Robotica: (2021) volume 39, pp. 428–437. © The Author(s), 2020. Published by Cambridge University Press.
doi:10.1017/S0263574720000454

Autonomous Intelligent Planning Method for Welding Path of Complex Ship Components

Tao Wang†, Zhilong Xue†, Xiaoqing Dong‡ and Senlin Xie‡*

†*School of Automation, Guangdong University of Technology, Guanzhou 510006, China. E-mails: wangtao_cps@gdut.edu.cn, 791396753@QQ.com*
‡*School of Physics and Electronic Engineering, Hanshan Normal University, Chaozhou 521041, China. E-mail: dxqzq110@163.com*

(Accepted May 3, 2020. First published online: June 18, 2020)

SUMMARY
Aiming at planning the welding path of complex ship components, a welding path planning optimization model was constructed with the shortest welding path and using the target and the welding process and welding starting and ending points as constraints. Based on the model, an improved ant colony algorithm with dynamic adaptive parameters was proposed to complete the path planning work. Simulation results showed that, compared with other classical optimization algorithms, the proposed algorithm improved optimization speed while ensuring optimization effect and achieving better results in path planning of complex ship components.

KEYWORDS: Ship welding; Complex components; Path planning; Improved ant colony algorithm.

1. Introduction

China's shipbuilding industry has developed rapidly and became the world's largest in 2010,[1] but it is facing the problems of big but not strong low-end excess capacity and insufficient high-end capacity. The '13th Five-Year Plan' pointed out that the development of the shipbuilding industry must change its thinking and improve the comprehensive technical level of shipbuilding, especially in the areas of intelligent manufacturing, introduction of intelligent robots and the quality and manufacturing capacity of shipbuilding products. High-end products enter the army and accelerate the development of the ship supporting industry.[2] Welding is one of the core technologies of the ship, mainly arc welding, and welding accounts for 70–90% of the whole production.[3] The number of complex sheet members of a 50,000-ton ship is generally more than 2000. At the same time, there are components with different shapes, sizes, thicknesses and structural forms. The proportion of identical components in the same ship is $<1\%$, making it difficult to mechanize mass production. Moreover, the ship's structure is very complicated in actual production; although the overall structure of different types of hulls is roughly the same, the internal structure is quite different. The main performance is the variety of welds, different lengths and small spaces. Its conventional complex sheet member is shown in Fig. 1. At present, ship welding relies mainly on labor, and the degree of automation is generally low. It is difficult to guarantee the quality and efficiency of welding. If offline robotic programming can only be applied to the welding of some conventional components, the adaptability may be poor, and the welding path planning is completely dependent on the experience of the path planning engineer. Therefore, the development of autonomous arc welding robots has become an urgent need in the ship welding industry.

* Corresponding author. E-mail: senlinxie@163.com

https://doi.org/10.1017/S0263574720000454 Published online by Cambridge University Press



Fig. 1. Schematic drawings of conventional complex sheet components.

At present, some research has been done to realize autonomous planning of robot welding path using artificial intelligence algorithms. In ref. [4], an ant colony algorithm was prone to problems of long search time, low efficiency and prone to falling into local optimum in robot welding planning. The Adadelta algorithm was introduced to update the pheromone and volatilization coefficient to improve the system search ability. One study[5] sought the obstacle avoidance path between two solder joints by using a fast ant colony algorithm, and optimized robot welding time by using a group competitive particle algorithm. Another study[6] was directed to the path planning problem of automobile white body welding. A fusion algorithm based on ant colony particle swarm optimization was proposed. The elite idea was added to the ant colony optimization (ACO) algorithm, and the state transition rule and pheromone update method were changed, which increased the global shrinkage ability of the algorithm. The chaotic total group was introduced and the evolutionary idea of genetic algorithm (GA) was added. The fast convergence ability of the algorithm was also improved. One study[7] proposed an ant colony algorithm using a hybrid pheromone update strategy to improve convergence speed and avoid falling into the local optimal solution. In literature,[8–13] several other algorithms (such as GA, particle swarm algorithm [PSO]) in combination with ant colony algorithm have shown good effect. However, combining with other algorithms cannot solve the problem of performance-dependent parameter settings of ant colony algorithms. Generally, the relationship between the parameters and optimal path needs to be analyzed, which is cumbersome.

Although successful cases have been achieved in the autonomous intelligent planning of welding paths, most of the results were applied in the field of automobile body-in-white electric welding; there are few studies and applications based on ship arc welding. First of all, the processes of ship arc welding and automobile electric welding differ greatly; secondly, there are many complex components in ships with complex specifications, and some components have large welding spans and many welding points. Therefore, this article focuses on the welding path planning of complex ship components, analyzes and mathematically models the welding constraint problems, and proposes an autonomous intelligent planning algorithm suitable for ship welding path planning. Through simulation comparison, the algorithm effectively seemed to solve the above problems.

2. Welding Path Model and Problem Definition

2.1. Welding path model

The paths in arc welding path planning can be divided into two types: one is the welding path and the other is the empty path. The welding path represents the path and the constraints the robot must pass when welding. The no-load path is the path from the end of one weld to the start of another weld during welding. The free-load path is an arbitrary-free path.

According to process requirements, the welding direction is divided into a directed weld and a non-directional weld. The direct weld refers to the weld direction that is unidirectional (e.g. the fillet weld specifies that the welding be carried out from bottom to top and be unidirectional). A non-directional weld means that the weld can be two-way and can start from either end of the weld.

Robotic welding is considered a traveling salesman problem (TSP) with multiple constraints. The problem can be described as: given n welds, m welds or n–m welds, the manipulator completes all welding tasks from the starting point and finally returns to the starting point in the shortest distance. A mathematical description is:

https://doi.org/10.1017/S0263574720000454 Published online by Cambridge University Press

$$G = (L, V, E, D) \tag{1}$$

where the weld set is $L = \{l_1, l_2, l_3 \ldots l_n\}$, directional weld is $L_0 = \{l_1, l_2, l_3, \ldots l_m\}$, and non-directional weld is $L_1 = \{l_{m+1}, l_{m+2}, l_{m+3}, \ldots l_n\}$. The set coordinates of the starting point and the end point of n welds are expressed as $V = \{v_1, v_2, v_3 \ldots v_{2n}\}$; $E = \{e_{ij}(i, j)|i \in V, j \in V, i \neq j\}$ is a collection of all solder joints that line the edges (representing all the paths robotic soldering can pass). $D = \{d_{ij}|(i, j)|i \in V, j \in V, d_{ij} > 0\}$, d_{ij} represents the distance between the i-th point and the j-th point. When d_{ij} represents a undirected weld, $d_{ij} = d_{ji}$; when d_{ij} indicates a directional weld, d_{ji}=M, where M represents an infinite value.

Let $x_{ij} = (0|1)$ denote the path e_{ij} through which the i-th point to the j-th point pass, where 0 means no pass and 1 means pass.

2.2. Mathematical model of welding path planning

Welding efficiency is represented by the length of the welding path, and the shortest welding path indicates the highest welding efficiency. The shortest welding path can be expressed as follows:

$$\min F = \sum d_{ij} x_{ij} \tag{2}$$

In this paper, the welding path is the shortest target, and the welding direction, welding start and end points and welding path are the constraints. A welding path planning optimization problem of an arc welding robot can be described as:

$$\min F = \sum d_{ij} x_{ij} \tag{3}$$

$$s.t. \begin{cases} x_{ij} = 1 & i \to j \in L0 \quad (4) \\ x_{ij} + x_{ji} = 1 & i \leftrightarrow j \in L1 \quad (5) \\ \sum_{i=1}^{2n} x_{ij} & i \in V \quad (6) \\ \sum_{j=1}^{2n} x_{ij} & j \in V \quad (7) \\ \sum_{i \in V} \sum_{j \in V} x_{ij} \leq |S| - 1 & \forall S \in V, V = 1 : S \quad (8) \\ x_{ij} \in \{0, 1\} & i, j \in V, i \neq j \quad (9) \end{cases}$$

Here, Eq. (3) represents the objective function established by the shortest manipulator trajectory, and Eqs. 4–9 are compression conditions. Eq. 4 indicates the welding is unidirectional, and Eq. 5 indicates the welding direction is bidirectional. Eqs. 6 and 7 indicate the weld start and end joints have only one inlet and one outlet, and Eqs. 8 and 9 indicate there is only one loop for any connection path.

3. Improved ACO Algorithm

ACO[14] derives from the behavior of path finding during ant foraging. This is a meta-heuristic algorithm for solving combinatorial optimization problems. Path planning issues have unique advantages. However, the basic ant colony algorithm has some obvious defects, such as prone to falling into a local optimal solution during the search and the time to converge to a global optimal may be too long. Moreover, the classical ant colony algorithm usually cannot balance convergence speed and optimization effect. The setting of several parameters mutually restricts each other. At different stages of the system, the coefficients have different requirements. Therefore, according to different stages of system iteration, dynamic optimization parameters can effectively improve system performance.

The Ant Colony System[14] and the Maximal Minimum Ant System (MMAS)[15] are two adaptive ant colony algorithms that have solved the TSP and quadratic assignment problems. The graph coloring problem (GCP) is the most effective ant colony algorithm.[16] Combining these two algorithms with the characteristics of ship welding, the following improvements could be made to the classical ant colony algorithm.

https://doi.org/10.1017/S0263574720000454 Published online by Cambridge University Press

3.1. Increase convergence speed control coefficient (q)

State transitions are performed using pseudo-random scale rules. This provides a more reasonable approach to the choice of a new path. Here, the control parameter q is introduced, and the degree to which an ant could explore a new path can be adjusted by the q value, thereby effectively enhancing the searching ability of the ant in the range space concentrated in the optimal solution. Each time an ant starts from solder joint i, the transfer rule of the next solder joint j that has not been assigned is:

$$j = \begin{cases} \arg\max\limits_{u \notin tabu_k} \{[\tau_{iu}(t)]^{\alpha}[\eta_{iu}(t)]^{\beta}\} & \text{if } q \le q_0 \\ J & \text{if } q > q_0 \end{cases} \tag{10}$$

$$P_{ij}^{k}(t) = \begin{cases} \left\{ \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{s \in a_k}[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}} \right. & j \in a_{tabu_k} \\ 0 & \text{other} \end{cases} \tag{11}$$

In Eqs. (10) and (11), the pheromone level on the path from point i to point u is represented by $\tau_{iu}(t)$ at time t; $\eta_{ij}(t) = 1/d_{ij}$ is a heuristic factor, where d_{ij} is the distance from point i to point j, so if the path is shorter, the higher the probability of being selected; α is the information heuristic factor – the larger value of α, the greater the probability that an ant will choose the path that has passed before, reducing the randomness of the path searching ability. On the contrary, the ant colony search range will be reduced and prone to falling into local optimum. β is the heuristic factor; the larger the value of β, the easier it is for the ant colony to select a locally shorter path, which leads to faster convergence of the algorithm. However, the randomness may not be high, and it would be easy to get the local relative optimal solution. Therefore, in the classical ant colony algorithm, it is difficult to balance both convergence speed and global optimal solution.

The value range of q_0 is $q_0 \in [0, 1]$, where q_0 is a random value, obeying a uniform distribution, and q is a set threshold. Generally, when the value is in the range of [0.8–0.9], the algorithm works better. If $q \le q_0$, the ants will select the next solder joint with the largest value of $[\tau_{iu}(t)]^{\alpha}[\eta_{iu}(t)]^{\beta}$, according to Eq. (10). In this case, the system only selects the optimal path to speed up convergence; otherwise, the next solder joint is selected probabilistically according to Eq. (11). Therefore, by adjusting the value of q, the ability of the algorithm to explore the new path can be changed, thereby determining whether the next step of the algorithm should be to explore a new region or accelerate convergence near the current optimal path, thereby ensuring convergence speed and randomness.

3.2. Strengthening the optimal solution

The pheromone of all ants will no longer be updated, but only that of the best ant in each cycle. The rules are as follows:

$$\tau_{i,j}(t+n) \leftarrow (1-\rho)\tau_{i,j}(t) + \rho\Delta\tau_{i,j}(t) \tag{12}$$

$$\Delta\tau_{i,j}(t) = \begin{cases} \&\dfrac{Q}{L_{best}} & L_{best} \text{ The shortest path, overall situation} \\ \&0 & \text{Other situations} \end{cases} \tag{13}$$

In Eq. (12), ρ represents a volatility coefficient, and may take a value of 0–1. When the value of ρ is too small, there are too many pheromones remaining on each path, which causes an invalid path to continue to be searched, affecting the convergence rate of the algorithm. Conversely, although an invalid path can be excluded from searching, it cannot guarantee a valid path will be discarded, resulting in a search that affects the optimal value.

In order to enhance the utilization of an optimal path, only the pheromone left by the global optimal solution can be enhanced at a time (which could be the ant of the optimal solution in the current iteration, or that which appeared in the previous iteration), and the pheromones of other paths will only weaken. The stronger the pheromone, the greater the probability that the path will be selected by the ant in next optimization. Through a positive feedback, the optimal path will be utilized efficiently, speeding up system convergence.

https://doi.org/10.1017/S0263574720000454 Published online by Cambridge University Press

3.3. Dynamically update the upper and lower limits of the pheromone

All paths are set to a random value τ_0 at system initialization, and $\tau_0 = \tau_{\max}$, and the pheromone concentration of each path is limited to $[\tau_{\min}, \tau_{\max}]$, and the pheromone limit is dynamically adjusted according to optimization. If the value of $\tau_{\min}$ is too small, the pheromone's guiding effect on ants is weakened, the randomness of ant motion is enhanced and the convergence speed of the system is slow or even stagnant. If the value of $\tau_{\min}$ is too large, the probability of ants selecting a certain path is too large (probability close to 1), the ants no longer have randomness and the system may fall into local optimum. Therefore, limiting the value of pheromone to a dynamic and controllable range can effectively avoid stagnation during optimization. The specific settings are as follows:

At each iteration update, $\tau_{\min}$ is taken as a function of the maximum value in the current pheromone matrix:

$$\tau_{\min}(t+1) = \frac{1}{c}\left\{\max_{\forall i,j}(\tau_{ij}(t))\right\} \tag{14}$$

In Eq. (14), c is a constant and is the maximum value of the current iteration pheromone.

$$\tau_{\max}(t+1) = \max\left(\rho\tau_0(t) + \sum_{i=1}^{n}\Delta\tau_{ij}^{k}(t, t+1) + \sigma\Delta\tau_{ij}^{*}(t, t+1)\right) \tag{15}$$

$$\Delta\tau_{ij}^{k}(t, t+1) = \begin{cases} \dfrac{Q}{L} & (i,j) \in a_k \\ 0 & \text{other} \end{cases} \tag{16}$$

$$\Delta\tau_{ij}^{*}(t, t+1) = \begin{cases} \dfrac{Q}{L_{best}} & (i,j) \in a_k \\ 0 & \text{other} \end{cases} \tag{17}$$

In Eqs. (15)–(17), ρ represents the volatility of the pheromone, and represents the pheromone increment of the k-th ant in the time interval $[t, t+1]$ on path (i, j). Q is a constant, indicating the total amount of pheromone released by the ant after completing one iteration. L is the path length through which the ant passes, and is the current optimal path length.

The Ant Colony System has several parameters to set, and currently there is no set of common coefficient settings for different objects. The convergence speed and optimization performance of a system are a contradiction. Through optimization, the system can effectively avoid falling into the local optimal solution and improve convergence speed.

3.4. Algorithm flow

The specific implementation steps of the optimized ant colony algorithm are as follows:

Step 1: Initialize the settings. Set the initial coefficients of the ant colony algorithm.
Step 2: Randomly place the ants at the positions of the ends of each weld. Under each mathematical constraint of the weld, calculate the coordinates of the starting and ending positions of the next weld and modify the tabu table until all the welds' start and end positions are completed accessed.
Step 3: Perform collision detection on the idle line segment path after the ant selects the weld, assuming the distance between two straight segments is d. If path collision occurs, the distance between two solder joints in the no-load path is changed to $2H+ d$ (where H is the span height and D is the length of the distance the robot projects from the end of one weld to the start of other weld on a horizontal plane).
Step 4: Calculate the total length of the path traveled by each ant after accessing all the welding paths, record the welding path with the shortest number of iterations, and update the pheromone concentration on the welding path connection line.
Step 5: Determine whether the maximum number of iterations is reached. If not, return to step 2; if yes, end the procedure.
Step 6: Output the shortest path result after multiple iterations, as well as path result and convergence iteration number.

https://doi.org/10.1017/S0263574720000454 Published online by Cambridge University Press

Table I. Arithmetic parameter settings.

Particle Swarm Optimization	Total group size 60, number of iterations 200, $c_1 = 2$, $c_2 = 2$
Genetic algorithm	Total group size 60,number of iterations 200
Ant Colony Optimization	Total group size 60, number of iterations 200, $\alpha = 1$, $\beta = 5$, $p = 0.95$
Improved Ant Colony Optimization (ACO-STA)	Total group size 60, number of iterations 200, $\alpha = 1$, $\beta = 5$, $p = 0.95$

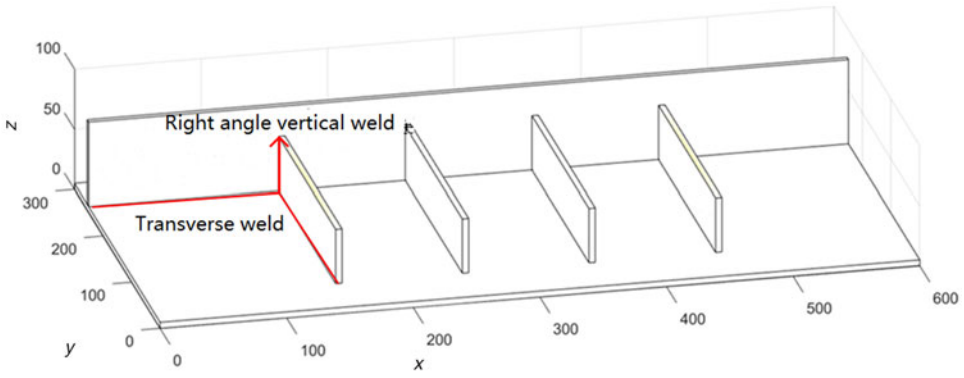


Fig. 2. 3D stereo model of ship plate components.

4. Implementation and Results Analysis Based on Improved ACO Algorithm

4.1. Experimental methods and parameter settings

Ship plate welding, which combines a vertical plate and a horizontal plate, is a common welding object in ship welding. We choose it as our simulation object. We first established a 3D model of a ship's slab as shown in Fig. 2. Then the proposed ant colony algorithm was used to optimize the welding path. The arrow marks indicate right-angled vertical welds, and welds without arrow marks are transverse welds. The right-angled weld is a directional weld with a welding direction from bottom to top; the transverse weld is a two-way weld, with the welding direction from any end of the weld.

Assuming there are $2n$ welds, the algorithm uses the combinations of coordinates of the ends of each weld for selecting the points of a welding path. The combination of the start and end points of the coordinates of two ends is fixed. For a two-way weld, the combination of start and end points of the coordinates of two ends can be randomly selected.

The parameters used in the experiment are as follows. Set the ant population to m and place the ants at the end positions of welds. The information heuristic factor α is set to 1; the expected heuristic factor is set to 5; the information volatilization factor is set to 0.2; the convergence rate control coefficient is set to 0.85; the Q constant coefficient is set to 10; the maximum iteration number is set to 200; the directional weld is reversed; the distance M between endpoints is set to 1000; the height H across the steel sheet is 60; the size of the tabu table initialization information matrix is $2n \cdot 2n$; and the path record table matrix size is $m \cdot 2n$.

4.2. Experimental comparison

To verify the performance and efficiency of the algorithm, three complex components of the ship were selected for modeling, and several experiments were carried out by ACO algorithm, PSO, GA and improved ACO algorithm. Initial parameter settings are shown in Table I.

The simulation hardware platform is Intel$^{(R)}$ Core$^{(TM)}$ i7-8550 CPU @1.80GHz. The operating system is Windows 10, and the simulation platform is Matlab2017a.

Table II shown the optimal values, average values and worst values of autonomous path planning obtained through multiple experiments with the three components. Figure 3 compares the averages of optimizations of component 3. The abscissa is the number of iterations, and the ordinate is the planned path length.

https://doi.org/10.1017/S0263574720000454 Published online by Cambridge University Press

Table II. Simulation results.

Ship welding component	Statistical method for the shortest path optimization	ACO	GA	POS	ACO-STA
Component 1	Optimal value	956.055	956.055	957.055	956.755
	Average value	957.455	957.326	957.43	957.056
	Worst value	957.655	957.655	957.856	957.655
Component 2	Optimal value	3202.454	3190.454	3190.454	3190.454
	Average value	3216.454	3212.454	3222.454	3201.454
	Worst value	3223.454	3223.454	3234.354	3223.454
Component 3	Optimal value	10275.432	10280.495	10634.483	10624.43
	Average value	10656.432	10650.432	10661.433	10639.427
	Worst value	11101.472	10975.407	10682.459	10651.425

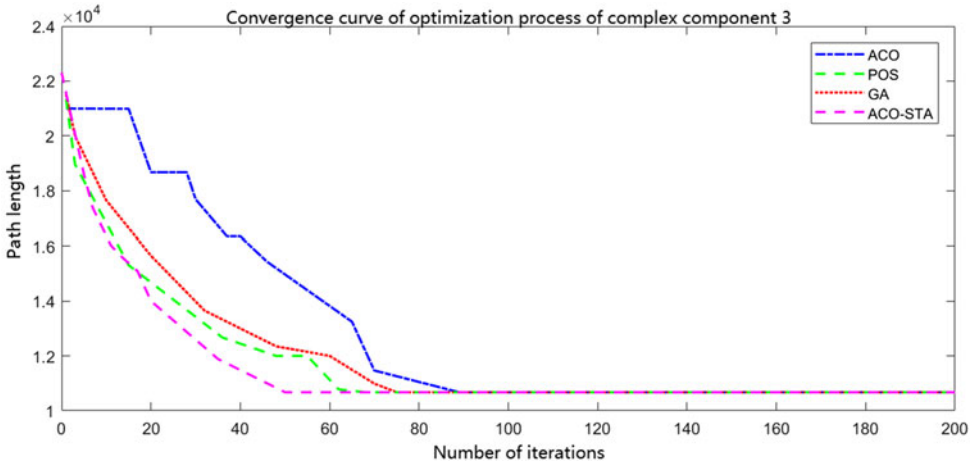


Fig. 3. Convergence curves for different complex component algorithms.

4.2.1. Comparison of path planning capabilities. When dealing with different application objects, it is usually necessary to experimentally debug the parameters to find a balance between system convergence speed and optimization performance. When the optimization object changes greatly, the performance of the optimal solution may reduce or even fall into a local optimal solution. In Table II, the optimal and worst values for components 1 and 2 using ACO algorithm are smaller than those obtained using other algorithms, but its average performance is the largest. When the complexity of the object increases (component 3), the optimization performance of ACO decreased significantly. The reason being that, since the information heuristic factor and the expected heuristic factor in the ACO algorithm were incorrectly set, the system was biased and convergence near the current optimal path was low, leading to poor stability of optimization and fluctuation in performance. Given the object was more complicated, it converged to the local optimal solution, so the algorithm did not perform well for component 3.

GA is a highly parallel, random and adaptive algorithm based on 'survival of the fittest'. The 'chromosome' group represented by the solution is replicated, crossed and mutated, eventually converging to the most suitable group so as to obtain an optimal or satisfactory solution to the problem. The performance of GA in this experiment was similar to that of ACO, and the path optimization effect fluctuated greatly. The reason being that single GA coding in the GA algorithm cannot fully express the constraints of the optimization problem, and for some complex objects, there are also problems of non-standard encoding and inaccurate encoding representation. Therefore, an optimal solution could not be obtained steadily in this experiment. GAs are generally less efficient than other traditional optimization methods, so it converged slowly.

PSO and GA have a lot in common: both initialize the population randomly; both use the fitness value to evaluate the system, and both perform random searches based on the fitness value. From Table II, we see that the PSO algorithm has the longest planned welding path in five of the nine

https://doi.org/10.1017/S0263574720000454 Published online by Cambridge University Press

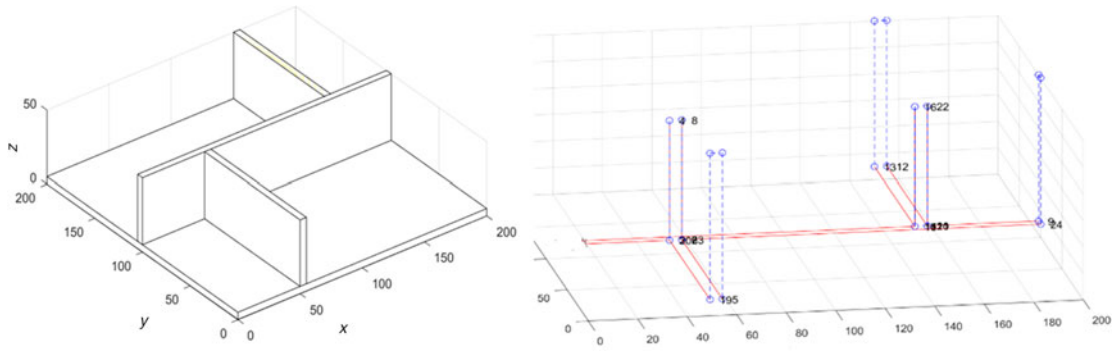


Fig. 4. Simulation results of component 1.

groups of statistical data. It can be said that PSO performed the worst in this experiment. The main reason being that PSO does not have genetic operations such as crossover and mutation, but rather determines the search at its own speed. In GA, the chromosomes share information with each other, so the movement of the entire population is relatively uniform. In PSO, only gBest (or lBest) gives information to other particles, which is a unidirectional information flow. Therefore, the operation speed may be relatively advantageous, but it does not perform well on some problems.

As shown in Table II, ACO-STA has the best performance in seven of the nine groups of statistics, in terms of the average values of the optimization path of all the components. The reason being that ACO-STA strengthens the optimal solution and accelerates the system's convergence speed. Therefore, the system can enhance the randomness of its search under the same conditions and increase the possibility of finding a global optimal solution. From these three sets of data, it can be seen that the path length planned by ACO-STA is relatively stable, showing that the optimal value, average value and worst value are not much different, so the algorithm can be highly reliable. The reason being that ACO-STA adds the upper and lower limits for dynamically updating the pheromone. Limiting the pheromone number to a dynamic and controllable range can effectively avoid stagnation of optimization.

4.2.2. Comparison of algorithm convergence speeds. As shown in Fig. 3, as the iterative operation proceeds, the path lengths planned by various algorithms decrease, and before iterating to 100 generations, all the algorithms converge to about 10,600, but the convergence speed of various algorithms is different. The order of convergence speed, from slow to fast, is ACO, GA, PSO, ACO-STA.

ACO may fall into a local optimal solution during optimization, resulting in a large change in the slope of the optimization curve and the slowest convergence rate. The algorithm iterates to the 90th generation and the results converge. The reason being that since the initial pheromone is lacking, the calculation speed is slow. GA is relatively smooth, but the optimization speed is average and the convergence is slow. It approximately iterated to the 75th generation and converges. GA has global search capabilities, but the algorithm is more complex, so the operation speed is generally slower than other intelligent algorithms. The convergence speed of PSO is relatively good. It iterates to the 62nd generation and converges, and its performance is only inferior to ACO-STA. The reason being that the particles in PSO only share the best information through the current search. To a large extent, this is a single information-sharing mechanism, so the convergence speed is faster. ACO-STA iterates to the 50th generation and converges, and the convergence speed is the fastest. Since ACO-STA no longer updates the pheromone of all ants each time but only updates the pheromone of the optimal ant in each cycle, it can effectively improve the iteration speed.

4.2.3. ACO-STA path planning capability test. Figures 4, 5 and 6 are the simulation results of AOC-STA algorithm for iterative welding of different components. The left side is a 3D model of the weld of a complex component, and the right side is the result of a robot welding path. Blue hollow dots represent the start or end solder joints of the welding gun; red line segments represent the real objects; and blue line segments represent the welding route. From the simulation diagrams on the right of Figs. 4–6, we see that the welding route planned by the algorithm can completely cover

https://doi.org/10.1017/S0263574720000454 Published online by Cambridge University Press

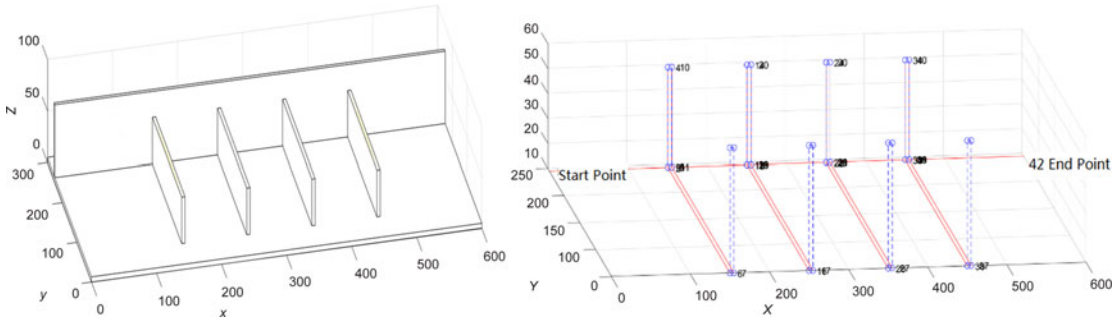


Fig. 5. Simulation results of component 2.

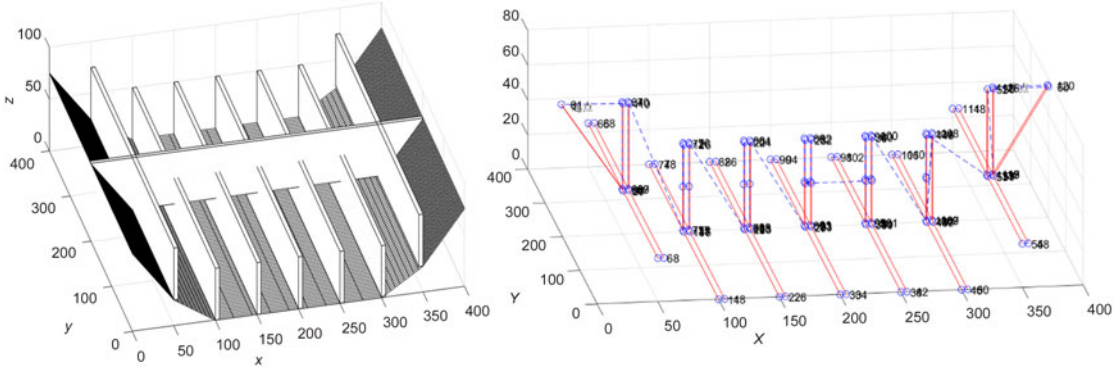


Fig. 6. Simulation results of component 3.

all the welds, and so it can effectively solve the problem of welding path planning for complex components.

5. Summary

Although some conventional components are robotically welded, their operations rely on offline programming for planning, and the flexibility may be poor. They cannot meet the welding requirements of multi-variety and large-volume complex components. It is necessary to realize autonomous intelligent planning for welding operations of diverse and complex components. However, ship welding is diverse in variety, with many types of welds, complex spatial distribution and many constraints. Existing welding planning methods are difficult to apply to such complex scenes. Based on the Ant Colony system and MMAS, a multi-parameter dynamic optimization adaptive ant colony algorithm proposed in this paper exhibited certain advantages in terms of convergence speed and optimization effect compared with other classical optimization algorithms.

Acknowledgments

This work was funded by the National Natural Science Foundation of Guangdong Joint Fund (U1801263, U1701262), Science and technology project of Guangdong Province (2015B090922013, 2016B090918045, 2017B090901019), Ministry of Education 'Blue Fire Plan (Huizhou)' (CXZJHZ201730), Guangdong Provincial Department of Education Innovation and Strong School Project (Nos. 2015KQNCX096, 2018KTSCX140), and the Science and Technology Project of Chaozhou (Nos. 2016GY29, 2015GY11).

Conflicts of Interest

The authors declare they have no conflicts of interest.

References

1. Y. Hu and Z. Meng, "Bottlenecks in the development of China's shipbuilding industry and analysis of the development strategy of "big to strong"," *Globalization* (02), 94–104+136 (2019). DOI:10.16845/j.cnki.ccieeqqh.2019.02.012.

https://doi.org/10.1017/S0263574720000454 Published online by Cambridge University Press

2. B. Hu, D. Wu and J. Yang, "Development strategy of green ship technology," *Guangdong Shipbuild.* **38**(01), 86–88 (2019).
3. Ministry of Industry and Information Technology Website, "The intelligent ship development action plan symposium was held in Dalian," *Shipbuild. Standardization Qual.* **31**(02) , 7–12 (2018).
4. M. Wu, H. Huang and X. Wang. "Robot welding path planning based on improved ant colony algorithm," *Trans. China Weld. Soc.* **39**(10), 113–118 (2018).
5. X. Wang, B. Tang and X. Gu. "Research on obstacle avoidance strategy of welding robots," *J. Mech. Eng.* **55**(17), 71–78 (2019).
6. R. Zhang, X. Li and H. Gao. "Research on collaborative path planning of dual welding robots," *Modular Mach. Tool Autom. Manuf. Tech.* (06), 81–85 (2019). DOI:10.13462/j.cnki.mmtamt.2019.06.022.
7. Z. Lin and L. Xu, "An improved ant colony optimization applied in programing laser welding path," *Trans. China Weld. Inst.* **39**(1), 107–110 (2018).
8. H. Tu and X. Xu. "Optimizing logistics distribution routing problem based on improved ant colony algorithm," *Mach. Des. Manuf.* (08), 265–268 (2017). DOI:10.19356/j.cnki.1001-3997.2017.08.074.
9. S. He, J. Shi and H. Wang, "Path planning of mobile robot based on improved ant colony particle swarm optimization algorithm," *Electr. Meas. Tech.* **34**(4), 765–770 (2014).
10. H. Wu and X. Chen, "Improved ant colony algorithm based on natural selection strategy for solving TSP problem". *J. Commun.* **34**(04), 165–170 (2013).
11. L. Xu and D. Pan, "An improved genetic ant colony algorithm for solving TSP problem," *Intell. Comput. Appl.* **7**(3), 34–36 (2017).
12. H. Du and Y. Li, "Research on affect performance of parameter settings in ant colony algorithm," *Mod. Comput.* **13**, 3–7 (2012).
13. X. Wei and Y. Li. "Research on parameters optimization and simulation of the ant colony algorithm," *Manuf. Autom.* **37**(10), 33–35 (2015).
14. M. Dorigo, V. Maniezzo and A. Colorni, "The ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst. Man Cybern.* **26**(01), 29–41 (1996).
15. K. H. Kim and K. C. Moon, "Berth scheduling by simulated annealing," *Transp. Res. Part B* **37**, 541–560 (2003).
16. T. Stutzle and H. H. Hoos, "MAX-MIN ant system," *Future Gener. Comput. Syst.* **16**(9), 889–914 (2000).
17. M. Dorigo and L. M. Gambardella, "Ant colony system : A cooperative learning approach to the traveling salesman problem," *IEEE Trans. Evolut. Comput.* **1**(1), 53–66 (1997).

https://doi.org/10.1017/S0263574720000454 Published online by Cambridge University Press