IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 17, 2018, accepted October 26, 2018, date of publication October 30, 2018,
date of current version November 30, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2878615

Research on Intelligent Welding Robot Path Optimization Based on GA and PSO Algorithms

TONG YIFEI, ZHONG MENG, LI JINGWEI, LI DONGBO, AND WANG YULIN
School of Mechanical Engineering, Nanjing University of Science & Technology, Nanjing 210094, China

Corresponding author: Wang Yulin (38336298@qq.com)

This work was supported in part by the Prospective Joint Research Project of Jiangsu Province under Grant BY2015004-05, in part by the Jiangsu Provincial Six Talent Peaks Project under Grant 2015-ZBZZ-006, in part by the Science and Technology Achievements Transformation of Jiangsu Province under Grant 2015-67, and in part by the High Local Industrial Robots and Intelligent Equipment Development and Industrialization, Nanjing PANDA Electronics Equipment Co., Ltd.

ABSTRACT To make the welding robot more reasonable and furthermore improve the productivity and reduce costs, two intelligent algorithms for welding path optimization, genetic algorithm (GA) and discrete particle swarm optimization, are proposed to optimize the welding robot path. Through the improved selection of the operator, the GA achieves the fastest iterative efficiency. By introducing the ‘‘swap operator’’ and ‘‘swap sequence’’ in the particle swarm optimization algorithm, the PSO algorithm is improved for the solution of the discrete problem (welding robot path planning) which is superior to the continuous optimization problem. Besides, for the better iterative efficiency of PSO, the parameters of traditional inertia weight are determined by a linear inertia weigh, which can improve the convergence performance of the algorithm. The modeling and solutions of the two algorithms are discussed in detail to illustrate the applications in the welding robot path optimization. In order to compare the pros and cons of the two algorithms, the same welding tasks are presented, and Matlab simulation is carried out. The simulation results show that both genetic algorithm and particle swarm optimization algorithm can obtain the optimal or near-optimal welding path by iterative calculations.

INDEX TERMS Discrete particle swarm optimization algorithm, genetic algorithm, Matlab, welding path optimization, welding robot.

I. INTRODUCTION

Welding robots play a significant role in industrial production. In a large-scale welding operation, a lot of welding joints need to weld with irregular distribution. The welding path will directly influence the efficiency of robots [1]. Path planning of robots, is to seek the optimal or suboptimal path of a robot from the initial position to the target position with some optimization criteria (such as the minimum of working costs, shortest route, shortest running time, obstacles avoidance etc.). According to the start-stop movement in drilling/spot welding task, the path planning problem can be converted into a traveling salesman problem (TSP) [2]. Home and abroad experts have made significant studies of TSP problems. Goldberg first applied genetic algorithms (GAs) for TSP, and achieved a short tour [3]. Wang [4] used natural number coding genetic algorithms for TSP and found the global optimum. A new crossover mechanism called Same Adjacency (SA) for GA with variable-length chromosomes for path optimization problem was proposed, which can outperform GA with SP by a better search capability as the mathematical analysis shows [5]. Kennedy and Eberhart [6] firstly proposed a Discrete Particle Swarm Optimization (DPSO) to solve combinatorial optimization problems of engineering practice. Clerc [7] promoted researches of DPSO for solving TSP with the definition of velocity as a ‘‘swap sequence’’ as well as other variables and rules, and achieved good results. Çunkaş and Özsaglam [8] redefined PSO operators for TSP problems by ‘‘Swap operator’’ and ‘‘Swap sequence’’. To solve the problem of premature, a new PSO algorithm by combining the random learning mechanism and Levy flight was developed, which can increase the diversity of the population by learning from random particles and random walks in Levy flight [9].

For solving urban search and rescue problems or some special work, the robot is used widely for favor. Zhang *et al.* [10] a proposed a multi-robot cooperation method based on niching particle swarm optimization for multiple odor sources localization. Geng *et al.* [11] presented a modified centralized algorithm based on particle swarm optimization (MCPSO) to solve the task allocation problem in the search and rescue domain, which provides a benchmark against distributed algorithms in search and rescue application area. Gong *et al.* [12] proposed a modified particle swarm optimization (PSO) algorithm for robots cooperation to search for an odor source. The difficulties of these researches lie in the 3D planning, obstacle avoidance and collaboration. For the research presented here, due to that the welded workpiece is large thin-walled part (the dimension is small in one direction), the path planning can be considered as TSP problem, which is a 2D planning. Therefore, the 3D planning, obstacle avoidance and collaboration will not be considered for single robot welding.

Besides, different from the researches, which focused on robot joint control belonging to trajectory planning, this paper focuses on path planning, emphasis on the trajectory of the welding robot's end gun instead of the trajectory of each joint [13]. Two bionic optimization algorithms, genetic algorithm and particle swarm optimization algorithm, are researched for robot path optimization. For better iterative efficiency and performance, the operator selection of GA is improved as well as the PSO algorithm with improved inertia weight determination, and the "swap operator" and "swap sequence" introduced into the solution of the discrete problem (welding robot path planning). In the next section, the welding robot path planning problem and mathematical model are described in detail. Welding path optimization based on the genetic algorithm and Discrete Particle Swarm Optimization are discussed separately in Sects. 3 and 4. Comparisons of the two optimization algorithms for welding robot path optimization is presented in Sects. 5. Finally, the conclusion is summarized.

II. PROBLEM DESCRIPTION AND MODELING

A. PROBLEM DESCRIPTION

Welding path planning aims to search a shortest path through each joint once and only once in the robot working area.

Suppose that a set $V = \{v_1, v_2, \cdots, v_n\}$ contains n different joints to be welded, and a set $A = \{a_{ij} | v_i, v_j \in V\}$ represents the distance between any two welding joints of V. Path optimization problems in directed graph $G = (V, A)$, means to seek the shortest Hamiltonian circuit of a integer subset $X = \{x_1, x_2, \cdots, x_n\}$ (X represents a full array of the n different points) with a minimum total path length of the robot. That is the path length function $f_x = \sum_{i=1}^{n-1} d(x_i, x_{i+1}) + d(x_n, x_1)$ obtains the minimum, where $d(x_i, x_j)$ represents the distance of the welding joint from p_i to p_j [14].

Usually the shortest welding path can be determined empirically when with less welding spots. However, with the increased welding spots, it is difficult to make a combinatorial explosion. Therefore, it is of great importance to utilize some intelligent algorithms to rapidly find the optimal path.

B. MATHEMATICAL MODELING

In this research, welding path optimization can be described as: Robots need to weld n spots with known position, under rational planning welding sequences with the shortest welding path length. Below are some constraints [15]:

(1) The welding path with the same starting point and end;

(2) Each welding spot must be welded and only once.

The proposed mathematical model can be expressed as follows:

$$\min \sum_{i \neq j} d_{ij} x_{ij} \tag{1}$$

$$\text{s.t.} \sum_{j=1, j \neq i}^{n} x_{ij} = 1, \quad i = 1, 2, 3, \cdots, n \tag{2}$$

$$\sum_{i=1, i \neq j}^{n} x_{ij} = 1, \quad j = 1, 2, 3, \cdots, n \tag{3}$$

$$\sum_{i,j \in s} x_{ij} \leq |s| - 1, 2 \leq |s| \leq n, \quad s \subset \{1, 2, 3, \cdots, n\} \tag{4}$$

$$x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, 3, \cdots, n, \ i \neq j \tag{5}$$

Where d_{ij} denotes the distance between the welding spot i and j, x_{ij} is the bound variable (1 for the welding path from the welding spot i to j and 0 for the welding path not from the welding spot i to j), s denotes the set of joints which have been welded, and $|s|$ denotes the number of elements in the set s.

III. WELDING PATH OPTIMIZATION BASED ON THE GENETIC ALGORITHM

A. ENCODING AND INITIAL POPULATION

In general, the encoding of GAs for solution space is mostly binary-coded [16]. However, due to robot path planning are permutation problems, binary-coded expression requires special patching. Because a single bit change may result in an illegal path, nature number code (a chromosome gene that represents the welding path is the sequence of the welding joint in the path.) is selected [4].

B. FITNESS FUNCTION

For the welding robot path optimization, the evaluation function is the length of each legitimate weld path. The smaller the length, the individual is better [17].

So, the fitness function is defined as:

$$f(x) = D(s) \tag{6}$$

$$D(s) = \sum_{i=1}^{n-1} d(v_i, v_{i+1}) + d(v_n, v_1) \tag{7}$$

C. SELECTION

Selection is the process where individuals are selected based on their fitness values and generated offspring. Here we used the tournament method, so that the new generation is more excellent, and thus can accelerate the convergence speed [18].

D. CROSSOVER

In order to make sure that legitimate individual can be generated during cross process, it is required that chromosome coding of each path not duplicate. That is, each joint must and can only be welded once [19]. So crossover which is suitable for traditional genetic algorithms may not fully be applicable here. Because the partial mapping crossover (PMX), sequential crossover (OX), and cyclic crossover (CX) have two disadvantages. One is that the cross-overspring do not retain the good genes of the parent and the other is that the children may need some changes to be feasible, which will affect the convergence effect and calculation speed. While single point crossing (OC) refers to the exchange of a part of chromosomes of two paired individuals at a point after randomly setting an intersection point in an individual code string. It can satisfy the convergence effect and guarantees the operation speed. Therefore, OC is chosen as the crossover operator [20]. Therefore, the OC is adopted for crossover as follows:

Parent1: 1 2 3 |4 5 6| 7 8 9

and

Parent2: 6 8 7 |1 3 9| 5 0 4 2

(Parent1 and Parent2 are two chromosomes to be cross-operated.)

The joints in the selected substring in Parent1(here 4, 5, and 6) are firstly replaced by "$*$" in the receptor Parent2.

Parent1: 1 2 3 |4 5 6| 7 8 9 0

and

Parent2: $*$8 7 |1 3 9| $*$ 0 $*$ 2

To preserve the relative order in Parent2 (the receiver), a sliding motion is made to leave the holes in the matching section marked in the receiver. In this research, start this sliding motion in the second crossover site, so after the rearrangement we have:

Parent1: 1 2 3 |4 5 6 |7 8 9 0

and

Parent2: 1 3 9 | $* * *$ |0 2 8 7

After that, the three stars are replaced with the joint taken from the donor Parent1 in the Offspring of Parent2(Offspring2).

Offspring2: 1 3 9 |4 5 6|0 2 8 7

The similarly operation can produce the Offspring of Parent1(Offspring1):

Offspring1: 4 5 6|1 3 9|7 8 0 2

As can be seen, the generated offspring are legitimate weld paths by OC without repetition of welding joints.

E. MUTATION

Swap mutation is used for operations. Randomly select two bits in one chromosome and swap [21]. Thus, we still have legitimate path after swap mutation.

F. THE OPTIMIZATION PROCEDURE FOR WELDING PATH BASED ON THE GA

Input: maximum iterations (n), population size (m), selection probability (Ps), crossover probability (Pc) and mutation probability (Pm).

Output: The optimal path and fitness function evolution curve.

Step 1 (Initialization): Randomly generate an initial population (m denotes the number of the chromosomes, N denotes the number of genes in each chromosome representing a welding path;

Step 2: Evaluate the fitness of each chromosome(each path). If the termination condition of iterations is satisfied, end the GA calculation and go to Step 6. Else if, continue the next step;

Step 3: Selection with Ps;

Step 4: Crossover with Pc;

Step 5: Mutation with Pm. Retain the new offspring and go to Step 2;

Step 6: Output the optimal path and fitness function evolution curve.

The flow chart is shown in Fig. 1.

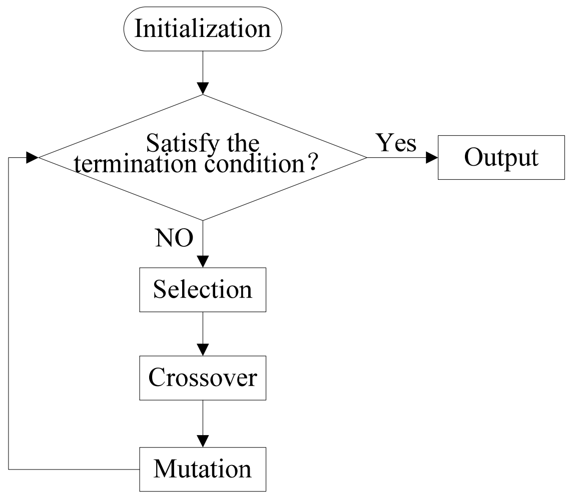


FIGURE 1. The flow chart of the GA.

G. SIMULATION RESULTS AND ANALYSIS

Generally, a work station's welding spots of a production line are hundreds, however, taking into account the welding robot working area and the efficiency of entire production line, often arrange multiple robots in a welding workstation and simultaneously operating, so welding points of each welding robot are not too many, average 30 or so, so the experimental task of this thesis also selects 30 welding joints. As shown in Fig. 2, it is 30 joints welding joint coordinates.

To validate the application of the genetic algorithm, in this paper, Matlab programs are run for simulation. Use natural number coding to produce M chromosomes with n genes. Then, decide whether or not to terminate the program based on the termination condition. If the termination condition is satisfied, the program is end and the optimal path and fitness function curve are output. If the condition of termination has not been met, proceed to the next round iteration (selection, crossover and mutation). As shown in Fig. 1.

The algorithm parameters are set as follows:

population size: m = 50,

selection probability: Ps = 0.3,

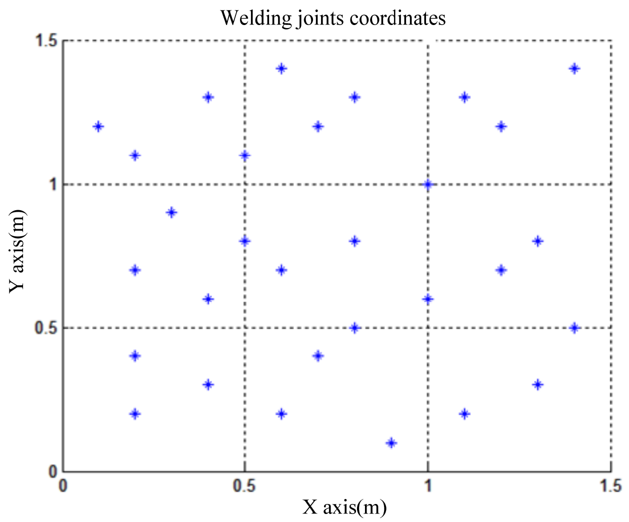


FIGURE 2. Welding joints coordinates.

crossover probability: Pc = 0.8,
mutation probability: Pm = 0.05,
termination iterations: n = 30, 50, 100, 200, 300, 400.

Fig. 2 shows the welding joint coordinates for study. Repeat the simulation 50 times, and the statistical results of solving are presented in Table 1.

When the number of iterations is 400, GAs get the optimal path and fitness function evolution curve, as shown in Fig. 3 and Fig. 4

The optimal path length is 10.4628m, and the average operation time was 8.9305s when the number of iterations is 400 as listed in Table 1.

IV. WELDING PATH OPTIMIZATION BASED ON DISCRETE PARTICLE SWARM OPTIMIZATION

A. DISCRETE PARTICLE SWARM OPTIMIZATION(DPSO)

Suppose a D-dimensional search space composed of m particles, where the ith particle denotes with a D-dimensional position vector $X_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$ and a D-dimensional velocity vector $V_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$ $(i = 1, 2, \cdots, m)$. $P_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$ denotes the position of the present optimal value $pBest$ of the ith and $P_g = (p_{g1}, p_{g2}, \cdots, p_{gD})$ denotes the position of the global optimal value $gBest$ of the whole particle swarm. Fitness function is expressed as $f(x)$. For ease of discussion, assume the optimization as a minimum problem, that is to determine the length of the shortest Hamilton Circle[22].

The current best position of the particle i is determined by:

$$p_i^{k+1} = \begin{cases} p_i^k & if\ f(X_i^{k+1}) > f(p_i^k) \\ X_i^{k+1} & if\ f(X_i^{k+1}) \le f(p_i^k) \end{cases} \tag{8}$$

The best position of the global population is determined by:

$$p_g^k = \arg \min_{|f(p_1^k), f(p_2^k), \cdots, f(p_m^k)|} \{f(p_1^k), f(p_2^k), \cdots, f(p_m^k)\} \tag{9}$$

After finding the two optimal positions, update the particles' velocity and position according to the following two formulas:

$$v_{id}^{k+1} = \omega v_{id}^k + c_1\, rand(\,)\, (p_{id} - x_{id}^k) + c_2\, rand(\,)\, (p_{gd} - x_{id}^k) \tag{10}$$

$$x_{id}^{k+1} = x_{id}^k + v_{id}^{k+1} \tag{11}$$

Where $i = 1, 2, \cdots, m$ denotes the sequence number of particles, $d = 1, 2, \cdots, D$ denotes the dimension of the search space, k denotes the number of iterations, V_{id}^k denotes the d-dimensional velocity of the ith particle in the kth iteration, ω is the non-negative inertia weight, c_1 and c_2 are constant acceleration coefficients, $rand(\,)$ is a function to generate random number in the interval (0, 1). There is a maximum $V_{\max}$ to limit the particle speed, in favor to make the PSO algorithm achieve the best search capability by regulation ω. Termination conditions for the iterations are generally selected to a predetermined maximum number of iterations.

TABLE 1. Comparisons of the two optimization algorithms.

	GA				PSO			
Iterations	The optimal path length (m)	The worst path length (m)	Average path length (m)	Average running time (sec)	The optimal path length (m)	The worst path length (m)	Average path length (m)	Average running time (sec)
30	14.8342	17.7672	16.0105	0.2412	16.3176	19.6230	18.4705	0.1346
50	14.0065	17.0153	15.0426	0.3547	14.2153	19.1413	17.4079	0.1813
100	12.3872	14.5739	13.3592	0.8286	11.3840	13.7756	12.3239	0.3004
150	11.5057	14.6881	12.9398	1.5270	10.5422	13.4667	11.8835	0.4092
200	11.2477	14.3232	12.6854	2.3773	10.3438	13.3954	11.4606	0.5213
300	10.8465	13.7317	12.5452	5.1948	10.3438	13.2338	11.5646	07606
400	10.4628	13.6826	12.5514	8.9305	10.3235	13.0373	11.8829	0.9801

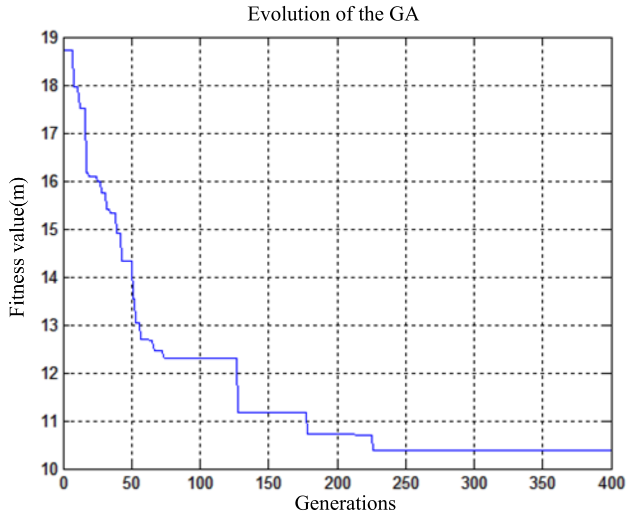


FIGURE 3. Evolution curve of the GA.

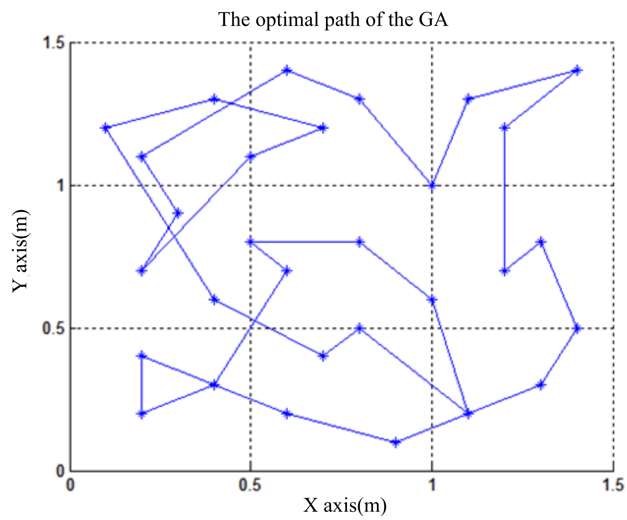


FIGURE 4. The optimal path of the GA.

The PSO is originally used for continuous space optimization problems and achieved good results[23]. But for non-continuous combinatorial optimization problems, the traditional PSO may not be fully applicable. Therefore, Discrete Particle Swarm Optimization(DPSO) is evolved from PSO to solve discrete optimization problems (such as TSP problems)[24].The DPSO has no crossover and mutation operations of GA, and relies on particle velocity to search with fast convergence speed, short time and fewer parameters, which is easy to adjust and remain stable.

B. THE DPSO FOR WELDING ROBOT PATH OPTIMIZATION

The discrete particle swarm optimization was reported to solve TSP problems with definitions of "swap operator" and "swap sequence" for the velocity and position respectively [8]. In our research, the DPSO is utilized to optimize the welding robot path. The core of the DPSO algorithm is that each particle corresponds to a random arrangement of spots position, representing a legitimate weld path.

C. SWAP OPERATOR [8]

Suppose the path optimization problem has n weld joints. One solution is $S = (a_1, a_2, \cdots, a_n)$. Define the swap operator $SO\,(i, j)$ indicates the exchange position of joint a_i and a_j in the solution S in order to generate a new solution sequence S'. That is $S' = S + SO(i, j)$, where the symbol "+" denotes the exchange operation.

D. SWAP SEQUENCE [8]

The swap sequence SS is defined as ordered arrangement of one or more swap operators.

$SS = (SO_1, SO_2, \cdots, SO_n)$, Where $SO_1, SO_2, \ldots, SO_n$ are swap operators, the order of which are very important.

E. THE RELATIVE CONCEPTS OF SWAP SEQUENCE [8]

The swap sequence acts on a solution sequence, indicating that all the swap operators in the swap sequence are acting on the solution in order, which can be described by (12):

$$\begin{aligned} S' &= S + SS = S + (SO_1, SO_2, \cdots, SO_n) \\ &= [(S + SO_1) + SO_2] + \cdots + SO_n \end{aligned} \tag{12}$$

Different swap sequences acting on the same solution may produce the same new solution, which can be called the equivalent set. Two or more swap sequence can be combined into a new swap sequence, which defined as the merge operator of these swap sequences by "$\oplus$".

For instance, two swap sequences SS_1 and SS_2 act on the solution sequence S in order, and get a new solution S'. Suppose another swap sequence SS', acting on the same solution S, the same new solution sequence S' can be obtained, and then define:

$$SS' = SS_1 \oplus SS_2$$

Here SS' and $(SS_1 \oplus SS_2)$ belong to the same equivalence set. In general, SS' is not unique.

In equivalent sets of swap sequences, the sequence with the minimum number of swap operators is defined as the basic swap sequence of equivalent sets. The basic swap sequence can be constructed as follows:

Given two paths (A and B), need to construct a basic swap sequence SS, which can satisfy $B + SS = A$.

For example: $A = (123456)$ and $B = (426153)$.

As can be seen, $A(1) = B(4) = 1$.

So the first swap operator is $SO_1(1, 4)$, and then $B_1 = B + SO_1(1, 4) = (126453)$. $A(2) = B_1(2) = 2$, so $B_1(2)$ doesn't need to exchange. $A(3) = B_1(6) = 3$, so the second swap operator is $SO_2(3, 6)$ and $B_2 = B_1 + SO_2(3, 6) = (123456)$. Because $B_2 = A$, no additional swap operators is needed.

Thus, get a basic swap sequence:

$$SS = A - B = (SO_1(1, 4), SO_2(3, 6))\,.$$

According to the above analysis, the velocity (10) is not suitable for welding robot path optimization, so it is necessary to reconstruct it as :

$$v_{id}^{k+1} = \omega v_{id}^{k} \oplus c_1 rand(\,)(p_{id} - x_{id}^{k}) \oplus c_2 rand(\,)(p_{gd} - x_{id}^{k}) \tag{13}$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \tag{14}$$

Where ω is the inertia weight, and larger ω is beneficial to jump out of local minimum, while smaller ω is in favor of convergence. So it is important to propose a linear inertia weight to improve the convergence performance of the algorithm [25], [26]. Here let $\omega = \omega_{\max} - \frac{(\omega_{\max}-\omega_{\min})}{k_{\max}} \times k$, where k is the current iteration number, $k_{\max}$ is the maximum number of iterations, $\omega_{\max}$ and $\omega_{\min}$ respectively are the maximum and minimum inertia weight factor[27]. c_1 and c_2 are random numbers in the interval (0, 1), *rand*( ) is a random number $\in$ (0, 1) generating function.

F. THE OPTIMIZATION PROCEDURE FOR WELDING PATH BASED ON THE DPSO

Input: maximum iterations(n), swarm size(m), inertia weight(ω), and learning coefficients (c_1, c_2).

Output: The optimal path and fitness function evolution curve.

Step 1 *Initialization:* Each of the particles gets a random solution/path and a random swap sequence(velocity);

Step 2: Evaluate the fitness value of each particle. If the termination condition of iterations is satisfied, end the calculation and go to Step 4;

Step 3: For all the particles in the position X_{id}^{k}, calculate the next position X_{id}^{k+1};

Step 3.1: Calculate $A = (p_{id} - x_{id}^{k})$ and $B = (p_{gd} - x_{id}^{k})$ (A and B are basic swap sequence);

Step 3.2: According to (13), convert the basic swap sequence v_{id}^{k} to the basic swap sequence v_{id}^{k+1};

Step 3.3: Calculate a new solution based on the (14);

Step 3.4: According to (8) to update the optimal position of a single particle searched, and according to (9) to update global optimal position, and then go to Step 2;

Step 4: Output the optimal path and fitness function evolution curve.

The flow chart is shown in Fig. 5.

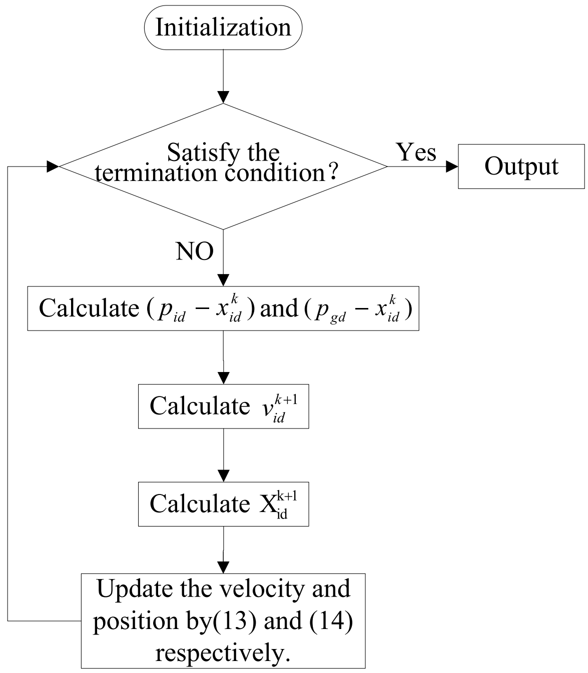


FIGURE 5. The flow chart of the DPSO.

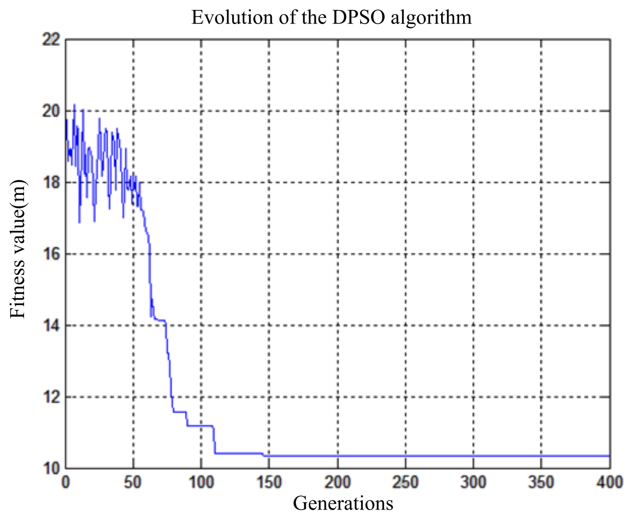


FIGURE 6. Evolution curve of the DPSO algorithm.

G. SIMULATION AND RESULTS ANALYSIS

In order to verify the DPSO for solving welding robot path optimization, Matlab programs are also run by simulation as shown in Fig. 5, using the data in Fig. 2 and the fitness function of (6). Use natural number coding to randomly generate a legitimate welding path for each particle and produce a switching sequence for the speed. Then, decide whether or not to terminate the program based on the termination condition. If the termination condition is satisfied, the program is end and the optimal path and fitness function evolution curve are output. If the termination condition has not been satisfied, proceed to the next round iteration calculation: calculate according to (13), and according to (14). Then update the individual optimal path according to (8), and the global optimal path according to (9). After an iteration round is finished, repeat step 2-3 in Section 4.6.

The parameters of the DPSO are set as follows:
population size: m = 50,
learning coefficients: $c_1 = 0.5$, $c_2 = 0.7$,
inertia weight: $\omega_{\max} = 0.9$, $\omega_{\min} = 0.4$,
maximum iterations: n = 30, 50, 100, 200, 300, 400.

Fig. 2 shows the welding joint coordinates for study. Repeat the simulation 50 times, and the statistical results of solving are shown in Table 1.

When the number of iterations is 400, DPSO algorithm get the optimal path and fitness function curve, as in Fig. 6 and Fig. 7.

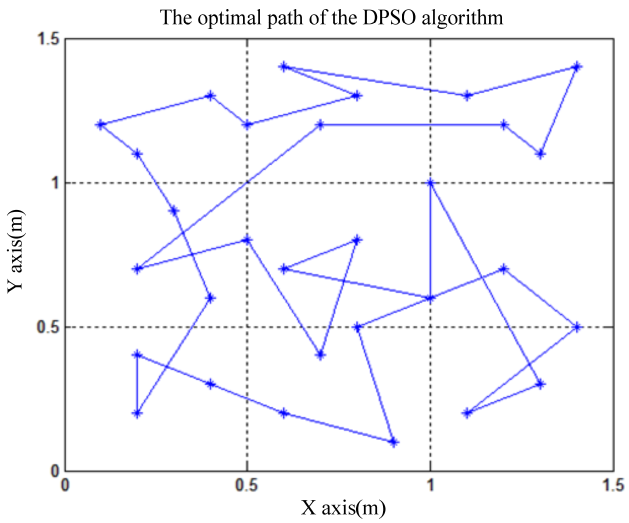


FIGURE 7. The optimal path of the DPSO algorithm.

When the number of iterations is 400, the PSO found the optimal path shown in Fig. 7.

V. COMPARISONS OF THE TWO OPTIMIZATION ALGORITHMS FOR WELDING ROBOT PATH OPTIMIZATION

The optimal path length is 10.3235m, and the average operation time was 0.9801s when the number of iterations is 400, as listed in Table 1. In order to better understand the two optimization algorithms, the experimental data for the two algorithms are listed and compared in Table 1.

As can be seen from Table 1, both the genetic algorithm and particle swarm optimization algorithm can find the optimal solution or near-optimal solution, whose average path length value is 12.5514m and 11.8829m respectively when iterations are 400, and the optimal values are 10.4628m and 10.3235m respectively. The difference in the two optimal path length probably results from inadequate iterations times or calculation errors, etc. From the view of the running time, the DPSO algorithm computation time is shorter than the GA after the same iterations, indicating the DPSO algorithm has higher efficiency and especially better search performance for more iterations.

VI. CONCLUSION

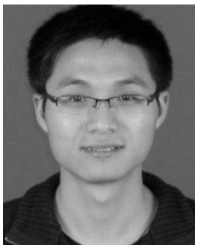
Application of the Discrete Particle Swarm Optimization algorithm to welding robot path optimization is a relatively new attempt. In this paper, the genetic algorithm and discrete particle swarm optimization algorithm are applied to the welding robot path optimization, and the effectiveness of the two algorithms are verified through simulation. The results show that both the two algorithms show a good convergence and optimization capability for welding path optimization. Besides the above analysis, why the DPSO has better efficiency and search performance can be concluded that: ① The DPSO algorithm need no genetic operations (selection, crossover and mutation) and just use the random velocity to change the individual' s search direction, which can result in lower computational complexity; ② Particles of the PSO algorithm have a "memory" feature, and the optimal solution or near optimal solution can be obtained within a less iterations by "self" learning and learn from "groups"; ③ The entire population of GA is relatively homogeneous closer to the optimal solution with information sharing. But in the PSO algorithm, only the global optimal solution and individual optimal solution will pass information to other particles in a one-way flow.

Although it is common for industrial applications to consider the robot path planning as TSP problem, the welding robot trajectory is planned in two-dimensional space with reduced computational complexity, which is not exactly the same as the actual situation. The next steps are to study 3D path planning for large size so as to be in line with industrial application scenarios.

REFERENCES

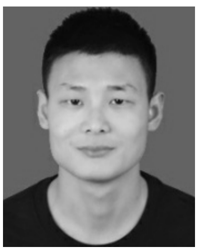
[1] J. N. Pires, "Welding robots: New trends and developments," *Ind. Robot Int. J.*, vol. 32, no. 4, p. 297, 2013.
[2] Q. Zhang and M.-Y. Zhao, "Minimum time path planning of robotic manipulator in drilling/spot welding tasks," *J. Comput. Des. Eng.*, vol. 3, no. 2, pp. 132–139, 2016.
[3] D. E. Goldberg and J. R. Lingle, "Alleles, loci, and the traveling salesman problem," in *Proc. 1st Int. Conf. Genetic Algorithms Appl.*, 1985, pp. 154–159.
[4] W. Hui, "Comparison of several intelligent algorithms for solving TSP problem in industrial engineering," *Syst. Eng. Procedia*, vol. 4, no. 1, pp. 226–235, 2012.
[5] Z. Qiongbing and D. Lixin, "A new crossover mechanism for genetic algorithms with variable-length chromosomes for path optimization problems," *Expert Syst. Appl.*, vol. 60, pp. 183–189, Oct. 2016.
[6] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. Comput. Simulation*, vol. 5, Oct. 1997, pp. 4104–4108.
[7] M. Clerc, "Discrete particle swarm optimization, illustrated by the traveling salesman problem," in *New Optimization Techniques in Engineering*, vol. 547, no. 1. Berlin, Germany: Springer, 2004, pp. 219–239.
[8] M. Çunkaş and M. Y. Özsaglam, "A comparative study on particle swarm optimization and genetic algorithms for traveling salesman problems," *Cybern. Syst., Int. J.*, vol. 40, no. 6, pp. 490–507, Aug. 2009.
[9] B. Yan, Z. Zhao, Y. Zhou, W. Yuan, J. Li, and J. Wu, "A particle swarm optimization algorithm with random learning mechanism and Levy flight for optimization of atomic clusters," *Comput. Phys. Commun.*, vol. 219, pp. 79–86, Oct. 2017.
[10] J. Zhang, D. Gong, and Y. Zhang, "A niching PSO-based multi-robot cooperation method for localizing odor sources," *Neurocomputing*, vol. 123, pp. 308–317, Jan. 2014.
[11] N. Geng, Q. Meng, D. Gong, and P. Chung, "How good are distributed allocation algorithms for solving urban search and rescue problems? A comparative study with centralized algorithms," *IEEE Trans. Automat. Sci. Eng.*, to be publised.
[12] D.-W. Gong, Y. Zhang, and C.-L. Qi, "Localising odour source using multi-robot and anemotaxis-based particle swarm optimisation," *IET Control Theory Appl.*, vol. 6, no. 11, pp. 1661–1670, Jul. 2012.
[13] W. He, Y. Dong, and C. Sun, "Adaptive neural impedance control of a robotic manipulator with input saturation," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 3, pp. 334–344, Mar. 2016.
[14] Z. H. Ahmed, "Improved genetic algorithms for the travelling salesman problem," *Int. J. Process Manage. Benchmarking*, vol. 4, no. 1, pp. 109–124, 2014.
[15] A. P. Adewole, K. Otubamowo, and T. O. Egunjobi, "A comparative study of simulated annealing and genetic algorithm for solving the travelling salesman problem," *Int. J. Appl. Inf. Syst.*, vol. 4, no. 4, pp. 6–12, 2012.
[16] X. Yan, C. Zhang, and W. Luo, "Solve traveling salesman problem using particle swarm optimization algorithm," *Int. J. Comput. Sci. Issues*, vol. 9, no. 6, pp. 264–271, 2012.

[17] S. F. Ghannadpour, S. Noori, R. Tavakkoli-Moghaddam, and K. Ghoseiri, "A multi-objective dynamic vehicle routing problem with fuzzy time windows: Model, solution and application," *Appl. Soft Comput.*, vol. 14, no. 1, pp. 504–527, Jan. 2014.
[18] G. Dong, X. Fu, and V. Xue, "An ant system-assisted genetic algorithm for solving the traveling salesman problem," *Int. J. Advancements Comput. Technol.*, vol. 4, no. 5, pp. 165–171, 2012.
[19] Y. Deng, Y. Liu, and D. Zhou, "An improved genetic algorithm with initial population strategy for symmetric TSP," *Math. Problems Eng.*, vol. 2015, no. 3, 2015, Art. no. 212794.
[20] M. T. Bhoskar *et al.*, "Genetic algorithm and its applications to mechanical engineering: A review," *Mater. Today, Proc.*, vol. 2, nos. 4–5, pp. 2624–2630, 2015.
[21] M. Wang, K. Liu, and Z. L. Jiang, "Research on vehicle routing problem models and algorithms of batteries logistics distribution for electric vehicle," *Adv. Mater. Res.*, vols. 791–793, pp. 1409–1414, Sep. 2013.
[22] R. Eswarawaka, S. K. N. Mahammad, and B. E. Reddy, "Genetic annealing with efficient strategies to improve the performance for the NP-hard and routing problems," *J. Exp. Theor. Artif. Intell.*, vol. 27, no. 6, pp. 779–788, 2015.
[23] W. Ding, C.-T. Lin, M. Prasad, Z. Cao, and J. Wang, "A layered-coevolution-based attribute-boosted reduction using adaptive quantum-behavior PSO and its consistent segmentation for neonates brain tissue," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 3, pp. 1177–1191, Jun. 2017.
[24] J. Ding and Y. Zhang, "A discrete particle swarm optimization algorithm for gate and runway combinatorial optimization problem," *Res. J. Appl. Sci., Eng. Technol.*, vol. 5, no. 10, pp. 1842–1856, 2013.
[25] A. Rathore and H. Sharma, "Review on inertia weight strategies for particle swarm optimization," in *Proc. 6th Int. Conf. Soft Comput. Problem Solving*, vol. 546, 2017, pp. 76–86.
[26] J. C. Bansal, P. K. Singh, M. Saraswat, A. Verma, S. S. Jadon, and A. Abraham, "Inertia weight strategies in particle swarm optimization," in *Proc. 3rd World Congr. Nature Biologically Inspired Comput.*, Salamanca, Spain, Oct. 2011, pp. 633–640.
[27] M. A. Arasomwan and A. O. Adewumi, "On the performance of linear decreasing inertia weight particle swarm optimization for global optimization," *Sci. World J.*, vol. 2013, pp. 1–12, 2013, Art. no. 860289, doi: 10.1155/2013/860289.



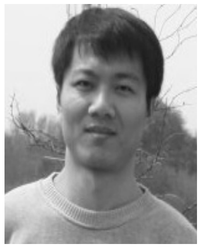
TONG YIFEI was born in Xinghua, China, in 1981. He received the B.S. and Ph.D. degrees in mechanical engineering from the Nanjing University of Science and Technology, Nanjing, in 2003 and 2008, respectively.

From 2008 to 2012, he was a Research Assistant with the School of Mechanical Engineering, Nanjing University of Science and Technology, where he has been an Associate Professor since 2012. From 2013 to 2014, he was a Visiting Scholar at Purdue University, West Lafayette, IN, USA. He has authored over 60 articles. His research interests include advanced manufacturing system and industrial engineering.



ZHONG MENG was born in Suqian, China, in 1995. He received the B.S. degree in mechanical engineering from the Taiyuan University of Science and Technology, Taiyuan, in 2017. He is currently pursuing the M.S. degree in mechanical engineering with the School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing, China.

His research interests include computer aided engineering, large crane structure damage and failure assessment techniques, and intelligent manufacturing.



LI JINGWEI was born in Luoyang, China, in 1990. He received the B.S. degree in mechanical engineering from the Zhengzhou University of Aeronautics, Zhengzhou, in 2013, and the M.S. degrees in mechanical engineering from the Nanjing University of Science and Technology, Nanjing, in 2017, where he is currently pursuing the Ph.D. degree in mechanical engineering with the School of Mechanical Engineering, Nanjing, China.

His research interests include intelligent manufacturing, process monitoring, big data analysis and processing, and service quality evaluation of production line.

LI DONGBO was born in 1957. He received the B.S., M.S., and Ph.D. degrees in mechanical engineering from the Nanjing University of Science and Technology, Nanjing, in 1982, 1984, and 1993, respectively.

From 1982 to 1984, he was a Research Assistant at the Beijing Water Conservancy and Hydropower Research Institute. From 1987 to 1995, he was a Research Assistant at the School of Mechanical Engineering, Nanjing University of Science and Technology. From 1995 to 2001, he was an Associate Professor at the Nanjing University of Science and Technology. Since 2001, he has been a Professor with the Nanjing University of Science and Technology. His research interests include advanced manufacturing system and industrial engineering.

WANG YULIN received the Ph.D. degree in mechanical engineering science from the Nanjing University of Science and Technology. He is currently a Professor with the Nanjing University of Science and Technology.

His research interests are in the areas of green manufacturing.