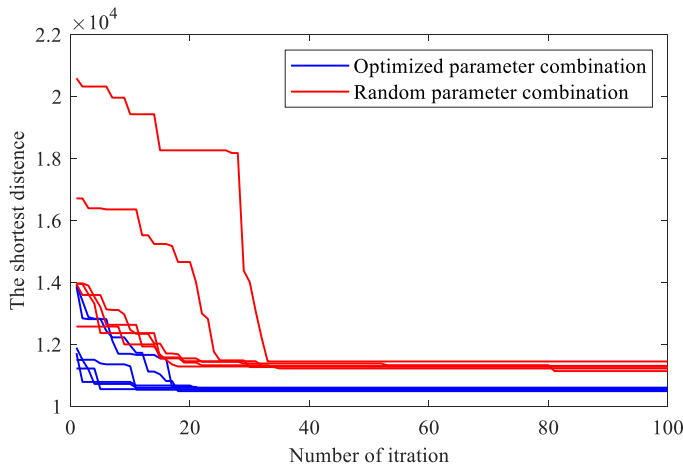


Figure 6. Convergence trajectory of original algorithm and optimized algorithm.

5. Conclusion

1. This paper analyzes the principles of ant colony algorithms and particle swarm algorithms, establishes a mathematical model for the spot welding path planning problem, designs a pheromone update scheme, and establishes an objective function.
2. Improve the ant colony algorithm by using the particle swarm algorithm to modify the pheromone factor  $\alpha$  in the ant colony algorithm and heuristic function factor  $\beta$  to avoid the disadvantage of relying on manual experience to select the parameters.
3. MATLAB was used to analyze the extent of the pheromone factor  $\alpha$  and the heuristic function factor  $\beta$  on the path planning algorithm.

4. The improved algorithm was used to plan the path for spot welding. The average path length obtained from the improved six sets of initial parameters was 10,357.7509 mm, which was 473.0885 mm less than the path distance planned by the unimproved algorithm. The improved algorithm had a higher quality solution.
5. The convergence analysis of the algorithm shows that the average number of iterations was 17 when the improved algorithm converges. The average time of improved ant colony algorithm for path planning was 4.33 s. The improved ant colony algorithm had fewer iterations and running time.

**Author contributions.** Yuesheng Tan conceived the study and formulated the overarching research aims. Jie Ouyang and Zhuo Zhang designed the study, designed computer program, analyzed data, and wrote the article. Yinglun Lao and Pengju Wen revised the article and conducted data gathering.

**Financial support.** This work was supported by National Nature Science Foundation of China (grant number: 51705022).

**Conflicts of interest.** None.

**Ethical considerations.** None.

## References

- [1] H. Z. Zhuang, S. X. Du and T. J. Wu, "Research on path planning and related algorithms for robots," *Bull. Sci. Technol.* **20**(3), 6 (2004).
- [2] J. Zhu and M. Y. Yang, "Path planning of manipulator to avoid obstacle based on improved artificial potential field method," *Comput. Meas. Cont.* **26**(10), 6 (2018).
- [3] K. Liang and J. L. Mao, "Path planning of indoor mobile robot based on improved ant colony algorithm," *Electron. Meas. Technol.* **42**(11), (2019).
- [4] S. M. Lavalle, "Rapidly-exploring random trees: a new tool for path planning," *Computer Science Dept.* **98**(11), 1–4 (1998).
- [5] X. Zhu, R. W. Chen and D. G. Xu, "Welding spot detection path planning method based on a novel particle swarm algorithm," *Chin J. Sci. Instrum.* **35**(11), 2482–2931 (2014).
- [6] C. Wurrll and D. Henrich, "Point-to-point and multi-goal path planning for industrial robots," *J. Intell. Robot Syst* **18**(8), 445–461 (2001).
- [7] C. S. Pei, S. P. Jeng and C. C. Shu, "A parallel compact cuckoo search algorithm for three-dimensional path planning," *Appl. Soft. Comput.* **94**, 106443 (2020).
- [8] L. S. Liu, J. X. Yao, D. W. He, J. Chen, J. Huang, H. Xu, B. Wang and J. F. Guo, "Global dynamic path planning fusion algorithm combining jump-A\* algorithm and dynamic window approach," *IEEE Access* **9**, 32–38 (2021).
- [9] H. N. Ma, Z. Zhang and Q. L. Li, "Improved artificial potential field method for dual-manipulator path planning algorithm," *J. Mech. Trans.* **45**(06), 77–84 (2021).
- [10] Z. Zhang, X. Y. Li, H. Z. Dong, L. Zhou and L. Gao, "Constrained sampling method based RRT algorithm for manipulator motion planning," *Comput. Integr. Manuf.* **40**(04), 1–18 (2022).
- [11] W. W. Xue, P. S. Yang, Y. Rui and X. S. Gu, "Intelligent Welding Robot Path Planning," *In: Proceedings of the 2015 China Intelligent Automation Academic Conference (2nd fascicle)* (2015), pp. 36–43.
- [12] J. H. Wang, Y and Wang, A study on BIW welding robot path planning based on ant colony algorithm," *Manufact. Automat.* **30**(5), 16–17 (2008).
- [13] J. G. Ling, S. X. Chen, H. C. Dai and W. J. Huang, "Application of ant colony algorithm in welding path planning of BIW," *Trans. China Weld. Inst.* **36**(01), 5–9 (2015).
- [14] G. Chen and J. Liu, "Mobile robot path planning using ant colony algorithm and improved potential field method," *Comput. Intel. Neurosci.* **b**(b), 1–10 (2019).
- [15] Q. Zhang, B. K. Chen, X. Y. Liu, X. Y. Liu and H. Yang, "Ant colony optimization with improved potential field heuristic for robot path planning," *Trans. Chin. Soc. Agric. Mach.* **50**(5), 11 (2019).
- [16] Y. Q. Xu, K. Luo, T. T. Li and W. G. Gao, "Path planning of mobile robot based on improved adaptive ant colony algorithm," *J. Electron. Meas. Instrument.* **31**(10), 531–533 (2019).
- [17] J. Z. Jia and X. J. Li, "Research on evacuation path planning in single-story building fire based on genetic-ant colony algorithm," *J. Saf. Sci. Technol.* **16**(6) (2020).
- [18] J. F. Yang, "Ant Colony Algorithm and Its Application Research," (Zhejiang University, Zhejiang, 2008) pp. 1–135.
- [19] C. Blum, "Ant colony optimization: introduction and recent trends," *Phys. Life. Rev* **2**(4), 353–373 (2005).
- [20] A. A. Kazharov and V. M. Kureichik, "Ant colony optimization algorithms for solving transportation problems," *J. Comput. Sys. Sc. Int* **49**(1), 30–43 (2010).

- [21] K. X. Min, H. W. Ge, Y. Zhang and Y. C. Liang, “Solving traveling salesman problem by an ACO and PSO based hybrid algorithm,” *J. Jilin Univ. (Inform. Sci. Edn.)* **13**(4), 402–405 (2006).
- [22] B. W. Tang, K. Xiang, M. Pang and Z. X. Zhu, “Multi-robot path planning using an improved self-adaptive particle swarm optimization,” *Int. J. Adv Robot. Syst* **17**(5), 1–19 (2020).