Time-Optimal Path Planning for Dual-Welding Robots Based on Intelligent Optimization Strategy

Xuewu Wang, Bin Tang, Yixin Yan and Xingsheng Gu

Abstract Dual-welding robots are widely used with the industry development, and dual-welding robots usually have to deal with a large number of weld joints. In this condition, traditional manual teaching method is time-consuming and inefficient. In this paper, an intelligent optimization strategy is proposed to realize time-optimal path planning for dual-welding robots. First, the welding robot path optimization problem is presented. Then, good diversity and convergence velocity of discrete group competition particle swarm optimization (GC-PSO) algorithm are tested. Compared with particle swarm optimization (PSO), genetic particle swarm optimization (GPSO) and chaos particle swarm optimization (CPSO) algorithms, GC-PSO algorithm shows its better optimization effectiveness. In addition, a method of collision detection and obstacle avoidance is given. At last, an intelligent optimization strategy is applied to time-optimal path planning for dual-welding robots, and the global optimal result can be obtained quickly. Simulation results show that the intelligent path planning strategy is effective and can be used for welding robot path optimization.

Keywords Particle swarm optimization (PSO) · Group competition
Welding robot · Path optimization · Obstacle avoidance

1 Introduction

Welding robot is widely used in industrial production process. Welding robot path planning mostly relies on the experience of engineers. This method is not only time-consuming and inefficient, but also difficult to find the desired welding path.

X. Wang (✉) · B. Tang · Y. Yan · X. Gu
Key Laboratory of Advanced Control and Optimization for Chemical Processes of Ministry of Education, East China University of Science and Technology, Shanghai, China
e-mail: wangxuew@ecust.edu.cn

S. Chen et al. (eds.), *Transactions on Intelligent Welding Manufacturing*,
Transactions on Intelligent Welding Manufacturing,
https://doi.org/10.1007/978-981-10-7043-3_3

Intelligent optimization algorithm provides a convenient and quick method for the welding robot path planning. Welding robot path optimization was simplified as the traveling salesman problem (TSP) problem and the path length was minimized based on double-global optimal particle swarm optimization (PSO) algorithm [1]. In [2], task sequencing and path planning in remote laser welding were studied based on TSP and meta-heuristic algorithm. The energy consumption and cycle time were optimized using restarted simulated annealing algorithm [3].

As widely used intelligent optimization algorithm, PSO is used to solve path planning problem. PSO [4] was first proposed by Kennedy and Eberhart in 1995. PSO algorithm has many advantages, such as simple structure, fast convergence speed and easy implementation. However, PSO has a disadvantage: when the optimized problem is complex, the dimension is high or there are a lot of local optimal values in the independent variables. In order to solve the premature problem of PSO algorithm and accelerate the convergence rate of the algorithm, many improvements were conducted. The first kind of improvement mostly aims at the PSO parameters, such as learning factor and inertia weight. In [5], a particle swarm algorithm with dynamic inertia weight adjustment was proposed to balance the global and local search ability of PSO. However, this improvement is largely dependent on the choice of random factors. The improvement of the position and velocity of PSO belongs to the second category. In [6], a position-weighted PSO algorithm was proposed to increase the determinacy and directionality of the particle searching for the optimal value. However, the improved method limits the search range and reduces the convergence rate of the particle. The third category is local search PSO algorithm based on the global optimal particle [7], such as chaos particle swarm optimization (CPSO) algorithm. In [8], the chaos was integrated into the motion of the particle, and the probability of falling into the local optimum was decreased. However, the algorithm complexity was increased and the convergence rate was reduced. The fourth category is based on the fusion of different intelligent optimization algorithms, such as genetic particle swarm optimization (GPSO) algorithm [9]. Incorporating the updating strategy into the PSO algorithm is the fifth category [10]. Improved PSO algorithm shows its advantages, such as fast rapid convergence and global optimization. Therefore, an improved PSO algorithm based on grouping and competition strategy is proposed to realize the welding robot path optimization.

Welding robot path optimization problem is described in Sect. 2. Group competition particle swarm optimization (GC-PSO) algorithm is presented in Sect. 3, and its discretization is also given. Then, the dual-robot obstacle avoidance strategy is presented in Sect. 4. Furthermore, time-optimal path planning for dual-welding robots is conducted based on GC-PSO in Sect. 5.

2 Optimization Problem Description

A part of the white body is selected as workpiece in this paper. The shape of the workpiece and the position of the weld joints are shown in Fig. 1. The robot is ABB R2400 robot and the welding tong is GTAW10. The welding pose will be presented in Sect. 3.3. For convenience, the pose of the welding tong is defined as "1" or "−1". When the longer part is in the upright position, the pose is defined as "1", otherwise it is defined as "−1". In view of the actual situation of the workpiece and fixture, both poses in welding process can be applied to some welding joints. Such a condition is defined as "0".

Efficiency is the critical goal for industrial production process, and welding time is the most direct efficiency indicator. In actual welding process, welding pose is related to welding time and obstacle avoidance. Therefore, welding pose is considered during conducting welding robot path planning.

In this paper, two robots are placed symmetrically and oppositely. The welding joints are assigned based on the following principles. First, the working space of the welding robots does not overlap. Next, welding joints with the same pose are assigned to the same robot. Besides, welding joints are divided to obtain the shortest welding time and the welding time for the two robots is nearly the same. If the welding time difference between the two robots is greater than the minimum difference, the welding joint with the farthest distance from the robot is assigned to the other robot until the two robots have nearly the same welding time and the shortest total time.

The time-optimal path planning for the dual-welding robots requires that the welding tong walks through all welding joints and the cost time is the shortest. Suppose that the number of weld joints is M, the number of transition points is N, and the weld joint order is $\pi(i)$ ($i = 1, 2, \ldots, n$). Then, the time-optimal path planning problem can be regarded as a constraint TSP problem. The welding robot global path planning problem can be described as

$$\min T = \sum_{i=1}^{N-1} L_{\pi(i),\,\pi(i+1)}/v, \tag{1}$$

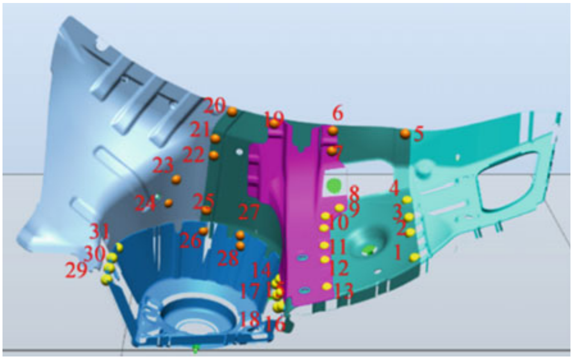


Fig. 1 Welding workpiece

$$\text{s.t. path}\,\pi(i)\,\pi(i+1)\ \text{is safe path},\ i=1,\ 2,\ \ldots,\ n-1, \tag{2}$$

where $\sum_{i=1}^{N-1} L_{\pi(i),\,\pi(i+1)}$ is the sum of the distances between two welding joints, v is the welding speed which is set as 2 m/s, and path $\pi(i)$ $\pi(i + 1)$ is the path between two welding joints $\pi(i)$ and $\pi(i + 1)$.

3 GC-PSO

3.1 Algorithm Introduction

Because traditional PSO algorithm slowly converges and easily falls into local optimum, GC-PSO algorithm is proposed in this paper. The algorithm divides the particle swarm into two parts according to the fitness value of each particle. Particles with fitness value in the top 20% of the total fitness value are regarded as leading particles, and the remaining particles are followers. After dividing all particles into two parts, all the particles are grouped randomly. Each group consists of a leading particle and some followers, where the followers are randomly assigned to the leading particles and the number of followers in each group is not unique. When iteration number satisfies t = 10, the fitness value of the particle is reordered. Then, the leading particle and followers are defined according to the fitness value. And all the particles are grouped randomly again.

GC-PSO algorithm adopts different speed updating strategies for different particles. In order to avoid the particle falling into local optimum, GC-PSO algorithm introduces intra-group competition and inter-group competition in the speed updating formula [11].

The velocity updating formulas for leading particles is described as

$$v_i^{t+1} = \omega v_i^t + v_i^t\,\mathrm{Randn}\left(0, \sigma^2\right), \tag{3}$$

where

$$\sigma^2 = \begin{cases} 1, \mathit{if} f_i < f_k \\ \mathrm{e}^{\frac{-f_i + f_k}{|f_i + \epsilon|}}, \text{otherwise} \end{cases}, k \in [1, N_l], k \neq i. \tag{4}$$

The location updating formula for leading particles is described as

$$x_i^{k+1} = x_i^k + v_i^{k+1}, \tag{5}$$

where Randn(0, σ^2) is a Gaussian distribution function with mean 0 and variance σ^2. The parameter Randn(0, σ^2) expands the searching range of particle and avoids the particle falling into local optimization. ε is an infinitely small number which promises the denominator is not zero. k denotes the number of the other leading particles which will increase the competition between the particles. This strategy can make the particle with poor fitness moves closer to the particle with better fitness. f is the corresponding fitness value of each particle. N_l is number of leading particles.

The speed updating formulas for follower are given as

$$v_1^{t+1} = \omega v_i^t + s_1 \,\mathrm{Rand}\,(v_{j1}^t - v_i^t) + s_2 \,\mathrm{Rand}\,(v_{j2}^t - v_i^t), \tag{6}$$

$$s_1 = \mathrm{e}^{\frac{f_i - f_{j1}}{|f_i| + \varepsilon}}, \tag{7}$$

$$s_2 = \mathrm{e}^{\left(f_{j2} - f_i\right)}. \tag{8}$$

The location updating formula for follower is given as

$$x_i^{k+1} = x_i^k + v_i^{k+1}. \tag{9}$$

The velocity updating formula for follower contains two parameters s_1 and s_2. s_1 is the intra-group competition coefficient, and j_1 is the number of leading particles in the group. Follower competes with the leading particle with probability s_1. s_2 is the inter-group competition coefficient, and j_2 is the number of the leading particles in other groups. Followers in this group compete with the leading particles in other groups with the probability s_2.

The detailed flow of the algorithm is presented as follows.

Step 1 Initialize the particle swarm, and define the related parameters: the number of leading particle, the number of following particle, and the particle size Popsize.

Step 2 Calculate the fitness value of the particles and determine the individual optimal position p_{best} and the global optimal position g_{best}; set t as 1.

Step 3 After iterating G times, the particles are reordered and grouped according to the fitness value. G = 10 denotes iteration time.

Step 4 Update the position, velocity and fitness values of the leading particles and followers according to Eqs. (3), (5), (6) and (9).

Step 5 Update the individual optimal position p_{best} and the global optimal position g_{best} of the current particle swarm.

Step 6 Set $t = t + 1$; stop if the iteration condition is satisfied; otherwise, return to Step 3.

The number of leading particles and the value of update coefficient have an important influence on the convergence precision and convergence speed of the algorithm. If N_l and G are set too large, the algorithm cannot converge quickly to the global optimal value. If N_l and G are set too small, the algorithm easily falls into local optimum. After tests, the following conclusions can be drawn. When N_l is set as 20% of the total number of particles, and G is set as 10, the convergence rate is improved obviously and the convergence precision is guaranteed. In addition, ω decreases exponentially from 0.9 to 0.4 with the increase of the iteration for the convergence accuracy, convergence rate and robustness of the algorithm.

3.2 Algorithm Discretization

Although the GC-PSO shows the ability of fast convergence and optimization, it can only solve the continuous problem. In order to solve the problem of dual-robot path planning, the GC-PSO algorithm needs to be discretized.

In discrete PSO algorithm, each particle represents a feasible solution, and the population is a set of feasible solutions. Like continuous PSO algorithm, x_i in discrete particle swarm algorithm also represents the ith sorting result, v_i represents the velocity of the ith particle, p_{best} represents the best individual, and g_{best} represents the best population sort. Among them, v_i is a set of directions the particle can search; x_i, p_{best} and g_{best} are the results of optimization. Equations (3), (5), (6) and (9) are updated as follows.

Velocity and position updating equations for leading particle are respectively presented as

$$v_i^{t+1} = \omega v_i^t + v_i^t \,\text{Randn}\left(0, \sigma^2\right), \tag{10}$$

$$x_i^{t+1} = x_i^t \oplus v_i^{t+1}. \tag{11}$$

Velocity and position updating equations for follower are respectively presented as

$$v_i^{t+1} = \omega v_i^t + s_1 \,\text{Rand}\left(v_{j1}^t - v_i^t\right) + s_2 \,\text{Rand}\left(v_{j2}^t - v_i^t\right), \tag{12}$$

$$x_i^{t+1} = x_i^t \oplus v_i^{t+1}. \tag{13}$$

In the above equations, the operators +, − and $\oplus$ have new definitions. The definitions include the rule of particle crossover and combination with individual and global, which is important to transfer continuous algorithm to the discrete algorithm. Subtraction operator "−" represents the difference set of individual optimal position and the current position. For the example of $x_i^t \oplus v_i^{t+1}$o, $\oplus$

operation refers to conduction exchange order v_i^{t+1} for x_i^t, where v_i^{t+1} is a set of particle exchange orders. Addition operator "+" represents the union of two edge sets. The above discretization method inherits the characteristics of continuous GC-PSO. The updating process of GC-PSO is the process moving to the global optimal solution.

3.3 Algorithm Validation

Convergence rate and accuracy among standard PSO, genetic algorithm (GA), GPSO and GC-PSO algorithms are compared based on four TSPs. Four algorithms independently run 30 times for each test function, the population size is set as 100, and the maximum number of iterations for each run is set as 500 ω decreases exponentially from 0.9 to 0.4 with the increase of the iteration. Other parameters for these algorithms are listed in Table 1.

The average convergence curves of four algorithms are shown in Fig. 2. It can be concluded that GC-PSO still shows excellent convergence speed and accuracy with the same parameters and discrete method. GC-PSO algorithm uses the intra-group and inter-group competitions by the speed updating formula to make each particle move toward the global optimal position. The group division strategy ensures that the algorithm does not fall into local optimum. The simulation results show that GC-PSO algorithm is still feasible and efficient after discretization.

4 Dual-Robot Obstacle Avoidance Strategy

4.1 Three-Dimensional Grid Method Modeling

Working environment model for robot obstacle avoidance is established first. Grid method can establish an intuitive working environment which is conducive to judge local environment. Hence, the three-dimensional grid method is selected in this

Table 1 Algorithms parameters

Algorithm	Parameter
PSO	$c_1 = 1.49445,\ c_2 = 1.49445,\ \omega_{\max} = 0.9,\ \omega_{\min} = 0.4$
GPSO [12]	$c_1 = 1.49445,\ c_2 = 1.49445,\ p_c = 0.7,\ p_{\mathrm{m}} = 0.05$ $\omega_{\max} = 0.9, \omega_{\min} = 0.4$
GA [13]	$p_c = 0.7,\ p_m = 0.05,\ \mathrm{GGAP} = 0.1$
GC-PSO	$c_1 = 1.49445,\ c_2 = 1.49445,\ \omega_{\max} = 0.9,\ \omega_{\min} = 0.4$ leading particle percent $= 0.2$, follower percent $= 0.8$

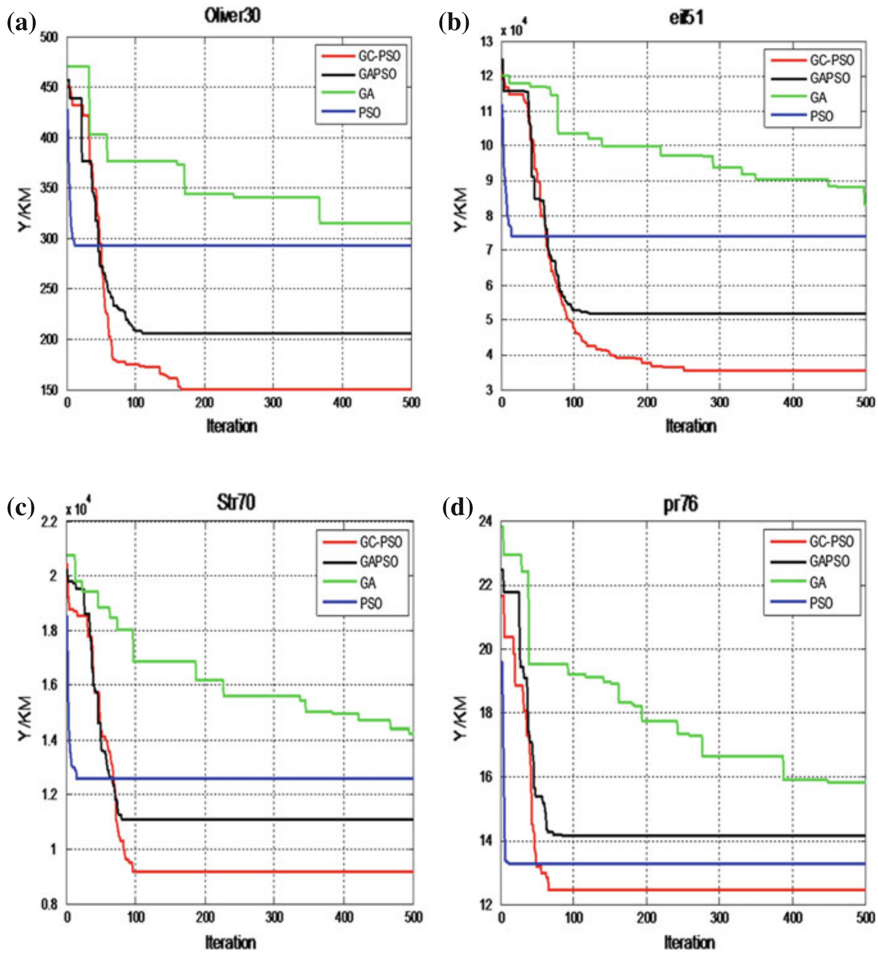


Fig. 2 The average convergence curves of four algorithms

paper to establish the working environmental model. And the steps are given as follows.

Step 1 Simplify the workpiece as a combination of some triangles. This is because free grid and obstacle grid are more easily identified through triangles.

Step 2 Create the grid matrix. Grid size affects the accuracy of path planning. The less the grid is, the better the accuracy of the path is, but this will take a long time to search the best path. The larger the grid is, the worse the accuracy of the path is, while the best path can be quickly found. In view of the searching time and accuracy, the whole space is divided into cubes

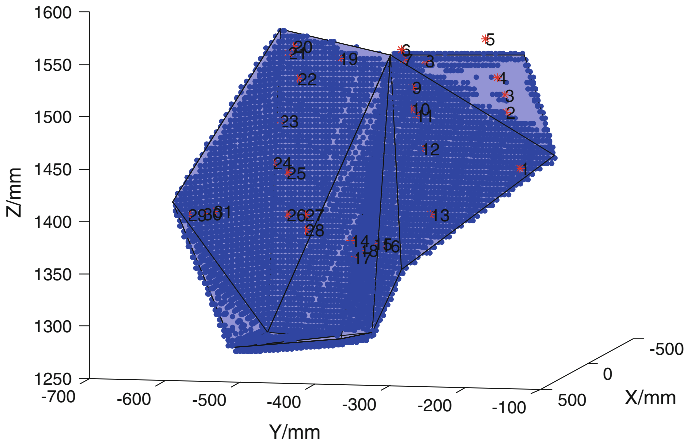


Fig. 3 Three-dimensional environment model using grid method

with a side length of 5 mm. The center of each cube is used as the starting point of the search path. Each center point is projected to a plane. If the projection point is outside the triangle, this triangle is not an obstacle at this point. If the projection point is inside the triangle and the length of the vertical line is less than 6 mm, the triangle is an obstacle.

Step 3 Identify the free grids and obstacle grids. If there is an obstacle for the center point, it means that the point is the obstacle point and the related grid is an obstacle grid; otherwise, the point is a free point and the related grid is a free grid. Obstacle points are indicated by *, as shown in Fig. 3.

4.2 Obstacle Avoidance Between Robot and Workpiece

Local searching starts from initial solution, and begins to search the vicinity field. If particle can find a better solution, then it replaces the initial solution. Ant colony algorithm is applied to realize local obstacle avoidance path planning [14].

The parameters of ant colony optimization (ACO) are initialized as follows. Based on the empirical value, the weight α of the pheromone is set as 1, the weight β of heuristic pheromone is set as 11, the evaporation coefficient ρ of pheromone is set as 0.9, and the pheromone quality coefficient Q is set as 5. The iteration number N is set as 50, and the population quantity M is set as 50. The coordinates of the starting point and the terminal point are initialized. The initialized pheromones for all points are set as 0.5. Iterator is defined as n. The number of ants is expressed as k.

The local obstacle avoidance path of two robots can be obtained by the local search algorithm. However, the path obtained by ant colony algorithm is not a

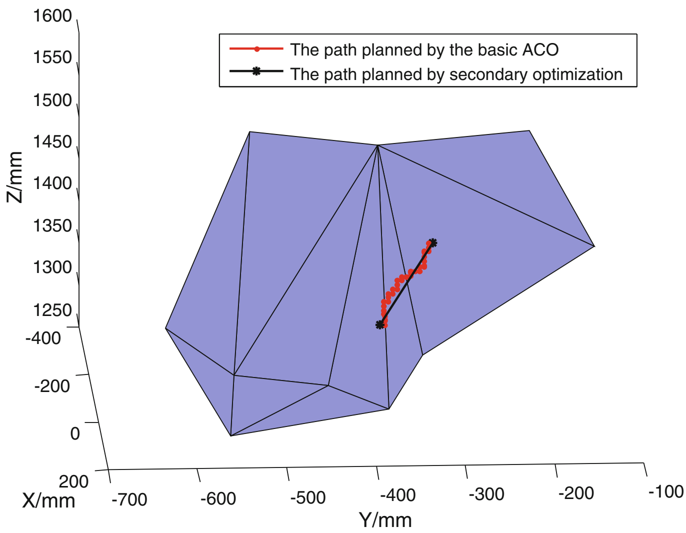


Fig. 4 Local obstacle avoidance path planning

straight line, so it cannot meet requirement of the shortest welding path. In order to achieve the shortest and collision free welding path, second optimization is conducted. Principles of the second optimization are presented as follows. Some nodes are canceled and leaved nodes are connected to obtain a shorter path. In the process, collision detection is always conducted to promise a collision free path. Welding joints 12 and 15 are taken as an example. The simulation results are shown in Fig. 4.

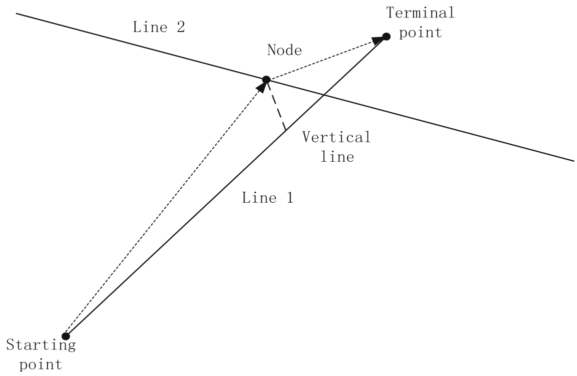
4.3 *Obstacle Avoidance Between Robots and Fixture*

The obstacle avoidance between robot and fixture needs to be studied. In this paper, the welding tong is regard as a point, and the distance between welding tong and fixture is calculated to conduct collision detection.

Collision detection between welding tong and fixture steps is given as follows. A welding path is obtained by optimization algorithm firstly. Then, the shortest distance between welding tong and fixture is calculated. If the shortest distance is less than the safety threshold, geometrical method [15] is used to obtain a transition point to avoid collision.

Figure 5 shows two welding joint positions in the adjacent region. Starting point and terminal point are connected in a line which is called Line 1. The intersection of two planes is called Line 2. Lines 1 and 2 locate on different surfaces. A transition point in Line 2 is obtained to make the path shortest, which moves from the starting

Fig. 5 Transition point solution

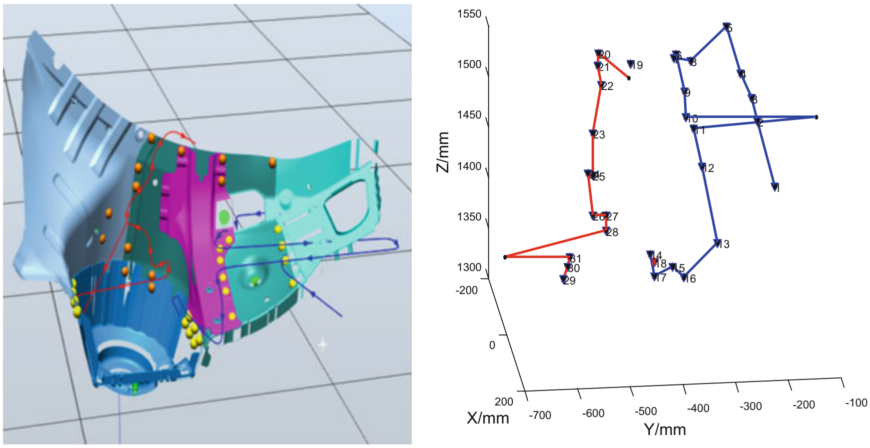


point, the transition point, and reaches the terminal point. This transition point is the intersection of the line 2 and the middle vertical line of the two lines.

5 Time-Optimal Path Planning for Dual-Welding Robots

Based on the environment modeling and the obstacle avoidance strategy, GC-PSO algorithm is used to optimize the robot welding time, and realizes the time-optimal obstacle avoidance path planning. Assume that the welding speed of the robot is 2 m/s, and the welding time of each weld joint is 0.5 s. The steps of time-optimal path planning are presented as follows.

Step 1 Set the position of two robots which are placed on the two sides of workpiece, and determine the weld joint coordinates.
Step 2 Initially assign all the welding joints for two robots according to the assignment principle.
Step 3 Establish weldment and robot workspace model according to the grid method.
Step 4 Obtain the local collision free path for robot and weldment by ant colony algorithm.
Step 5 Realize collision free path among the welding tong, tooling fixture and workpiece based on collision detection and geometry method.
Step 6 Based on the division result of the welding joints, calculate the welding time of each robot by discrete GC-PSO algorithm.
Step 7 If the welding time difference between two robots is greater than the set time difference, divide the weld joints again according to the division principle of weld joints, and return to Step 6. Otherwise, go to Step 8.
Step 8 Output the optimized welding joint order and the welding time of each robot.



(a) Dual-robot welding path in RobotStudio (b) Dual-robot welding path in Matlab

Fig. 6 Optimization results of welding path

In this paper, in order to meet the requirement of the shortest welding time, welding joints are divided according to the welding pose. This principle can reduce the welding pose change in welding process. In order to facilitate the calculation, the welding tong reverses when it arrives at the transition point, and the reversing time is set as 2 s. For example, there is a pillar between the welding joints 10 and 11 for the robot 2. Hence, geometric method is used to avoid collision between the robot and fixture. A transition point is selected at the edge of the workpiece. When the robot 2 finishes the welding of the welding joint 10, it moves to the transition point. Then, it moves to the welding joint 12 and welds the welding joint 12.

Based on the optimization strategy, welding path lengths for two robots are 70.2914 and 109.29004 mm, respectively. The final optimal welding time is 94.259072s. Global path planning orders are: 19–20–21–22–23–25–24–26–27–28–31–30–29 and 1–2–3–4–5–8–7–6–9–10–11–12–13–16–15–17–14–18, respectively. The path planning results with obstacle avoidance for dual-welding robots is shown in Fig. 6.

6 Conclusion

Compared with traditional manual teaching method, intelligent robot path planning has a high industrial application value. In order to realize intelligent welding path planning for two robots, GC-PSO algorithm and obstacle avoidance strategy are studied after the optimization problem is described. Then, the dual-robot time-optimal path planning is conducted based on the mentioned optimization strategy. The optimized welding path can help welding engineering by shortening the teaching time. As welding robot application and artificial intelligence technology increase rapidly, intelligent robot welding path planning will draw more attention, and will play an important role in welding automation in the future.

It can be seen that only simulation is performed in this paper. Detailed research works need to be done to improve the optimization strategy. And some experiments also need to be done to promise the strategy effectiveness.

Acknowledgements The work is founded by the Shanghai Nature Science Foundation (14ZR1409900), and the National Natural Science Foundation of China (61773165, 61573144).

References

1. Wang XW, Shi YP, Ding DY et al (2016) Double global optimum genetic algorithm-particle swarm optimization-based welding robot path planning. Eng Optim 48(2):299–316
2. Kovács A (2016) Integrated task sequencing and path planning for robotic remote laser welding. Int J Prod Res 54(4):1210–1224
3. Li Z, Tang Q, Zhang LP (2016) Minimizing energy consumption and cycle time in two-sided robotic assembly line systems using restarted simulated annealing algorithm. J Clean Prod 135:508–522
4. Kennedy J et al (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, vol 4, IEEE, Piscataway, pp 1942–1948
5. Amoshahy MJ, Shamsi M, Sedaaghi MH (2016) A novel flexible inertia weight particle swarm optimization algorithm. PLoS ONE 38:281–295
6. Guedria NB (2016) Improved accelerated PSO algorithm for mechanical engineering optimization problems. Appl Soft Comput 40(40):455–467
7. Ye Y, Yin CB, Gong Y et al (2016) Position control of nonlinear hydraulic system using an improved PSO based PID controller. Mech Syst Signal Process 83:241–259
8. Dong N, Fang X, Wu AG (2016) A novel chaotic particle swarm optimization algorithm for parking space guidance. Math Probl Eng 2016:1–14
9. Benvidi A, Abbasi S, Gharaghani S et al (2017) Spectrophotometric determination of synthetic colorants using PSO-GA-ANN. Food Chem 220:377–384
10. Wu DQ, Zheng JG (2013) Parallel particle swarm optimization algorithm based on hybrid strategy adaptive learning. Control Decis 28(7):1087–1093
11. Wu D, Xu S, Kong F (2016) Convergence analysis and improvement of chicken swarm optimization. IEEE Access 2016(99):1–1
12. Yan LM, Gu YC, Li JD (2011) Optimal strategy of searching FPD weights scanning matrix using GA-PSO. J Shanghai Univ 15(4):292–296
13. Jayalakshmi GA, Sathiamoorthy S, Rajaram R (2001) A hybrid genetic algorithm: a new approach to solve traveling salesman problem. Int J Comput Eng Sci 2(2):339
14. Wang XW, Xue LK, Yan YX et al (2017) Welding robot collision-free path optimization. Appl Sci 7(2):89
15. Wang XW, Yan YX, Ding DY et al (2016) Obstacle avoidance path planning of welding robot based on Lévy-PSO algorithm. J Shanghai Jiao Tong Univ 50(10):1517–1520