IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 October 2023, accepted 7 November 2023, date of publication 13 November 2023,
date of current version 16 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3332310

RESEARCH ARTICLE

Two Stage Path Planning Method for Co-Worked Double Industrial Robots

LI-XIANG ZHANG[1,3], XIN-JIA MENG[1,4], ZHI-JIE DING[1], AND TIAN-SHU WANG[2]
[1]School of Mechanical and Equipment Engineering, Hebei University of Engineering, Handan 056038, China
[2]School of Chinese Materia Medica, Beijing University of Chinese Medicine, Beijing 100029, China
[3]Key Laboratory of Intelligent Industrial Equipment Technology of Hebei Province, Handan 056038, China
[4]Collaborative Innovation Center for Modern Equipment Manufacturing of Jinan New Area (Hebei), Handan 056038, China

Corresponding author: Xin-Jia Meng (mxjdoctor@126.com)

This work was supported in part by the Hebei Natural Science Foundation under Grant E2021402052, and in part by the Handan Science and Technology Research Program under Grant 21422021299.

ABSTRACT It is a significant capability for multi-industrial robots to plan an optimal collision-free path for both end-effectors and robotic arms. However, the path planning methods for co-worked multi-industrial robots, especially for closely co-worked industrial robots is still very limited. In this paper, to tackle the planning problem that has a specified distance constraint of end-effector in the context of complex collision avoidance, a two stage path planning method is proposed for the co-worked double industrial robot. In this two stage path planning method, the dual path planning with the distance constraint and the joint space planning of double robots are integrated sequentially. For the first stage, an algorithm named random sampling particle swarm optimization (RSPSO) is developed to plan the path for each end-effector, which can plan an optimal collision-free path with the specified distance constraint. For the second stage, the joint space planning that combines the inverse kinematics, D-H method and collision detection is performed to find the angular displacements with collision avoidance for dual robotic arms. Two simulation examples and an experiment are used to verify the proposed method.

INDEX TERMS Industrial robots, dual path planning, distance constraint, collision avoidance, joint space planning.

I. INTRODUCTION

The work scenes of industrial robot are continuously expanded since industrial robot has been widely used in manufacturing industry. Some typical applications include painting, welding, packaging and so on. A clear trend towards industrial robots is that its capabilities are gradually improving to accommodate the increasingly raised application needs [1]. However, limited by its own capability, a single robot often seems powerless in the context of complex task. In contrast, multi-robot systems that consist of several single robots, either heterogeneous or homogeneous, are competent to accomplish complex tasks cooperatively [2]. It is therefore attracted more and more attention for multi-robot systems research.

The associate editor coordinating the review of this manuscript and approving it for publication was Wai-Keung Fung.

As one of the core work to multi-robot system, path planning aims to find an optimal collision-free path for both end-effectors and robotic arms. There are two types of planning task, workspace planning of end-effectors and joint space planning of robotic arms, to be conducted in its path planning. Besides, in either workspace planning or joint space planning, collision avoidance including robot-obstacle and inter-robot should be guaranteed primarily. For the workspace planning of end-effectors, a safety path from starting point to target point needs to be found for the co-worked robots. Since this type of planning task only involves end-effectors [3], the end-effectors can be considered as several cooperative mobile robots in some sense. Consequently, the path planning methods of multi mobile robots can be used for the path planning problem of end-effectors. For the path planning methods of multi mobile robots, it is generally divided into two categories, roadmap-based method and tree-based

method [4]. Some already developed roadmap-based methods include multi-agent path finding method [5], M* algorithm [6], sPRM [7], and so on. Additionally, some roadmap-based methods of single robot, such as A* algorithm, PRM, can also be directly used into multi robot simple planning problems [8]. Another tree-based method that finds the optimal path through constructing a search tree has also been well developed, and the famous tree-based methods for multi robot are MA-RRT* [9], dRRT [10], etc. Most of roadmap-based method and tree-based method are established on the basis of sampling technique, which have been practically verified as effective methods.

Once the workspace planning of end-effectors is completed, the joint space planning of robotic arms starts to find the angles that can ensure the exact position and pose for robot [11]. The direct joint space planning method is to perform the inverse kinematics solving for the specified path points, and then, the angular displacements of joints can be obtained. Generally, before inverse kinematics solving, the continuous path will be discretized into many waypoint. Since this direct method requires solving inverse kinematics function repeatedly, it usually has a high computation cost [12]. To alleviate the computational burden, the interpolation technique is naturally applied into joint space planning. The mature interpolation methods that have been widely used in robot joint space planning are cubic polynomial interpolation, quintic polynomial interpolation, B spline interpolation, cubic Hermite interpolation [13]. However, the interpolation methods are generally difficult to achieve the exactly tracking for the specified path of end-effectors [14]. Moreover, some recent attention also has been drawn to optimal trajectory and collision avoidance. Liu et al. [15] proposed a time-energy optimal trajectory planning method for collaborative welding robot. Glorieux et al. [16] developed a non-linear programming optimisation model that can optimise both the robot trajectories and the multi-robot coordination. Shu et al. [17] proposed a trajectory planning approach based on rapidly-exploring random tree-star for robotic assembly of lightweight structures. Chen and Song [18] presented a real-time motion planning and control design of a robotic arm for human-robot collaborative safety. Liu et al. [19] proposed a trajectory optimization technique by gradient-based optimization method, which can effectively optimize the robot trajectory under various task constraints. Tang et al. [20] presented a coevolution-based particle swarm optimization method to cope with the multi-robot path planning issue. Larsen and Kim [21] proposed a path planning strategy based on evolutionary algorithms for cooperating industrial robots.

The already developed path planning methods are effective and practical for supporting the operation of robots. However, the path planning for co-worked multi-industrial robots, especially for closely co-worked industrial robots, is still worth and necessary to be developed. Zhou et al. [22] clearly stated the reachable space of a single robot is not enough to cover the entire workbench sometimes, which may need two or more robots to finish the welding process, and the cooperative path planning problem for the welding robot also has potential for future research. The main challenge of the path planning for co-worked multi-industrial robots comes from complex collision avoidance and constraints of collaboration. The collision avoidance of co-worked industrial robots includes not only obstacle avoidance, but also avoidance of inter-robot collision. On the other hand, during the operation process, the closely co-worked industrial robots usually have specified constraints of collaboration, such as keeping unvarying posture and keeping constant distance between effectors. To tackle the problems of collision avoidance and specified constraints, a two stage path planning method is proposed for co-worked double industrial robots in this article. The involved stages are the dual path planning with the distance constraint and the joint space planning of double robots. In the first stage, an algorithm named random sampling particle swarm optimization (RSPSO) is developed to plan the path of each end-effector. RSPSO employs PSO to generate the feasible path of one end-effector, and then random sampling will generate samples that are located on a spherical surface, which used to generate the feasible path for another end-effector. In the second stage, the joint space planning is performed to find the angular displacements corresponding to the path of end-effectors, and the D-H method, inverse kinematics and collision detection are combined to achieve the collision avoidance.

The rest of the article is organized as follows. Section II analyses the problem. Section III gives the process of the proposed method. In Section IV, two simulation examples and an experiment are used to demonstrate the validity of the proposed method. Finally, Section V concludes this work and makes suggestions for future work.

II. PROBLEM STATEMENT

A. OVERALL PROBLEM ANALYSIS

There are lots of scenarios required the use of dual co-work industrial robots in practical engineering, such as the installation of reducer shaft that needs sufficient stability during installing operation, the heavy parts handling that cannot be done by a single robot, etc. Taking the installation of reducer shaft as an example, the path planning for co-worked double industrial robots is illustrated by Fig. 1.

In Fig. 1, the shaft is moved from starting position to target position by two robots R_1 and R_2. In this process, the end-effector of the robot R_1 moves from starting point S_1 to target point T_1 along the path L_1, while the end-effector of the robot R_2 moves from starting point S_2 to target point T_2 along the path L_2. During the process of robot moving, the distance of the path L_1 to the path L_2 is kept the constant d. In addition, the shaft and the two robots do not collide with the obstacles. Furthermore, when the end-effectors of two robots travel along their respective paths, there should be no collision between the robotic arms of two robots or between the robotic arm and the obstacle. The path planning of this

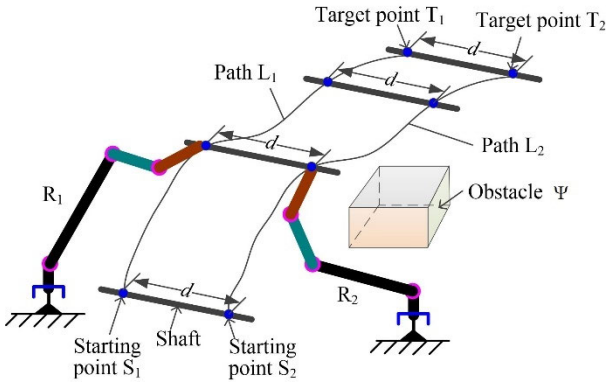

FIGURE 1. Diagrammatic sketch of path planning for co-worked double industrial robots.

problem can be defined by

$$\begin{aligned} &\text{find } \mathbf{L} = \{\mathrm{L}_1, \mathrm{L}_2\} \\ &\min f = f_1 + f_2 \\ &s.t.\ \ |P_1(i), P_2(i)| = d, P_1(i) \in \mathrm{L}_1, P_2(i) \in \mathrm{L}_2 \\ &\quad \mathrm{L}_1 \cap \Psi = \varnothing, \mathrm{L}_2 \cap \Psi = \varnothing, \mathrm{M} \cap \Psi = \varnothing \\ &\quad R_1 \cap R_2 = \varnothing, R_1 \cap \Psi = \varnothing, R_2 \cap \Psi = \varnothing \end{aligned} \tag{1}$$

where, f_1 and f_2 are the distance of the path L_1 and L_2, respectively. f represents the total distance. $P_1(i)$ and $P_2(i)$ are the ith path point of path L_1 and L_2, respectively. d represents the fixed distance, and Ψ is the set of obstacles. M is the set of shaft. R_1 and R_2 are the set of robotic arms of two robots, respectively. The symbol $\varnothing$ means 'empty set'. the symbol $\cap$ represents intersection. The symbol $\in$ means 'belong to'.

Actually, the path planning problem in equation (1) can be divided into two parts. One is the path planning of the end-effectors of two robots, and the other is the joint space planning of robotic arms of two robots. Consequently, the problem of equation (1) can be also divided into two problems, which is shown in equation (2) and equation (3).

$$\begin{aligned} &\text{find } \mathbf{L} = \{\mathrm{L}_1, \mathrm{L}_2\} \\ &\min f = f_1 + f_2 \\ &s.t.\ \ |P_1(i), P_2(i)| = d, P_1(i) \in \mathrm{L}_1, P_2(i) \in \mathrm{L}_2 \\ &\quad \mathrm{L}_1 \cap \Psi = \varnothing, \mathrm{L}_2 \cap \Psi = \varnothing, \mathrm{M} \cap \Psi = \varnothing \end{aligned} \tag{2}$$

$$\begin{aligned} &\text{find } \boldsymbol{\theta} = \left\{\boldsymbol{\theta}^1, \boldsymbol{\theta}^2\right\} \\ &s.t.\ P_{R1} \in \mathrm{L}_1, P_{R2} \in \mathrm{L}_2 \\ &\quad R_1 \cap R_2 = \varnothing, R_1 \cap \Psi = \varnothing, R_2 \cap \Psi = \varnothing \end{aligned} \tag{3}$$

where, n is the number of the joints. $\boldsymbol{\theta} = \left\{\boldsymbol{\theta}^1, \boldsymbol{\theta}^2\right\}$ is the joint angle vector of two robots, in which $\boldsymbol{\theta}^1$ and $\boldsymbol{\theta}^2$ are the joint angle vector of the first robot and the second robot, respectively. The problem in equation (2) is used to find two paths for the end-effectors of two robots, while the problem in equation (3) is used to find the joint angles.

On the basis of the above analysis, the main differences of path planning between co-worked double industrial robots and the single industrial robot are analyzed as follows. Firstly, for the path planning of co-worked double industrial robots, the installation path needs to be planned for each industrial robot, that is, it is necessary to plan two collision free installation paths. Furthermore, in the process of the shaft installation, the distance of the end-effectors of double industrial robots is always fixed from the beginning grasping the shaft until installing the shaft to the specified position, and consequently, the path planning of co-worked double industrial robots is a dual path planning problem under fixed distance constraint. Secondly, the joint space planning is different. For the joint space planning of co-worked double industrial robots, not only the problem of obstacle avoidance between each robot and obstacles needs to be taken into account, but also the problem of obstacle avoidance between two robots needs to be taken into account.

In this paper, a two stage path planning method is proposed to solve this problem. The first stage is the path planning of end-effectors for two robots, while the second stage is the joint space planning for two robots.

B. FIRST STAGE PROBLEM ANALYSIS

The first stage is used to solve the problem in equation (2), which is a dual path planning problem under distance constraint. The path planning problem in equation (2) can be decomposed into the following two nested optimization problem.

$$\begin{aligned} &\min f_1 \\ &s.t.\ P_1(i) \notin \Psi \end{aligned} \tag{4}$$

$$\begin{aligned} &\min f_2 \\ &s.t.\ \ |P_1(i), P_2(i)| = d \\ &\quad P_2(i) \notin \Psi, \mathrm{M} \notin \Psi \end{aligned} \tag{5}$$

The optimization problem (4) is actually an obstacle avoidance path planning problem of single industrial robot. The method such as A* algorithm, RRT, PSO, etc., can be used to solve the path planning problem of equation (4). PSO is simple and easy to implement, which has been widely used in the path planning of the robot [23], [24]. The optimization problem (5) is an obstacle avoidance path planning problem under the constraint of the known path. Fig. 2 shows the schematic diagram of path planning under distance constraint.

Owing to the distance constraints, the path points of the unknown path corresponding to that of the known path can be only taken from a spherical surface. This spherical surface takes the path point of the known path as the centre and takes the fix distance as the radius. As shown in Fig. 2, $P_1(i)$ and $P_1(i-1)$ are two adjacent path points of the known path. $P_2(i-1)$ that corresponds to $P_1(i-1)$, is an obtained path point of the unknown path. A, B, C and D are four points located on the spherical surface. For the next path point $P_2(i)$, these four points all satisfy the requirement of distance constraint. However, A is inside the obstacles, while the next connect line of B and $P_1(i)$ passes through the obstacles. That means both A and B do not meet the collision-free

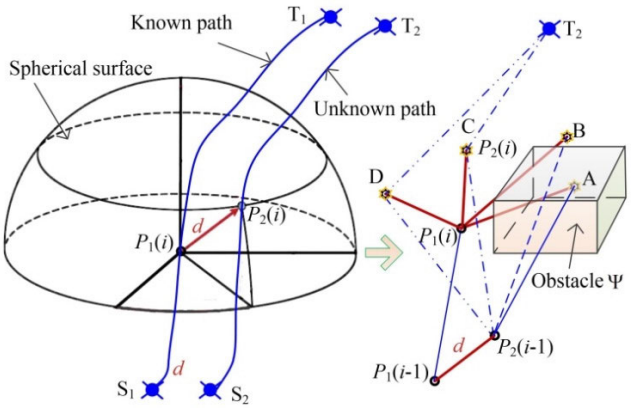

FIGURE 2. Schematic diagram of path planning under distance constraint.

constraint, and thus A and B cannot be taken as the path point $P_2(i)$. For C and D, neither of these points is in the obstacles, and the connect line from C or D to $P_1(i)$ does not pass through the obstacles. Thus, C and D are both the feasible point which can be taken as $P_2(i)$. However, the path length via $P_2(i-1)$, D and target point T_2 is larger than that via $P_2(i-1)$, C and target point T2, that is, $|P_2(i-1), \mathrm{D}| + |\mathrm{D},\mathrm{T}_2| > |P_2(i-1), \mathrm{C}| + |\mathrm{C},\mathrm{T}_2|$. As a result, C is better than D as the path point $P_2(i)$. On the spherical surface, once there is a feasible point, there will be a path point $P_2(i)$ corresponding to $P_1(i)$, that is, it always be found a feasible path under the distance constraint of the known path. It should be pointed out that, for a very special case (all the points on the spherical surface are the same as the condition of point A or B, that is, there is no feasible point for $P_2(i)$), there is no feasible path satisfying the distance constraint of the known path.

Through the above analysis, PSO and random sampling (RS) are used to solve the two nested optimization problem. On the basis of this, a dual path planning method based on RSPSO is proposed for the path planning of co-worked double industrial robots.

C. SENCOND STAGE PROBLEM ANALYSIS

The second stage is used to solve the problem in equation (3) which is a collision-free joint space planning problem of robotic arms. In this stage, collision avoidance including robot-obstacle and inter-robot should be guaranteed. The inverse kinematics solution is used to the problems of angular displacements, and the infeasible solutions of angular displacements are removed by collision detection.

Through the first stage, the position of end-effector of co-worked double robots is determined. The positions of end-effectors are the corresponding path points of two paths obtained by first stage. The posture needs to be determined according to the collaborative task. Then, the inverse kinematics solution is used to solve the joint angles. The position of the robotic arms will be calculated by the forward kinematics solution according to the joint angles. Then, the collision detection can be performed to determine whether the joint angles are the feasible solutions. The dual paths planned by the first stage are feasible only when the feasible joint angles are existed.

The collision detection criterion is that the distance between the robotic arms and between the robotic arm and the obstacle is larger than the allowable minimum distance. The specific collision detection method can be found in Step 5 of part B of Section III.

III. TWO STAGE PATH PLANNING METHOD

In the proposed two stage path planning method, the first stage is dual path planning with the distance constraint, in which RS is nested into the particle swarm optimization (PSO) algorithm to find the optimal dual paths of end-effectors. The second stage is the joint space planning, which takes the D-H method, inverse kinematics and collision detection together to plan a collision free path for the co-worked robotic arms corresponding to the dual path. For this proposed method, the key contribution is that a RSPSO algorithm is proposed for the first time to complete the collision-free dual path planning with the distance constraint for the end-effectors of co-worked double robots. The previous studies on the path planning for the co-worked robots pay more attention to the situation of no distance constraint between robot end-effectors (that is, the situation of that end-effector of each robot performs its own working task.) [20], [21]. For the situation of the path planning under the distance constraint, the planning method based on Coordinated Couple Motion (CCM) [25] is a common method to be used to get the dual path for co-worked robots in. The differences of the proposed RSPSO algorithm from the common dual path planning method is as follows: after obtaining the first path, the proposed RSPSO algorithm adopts RS to obtain the second path, while the CCM planning method adopts the method of moving a fixed distance based on the first path to obtain the second path.

The flowchart of the proposed two stage path planning method is shown in Fig. 3. The proposed two stage path planning method mainly contains initial setup, planning of the first stage and planning of the second stage. For the initial setup, it mainly conducts a preliminary analysis for the working environment to avoid the infeasible work areas. In this step, the path can be planned into multiple parts. Furthermore, the key parameters including the starting point, target point, and allowable spatial scope of each part need to be determined. This situation can refer to the path planning of installation of reducer shaft in Section IV. Then, the planning of the first stage and the planning of the second stage are performed. When the collision-free dual path of end-effectors and the collision-free joint angles of robotic arms are both existed, the path planning will end. Otherwise, the initial setup needs to be updated, and the new path planning will be executed. The details of the first stage and the second stage are explained in the following sections.

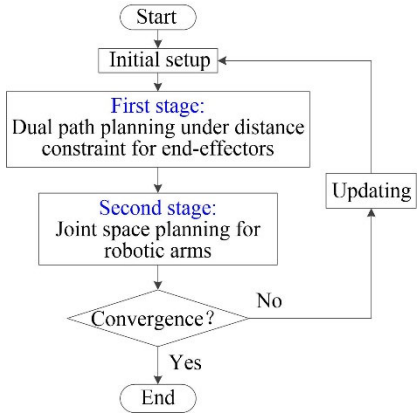

FIGURE 3. Flowchart of the proposed two stage path planning method.

A. DUAL PATH PLANNING WITH THE DISTANCE CONSTRAINT

The main idea of RSPSO is that the PSO is used to find a path for one end-effector, and then based on this generated path, the samples that produced on a spherical surface are used to find the optimal path for another end-effector. The reason of sampling on the spherical surface is that the radius of sphere is specified as the distance of the co-worked end-effectors, which can guarantee the distance constraint in actual operation process. The flowchart of dual path planning based on RSPSO is shown by Fig. 4.

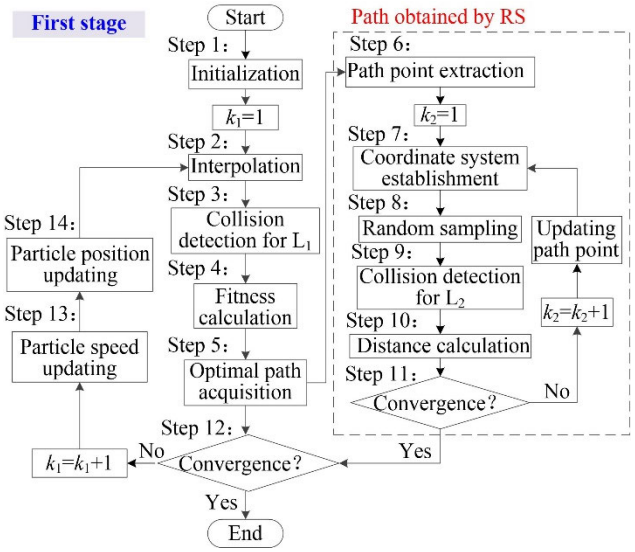

FIGURE 4. Flowchart of the RSPSO algorithm.

The dual path planning based on RSPSO is consists of two parts. One part is the planning of path L_1 by PSO. Another part is the planning of the path L_2 under the distance constraint. PSO algorithm is the main body of the proposed method. When PSO algorithm is used to plan the path L_1, one particle represents one path. RS is integrated into the PSO algorithm, which is from Step 6 to Step 11 shown by Fig. 4. The specific steps of the proposed method are described as below.

Step 1: Initialization

Firstly, the relevant parameters about the path planning, which mainly contain the starting point and the target point of two robots, as well as the workspace, need to be established according to the practical operation conditions. These relevant parameters can be expressed by:

$$\begin{cases} \mathrm{R_1}: \mathrm{S_1} = \left(x_{S1}^0, y_{S1}^0, z_{S1}^0\right), \mathrm{T_1} = \left(x_{T1}^0, y_{T1}^0, z_{T1}^0\right) \\ \mathrm{R_2}: \mathrm{S_2} = \left(x_{S2}^0, y_{S2}^0, z_{S2}^0\right), \mathrm{T_2} = \left(x_{T2}^0, y_{T2}^0, z_{T2}^0\right) \\ \text{Workspace}: \mathbf{W} = [\mathbf{W}_{\min}; \mathbf{W}_{\max}] \\ \text{where}, \mathbf{W}_{\min} = (x_{\min}, y_{\min}, z_{\min}) \\ \mathbf{W}_{\max} = (x_{\max}, y_{\max}, z_{\max}) \end{cases} \tag{6}$$

In equation (6), $[x, y, z]$ denotes the coordinates in the world coordinate system.

Secondly, the initial values of PSO solving need to be set up. The initial step is set as $k_1 = 0$. The initial speed of the particle updating is set as $\mathbf{V}^0 = [V_x^0, V_y^0, V_z^0]$, which is the preset initial value of particle updating speed corresponding to equation (18), where V_x^0, V_y^0 and V_z^0 are the initial speed in X, Y, Z direction, respectively. Generate the initial position control point of particles between starting point $\mathrm{S_1}$ and target point $\mathrm{T_1}$. The initial position of the control point is defined by:

$$\mathbf{W}^0 = \left[\mathbf{W}_1^0, \mathbf{W}_2^0, \ldots, \mathbf{W}_m^0\right] \tag{7}$$

where $\mathbf{W}_i^0$ $(i = 1, 2, \ldots, m)$ is the initial position control point of the ith particle, and m is the particle swarm size. Besides, $\mathbf{W}_i^0 = \left[C_1^0, C_2^0, \ldots, C_n^0\right]^T$, where C_j^0 $(j = 1, 2, \ldots, n)$ is the coordinates of the jth initial position control point, n is the number of position control points of each particle, and the superscript T represents transposition.

Step 2: Interpolation

The particles of the current iteration are obtained by B-spline interpolation. The B-spline interpolation equation is as follows:

$$B(u) = \sum_{i=0}^{n+1} C_i^{k_1} N_{i,j}(u) \tag{8}$$

where, $C_i^{k_1}$ is the ith position control point of the k_1th iteration, $C_0^{k_1} = \mathrm{S_1}$ and $C_{n+1}^{k_1} = \mathrm{T_1}$. u is the node vector, and $N_{i,j}(u)$ is the jth order basis function.

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1, & u_i \le u < u_{i+1} \\ 0, & \text{else} \end{cases} \\ N_{i,p}(u) = \dfrac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \dfrac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \end{cases} \tag{9}$$

Step 3: Collision detection for L_1

This step is mainly used to judge whether the path L_1 collides with an obstacle. The obstacle model is constructed according to actual environment. The feasible particles by

step 2 are obtained by collision detection. The basic detection criteria are

$$\begin{cases} P_{1,i} \in \Psi, & i = 1, 2, \ldots, m, \text{Yes} \\ P_{1,i} \notin \Psi, & i = 1, 2, \ldots, m, \text{No} \end{cases} \tag{10}$$

When one or some path point exited on the path belong to the set of the obstacle, the collision will occur, and this particle is an infeasible path. Otherwise, the particle is a feasible path.

Step 4: Fitness calculation

The fitness function is an important index to evaluate path quality. The fitness function is defined by

$$f_{1,j} = \sum_{i=0}^{N} \left| P_{1,i}, P_{1,i+1} \right| \tag{11}$$

where, $f_{1,j}$ is the path length of the jth particle, the subscript 1 represent the path L_1, N is the number of path point, P_0 is the starting point of robot R_1 (that is, $P_{1,0} = \mathrm{S}_1$), and P_{N+1} is the target point of robot R_1 (that is, $P_{1,N+1} = \mathrm{T}_1$).

Step 5: Optimal path acquisition

The fitness values of all particles are compared. The particle corresponding to the minimum fitness value is the optimal path in current iteration. The optimal particle is recorded as L_{1,k_1}, while the minimum fitness value is recorded as f_{1,k_1}. The minimum fitness value is passed to Step 12. The path point of path L_1 is passed to Step 6.

Step 6: Path point extraction

The initial values are set as $k_2 = 1$. Then, the k_2th path point of path L_1 is extracted, which is recorded as P_{1,k_2}. At the same time, the initial distance of path L_2 is set as $f^0_{2,k_1} = 0$.

Step 7: Coordinate system establishment

Due to the distance constraints, the path point P_{2,k_2} of path L_2 corresponding to the point P_{1,k_2} of path L_1 can be only taken from the sphere with point P_{1,k_2} as the centre and the distance d as the radius. Consequently, a local coordinate system needs to be established to perform the random sampling in order to obtain the path point of the path L_2 corresponding to the point P_{1,k_2}. The schematic diagram of path solution under distance constraint is shown in Fig. 5.

In Fig. 5, the XYZ coordinate system is the global coordinate system, while X′ Y′ Z′ coordinate system is the local coordinate system. The direction of coordinate axis is the same as the global coordinate system XYZ. The P_{1,k_2} is the k_2th point of the path L_1, which is also the original point of X′ Y′ Z′ coordinate system. The coordinate of P_{2,k_2} is given by

$$\begin{cases} x_{2,k_2} = x'_{2,k_2} + x_{1,k_2} \\ y_{2,k_2} = y'_{2,k_2} + y_{1,k_2} \\ z_{2,k_2} = z'_{2,k_2} + z_{1,k_2} \end{cases} \tag{12}$$

where, $(x_{1,k_2}, y_{1,k_2}, z_{1,k_2})$ is the global coordinate of the point P_{1,k_2}, $(x_{2,k_2}, y_{2,k_2}, z_{2,k_2})$ is the global coordinate of the point P_{2,k_2}, $(x'_{2,k_2}, y'_{2,k_2}, z'_{2,k_2})$ is the local coordinate of the point P_{2,k_2}, and the possible local coordinate can be calculate by

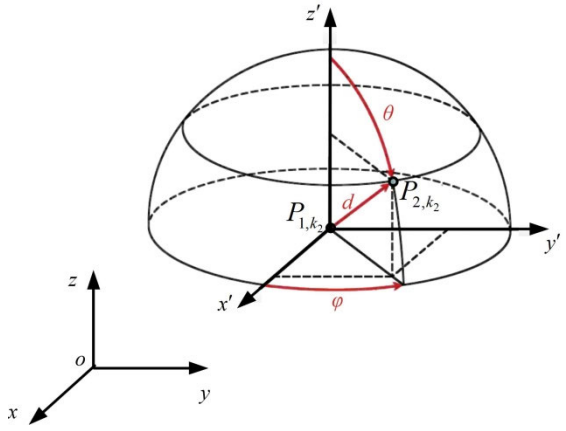

FIGURE 5. Schematic diagram of path solution under distance constraint.

$$\begin{cases} x'_{2,k_2} = d \sin\theta \cos\varphi \\ y'_{2,k_2} = d \sin\theta \sin\varphi \\ z'_{2,k_2} = d \cos\theta \end{cases} \tag{13}$$

where, d is the constraint distance between path L_1 and path L_2. θ and φ are the sampling parameters, which are the angle with the local coordinate axis (see Fig. 5). The maximum range of these two parameters are $\theta \in [0, \pi]$ and $\varphi \in [0, 2\pi]$, and the practical range can be defined according to the practical engineering in order to reduce the time cost of path planning.

Step 8: Random sampling

The parameters θ and φ are obtained by random sampling within the value range. Here, Latin hypercube sampling (LHS) is used to solve the samples. Then, the samples of the point P_{2,k_2} in the global coordinate system are calculated according to equation (12) and equation (13).

Step 9: Collision detection for L_2

This step is mainly used to judge whether the path points obtained by RS is feasible. Therefore, the collision detection needs to be performed for the path points by step 8. In addition, considering the size of the object moved by robots, the collision detection of the object also need to be performed. Only when both the path points and the object do not collide with the obstacles, the path points by RS can be considered to be feasible for path L_2. That is, there are two collision problems to be considered in this step. One is the collision problem of the path point P_{2,k_2}, another is the collision problem of the object moved by robots.

For the path points by RS, the collision detection can be performed according to Step 3. For the object, the basic detection principle of this problem is that the straight line from the path point P_{2,k_2} to P_{1,k_2} is outside the obstacle space. Furthermore, if the grasping position of the robot is not at the endpoint of the object, line segment from the endpoint of the object to the path point also need to be placed outside the obstacle space. The specific detection method can refer to the Part A of Section IV.

When the feasible sample points do not exist, that is, for the very special case mentioned in Section II, the feasible path L_{2,k_1} corresponding to the path L_{1,k_1} do not exist. In this condition, it can go to Step 1 to restart the process of the path planning by generating the new initial position control point.

Step 10: Distance calculation

The distances of the feasible samples to the last point P_{2,k_2-1} and the target point T_2 can be calculated by

$$\begin{cases} f^1_{2,k_2} = \left|P_{2,k_2-1}, \text{Samples}\right| \\ f^2_{2,k_2} = |\text{T}_2, \text{Samples}| \\ f'_{2,k_2} = f^1_{2,k_2} + f^2_{2,k_2} \end{cases} \tag{14}$$

The sample point corresponding to the minimum f'_{2,k_2} is taken as the point P_{2,k_2}. The distance of path L_2 is

$$\begin{cases} f_{2,k_1} = f^{k_2-1}_{2,k_1} + f^1_{2,k_2}, k_2 < N_{point} \\ f_{2,k_1} = f^{k_2-1}_{2,k_1} + \min f'_{2,k_2}, k_2 = N_{point} \end{cases} \tag{15}$$

where, N_{point} is the point number of the path L_1.

Step 11: Convergence checking

The convergence criterion of the path solving under distance constraint is set as

$$k_2 > N_{point} \tag{16}$$

When the convergence criterion is satisfied, the path solving under distance constraint will stop. Otherwise, let $k_2 = k_2 + 1$, and go to Step 7.

Step 12: Global convergence checking

The global convergence criterion is set as follows:

$$k_1 > N_{iter} \ or \ \left|f_{k_1} - f_{k_1-1}\right| < \varepsilon, N_\varepsilon > [N_\varepsilon] \tag{17}$$

where, N_{iter} is the maximum iteration time. f_{k_1} is the sum of paths L_1 and L_2, that is, $f_{k_1} = f_{1,k_1} + f_{2,k_1}$. ε is the convergence precision. N_ε is the minimum iteration times of satisfied convergence precision, and $[N_\varepsilon]$ is a predetermined minimum iteration times of satisfied convergence precision.

When the convergence criterion is satisfied, the whole solving process will stop. Otherwise, let $k_1 = k_1 + 1$, and go to Step 13.

Step 13: Particle speed updating

The updating principle of particle speed is as follows:

$$\begin{cases} \mathbf{V}_i^{k_1} = \omega \mathbf{V}_i^{k_1-1} + c_1 r_1 (\mathbf{W}_{i,best}^{k_1-1} - \mathbf{W}_i^{k_1-1}) \\ +c_2 r_2 (\mathbf{W}_{g,best}^{k_1-1} - \mathbf{W}_i^{k_1-1}) \\ \text{if } \mathbf{V}_i^{k_1} > \mathbf{V}_{\max}, \text{ then } \mathbf{V}_i^{k_1} = \mathbf{V}_{\max} \\ \text{elseif } \mathbf{V}_i^{k_1} < \mathbf{V}_{\min}, \text{ then } \mathbf{V}_i^{k_1} = \mathbf{V}_{\min} \\ \text{else } \mathbf{V}_i^{k_1} = \mathbf{V}_i^{k_1} \end{cases} \tag{18}$$

where $\mathbf{W}_{i,best}^{k_1-1}$ is the optimal position of ith particle, $\mathbf{X}_{g,best}^{k_1-1}$ is the global optimal position of all the particles, c_1 and c_2 are the learning factor of particle, ω is the inertia weight, r_1 and r_2 are the random number within 0 and 1. $\mathbf{V}_{\min}$ and $\mathbf{V}_{\max}$ are the permissive minimum and maximum of particle speed updating. The updating principle of individual and global optimal position is shown in equation (19) and equation (20), respectively.

$$\mathbf{W}_{i,best}^{k_1} = \begin{cases} \mathbf{W}_{i,best}^{k_1}, f\left(\mathbf{W}_{i,best}^{k_1}\right) \le f\left(\mathbf{W}_{i,best}^{k_1-1}\right) \\ \mathbf{W}_{i,best}^{k_1-1}, f\left(\mathbf{W}_{i,best}^{k_1}\right) > f\left(\mathbf{W}_{i,best}^{k_1-1}\right) \end{cases} \tag{19}$$

$$\mathbf{W}_{g,best}^{k_1} = \begin{cases} \mathbf{W}_{g,best}^{k_1}, f\left(\mathbf{W}_{g,best}^{k_1}\right) \le f\left(\mathbf{W}_{g,best}^{k_1-1}\right) \\ \mathbf{W}_{g,best}^{k_1-1}, f\left(\mathbf{W}_{g,best}^{k_1}\right) > f\left(\mathbf{W}_{g,best}^{k_1-1}\right) \end{cases} \tag{20}$$

where, f is the fitness function by Step 4.

Step 14: Particle position updating

The updating principle of particle position is as follows

$$\begin{cases} \mathbf{W}_i^{k_1} = \mathbf{W}_i^{k_1-1} + \mathbf{V}_i^{k_1} \\ \text{if } \mathbf{W}_i^{k_1} > \mathbf{W}_{\max}, \text{ then } \mathbf{W}_i^{k_1} = \mathbf{W}_{\max} \\ \text{else if } \mathbf{W}_i^{k_1} < \mathbf{W}_{\min}, \text{ then } \mathbf{W}_i^{k_1} = \mathbf{W}_{\min} \\ \text{else } \mathbf{W}_i^{k_1} = \mathbf{W}_i^{k_1} \end{cases} \tag{21}$$

Then go to Step 2, and continue to solve iteratively until the convergence criterion of Step 12 is satisfied.

B. JOINT SPACE PLANNING OF CO-WORKED DOUBLE INDUSTRIAL ROBOTS

The joint space planning is performed after two paths are obtained. The flowchart of joint space planning of co-worked double industrial robots is shown in Fig. 6.

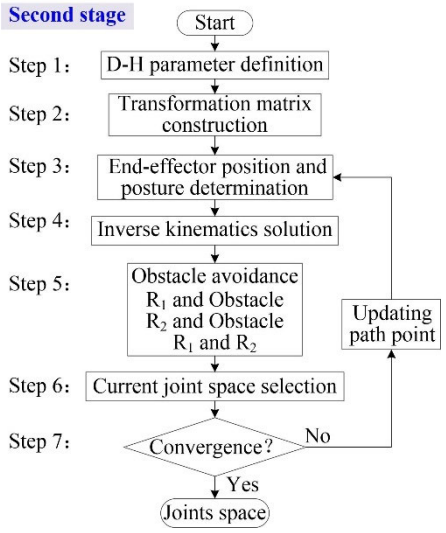

FIGURE 6. Flowchart of joint space planning of co-worked double industrial robots.

Step 1: D-H parameters definition

According to the joint coordinate system, D-H parameters of the robot R_1 and R_2 can be given by $\left[\theta_j^i, \alpha_j^i, a_j^i, d_j^i\right]$, where the superscript i ($i = 1, 2$) represents the robot, θ_j^i is the angle between x_{j-1}^i and x_j^i axes measured about z_j^i axis, α_j^i is the angle between z_j^i and z_{j+1}^i axes measured about x_j^i axis, a_j^i is the distance from z_j^i to z_{j+1}^i axes measured along x_j^i axis and

d_j^i is the distance from x_{j-1}^i to x_j^i axes measured along z_j^i axis. α_j^i, a_j^i, d_j^i are the constant value, while θ_j^i is the variable.

Step 2: Transformation matrix construction

Using D-H parameters, the transformation matrix of the two adjacent joints for the ith robot can be expressed as

$$
{}^{j-1}{}_jT_i = \begin{bmatrix} \cos\theta_j^i & -\sin\theta_j^i & 0 & a_{j-1} \\ \sin\theta_j^i\cos\alpha_{j-1}^i & \cos\theta_j^i\cos\alpha_{j-1}^i & -\sin\alpha_{j-1}^i & -d_j^i\sin\alpha_{j-1}^i \\ \sin\theta_j^i\sin\alpha_{j-1}^i & \cos\theta_j^i\sin\alpha_{j-1}^i & \cos\alpha_{j-1}^i & d_j^i\cos\alpha_{j-1}^i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{22}
$$

The transformation matrix of the ith robot from the base frame to the end-effector is obtained by multiplying all of the joints matrices, which is given by

$$
{}^0_nT_i = {}^0_1T_i{}^1_2T_i \ldots {}^{n-1}_nT_i \tag{23}
$$

where, n is the number of the joints. Step 3: End-effector position and posture determination Extracting the kth path points of path L_1 and L_2, which are recorded as $P_{1,k}$ and $P_{2,k}$. Determinate the position and orientation of the end-effector with respect to the kth path point for two robots, respectively. The initial value of k is 0, that is, the initial path point is the starting point. The posture can be given according to the grasping task of the robot end-effectors. The position and posture of the end-effector of the co-worked multi-industrial robots at the kth path point is recorded as

$$
g_{i,k}(\boldsymbol{\theta}^i) = \begin{bmatrix} n_{ik-x} & o_{ik-x} & a_{ik-x} & p_{ik-x} \\ n_{ik-y} & o_{ik-y} & a_{ik-y} & p_{ik-y} \\ n_{ik-z} & o_{ik-z} & a_{ik-z} & p_{ik-z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{24}
$$

where, the subscript i stands for the ith robot, and the subscript k stands for the kth path point. The matrix $[n_{ik-x}, n_{ik-y}, n_{ik-z};\ o_{ik-x}, o_{ik-y}, o_{ik-z};\ a_{ik-x}, a_{ik-y}, a_{ik-z}]$ stands for the posture of the end-effector of the ith robot at the kth path point. The vector $[p_{ik-x}, p_{ik-y}, p_{ik-z}]$ stands for the coordinate value of the end-effector of the ith robot at the kth path point, which is the position of the ith robot at the kth path point.

Step 4: Inverse kinematics solution

There are mainly two types of inverse kinematics solution techniques, namely analytical and numerical. Normally, when the robot satisfies the Pieper criteria, the analytical method can be used to perform the inverse kinematics solution. In analytical solution technique, the joint angle can be solved by

$$
\begin{aligned} &\left[{}^0_jT_i\right]^{-1}{}^0_nT_i = {}^i_{i+1}T_i \ldots {}^{n-1}_nT_i \\ &where, {}^0_iT_i = {}^0_1T_i \ldots {}^{i-1}_iT_i, \quad i = 1, 2, \ldots, n-1. \end{aligned} \tag{25}
$$

Through equation (25), the angle of the jth joint for the ith robot can be obtained.

Step 5: Collision detection

For the co-worked double industrial robots, the collision detection contains two parts. One is the collision detection between each robot and obstacles, and the other is the collision detection between two robots. The detection principles of these two conditions are shown in Fig. 7 and Fig. 8, respectively.

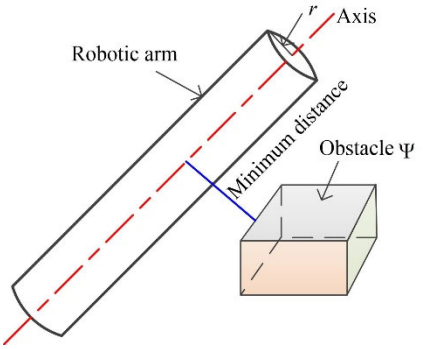

FIGURE 7. Collision detection between robotic arm and obstacle.

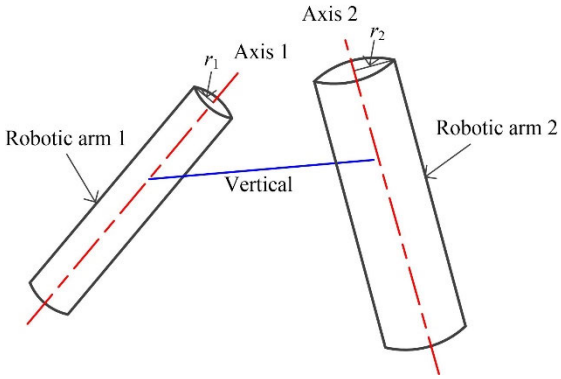

FIGURE 8. Collision detection between robotic arms.

In Fig. 7 and Fig. 8, the robotic arm is enveloped into a cylinder. For the condition of collision detection between robotic arm and obstacle, the detection principle is that the minimum distance from the cylinder axis to the obstacle is larger than the cylinder radius, which can be defined by

$$
\begin{cases} d_{\min} > r, & \text{No collision} \\ d_{\min} \le r, & \text{Collision} \end{cases} \tag{26}
$$

where, $d_{\min}$ is the minimum distance from the axis of the envelope cylinder to the obstacle, and r is the radius of the envelope cylinder. It should be noted that, the shape of the obstacle in Fig. 7 just is a schematic shape, which can be enveloped into different shapes (such as cuboid, sphere and ellipsoid) according to the actual shape of real obstacle. For the obstacles with different enveloped shapes, the solving equation of $d_{\min}$ is also different, which can refer to Ref. [26], [27], and [28].

For the condition of collision detection between robotic arms, the detection principle is that the common vertical line

between two axes is larger than the sum of the radius of two cylinders, which can be defined by

$$\begin{cases} d_V > r_1 + r_2, & \text{No collision} \\ d_V \le r_1 + r_2, & \text{Collision} \end{cases} \tag{27}$$

where, d_V is the distance of the common vertical line between two axes. r_1 and r_2 are the radius of two envelope cylinder, respectively.

The specific location of each joint of the robotic arm corresponding to each inverse solution can be obtained by using the forward kinematics solution. Then the collision detection under the specific location is performed. The collision-free inverse solution will be retained as a feasible solution. Once the collision detections are completed for all the inverse solutions, all the feasible inverse solution will be given.

Step 6: Current joint space selection

From all the feasible inverse solutions obtained by Step 5, the optimal solution is selected as the current joint space with respect to the kth path point. The selection principle is minimum total rotation of each joint, which can be expressed by

$$J = \min \sum_{j=1}^{n} \theta_j^i \tag{28}$$

Step 7: Convergence checking

The convergence criterion is that the joint space with respect to all the path point containing the starting point and the target point. Consequently, the convergence condition is as follows

$$k > N_{\text{point}} + 1 \tag{29}$$

When the convergence criterion is satisfied, the joint space planning will be stop. Otherwise, let $k = k+1$, and update the path point. Then go to Step 3, the joint space planning with respect to the next path point will be performed.

IV. EXAMPLE VALIDATION

A. CASE STUDY OF DUAL PATH PLANNING WITH THE DISTANCE CONSTRAINT

In this example, four ellipsoidal obstacles are designed to validate the effectiveness of RSPSO algorithm. The working task of two robots is to move a cylindrical object. The grasping position is at the ends of the cylindrical object. The grasping distance is $d = 5$, that is, the constraint distance between two paths is 5. The starting point and target point of R_1 and R_2 are S_1(2,2,3), T_1(95,100,102) and S_2(5,2,7), T_2(95,103,106), respectively. The ranges of the random parameters are $\theta \in [0, \pi]$ and $\varphi \in [0, 2\pi]$. The parameters of obstacles are listed in Table 1.

When the proposed RSPSO is used to finish the dual path planning, the collision detection contains two parts which are the collision detection of the path points in two paths and the collision detection of the straight line between two corresponding path points (represents that the collision is not happened between the working object and the obstacles).

TABLE 1. Obstacle parameters.

No.	Central coordinate			semi-axis		
	x_c	y_c	z_c	x_r	y_r	z_r
1	10	15	8	3	3	6
2	30	25	20	4	5	6
3	50	55	65	8	12	12
4	85	85	85	6	4	6

For the collision detection of the path points in two paths, the collision detection principle is to judge whether the path point is in the ellipsoid. The collision-free criteria can be expressed by

$$Q_T = \left(\frac{x - x_c}{x_r}\right)^2 + \left(\frac{y - y_c}{y_r}\right)^2 + \left(\frac{z - z_c}{z_r}\right)^2 > 1 \tag{30}$$

where, (x, y, z) is the global coordinates of path point.

For the collision detection of the straight line between two corresponding path points, the location of the foot point from the central point of the obstacles to the straight line is used to detect the collision. Firstly, the foot point from the central point of the obstacles to the straight line can be calculated by

$$\begin{cases} x_0 = k_0(x_{2,k_2} - x_{1,k_2}) + x_{1,k_2} \\ y_0 = k_0(y_{2,k_2} - y_{1,k_2}) + y_{1,k_2} \\ z_0 = k_0(z_{2,k_2} - z_{1,k_2}) + z_{1,k_2} \end{cases} \tag{31}$$

where,

$$\begin{aligned} k_0 &= -\frac{A_x + A_y + A_z}{A} \\ A_x &= (x_{1,k_2} - x_c)(x_{2,k_2} - x_{1,k_2}) \\ A_y &= (y_{1,k_2} - y_c)(y_{2,k_2} - y_{1,k_2}) \\ A_z &= (z_{1,k_2} - z_c)(z_{2,k_2} - z_{1,k_2}) \\ A &= (x_{2,k_2} - x_{1,k_2})^2 + (y_{2,k_2} - y_{1,k_2})^2 + (z_{2,k_2} - z_{1,k_2})^2 \end{aligned} \tag{32}$$

When the foot point is located between two path points, the collision detection principle is to judge whether the foot point is in the ellipsoid, which is the same as the detection principle of the path points (See equation (30)). Otherwise, as long as the path points satisfy the collision-free criteria, the straight line between two corresponding path points will not collide with the obstacles.

The proposed method involves the PSO algorithm and random sampling. Some system parameters have a significant influence on solving, such as population size, sampling quantity and the number of path points. The sensitivity of these three main parameters is analysed in this example. The sensitivity analysis of one parameter is simulated by MALAB when the other two parameters are fixed, and the results are shown in Table 2, Table 3 and Table 4.

Table 2 gives the results of different population size (which is 10, 30, 50, 70 and 90), and there are three simulations corresponding to each population size. From Table 2, when the population size is 70, the average of the fitness function (which is 356.1834) is the most minimum compared to other

TABLE 2. Sensitivity analysis of the population size.

Population size	Fitness function				Deviation from average		
	First	Second	Third	Average	First	Second	Third
10	357.0484	359.7270	364.9199	360.5651	−3.5167	−0.8381	4.3548
30	356.9505	361.9742	358.8979	359.2742	−2.3237	2.7	−0.3763
50	354.0637	356.3418	363.4679	357.9578	−3.8941	−1.616	5.5101
70	359.9724	352.4611	356.1167	356.1834	3.789	−3.7223	−0.0667
90	355.2916	359.1196	358.7740	357.7284	−2.4368	1.3912	1.0456

four data (which is range from 357.7284 to 360.5651). However, the three deviations from average corresponding to the population size 90 (which are −2.4368, 1.3912 and 1.0456) have the smallest gap between each other. This demonstrates that the simulation results are the most stable when the population size is 90.

TABLE 3. Sensitivity analysis of the sampling quantity.

Sampling quantity	Fitness function		Simulation time	
	Value	Decrease	Value	Increase
200	361.0480	——	501.33 s	——
300	353.2192	7.8288	813.05 s	311.72 s
400	348.9573	4.2619	1882.48 s	1069.43 s
500	347.7279	1.2294	2002.79 s	120.31 s
600	346.9976	0.7303	2082.45 s	79.66 s

Table 3 shows the simulation results when the sampling quantity is 200, 300, 400, 500 and 600, respectively. As can be seen from Table 3, when the sampling quantity changes from 200 to 600, the fitness function is getting smaller gradually, while the simulation time is getting longer gradually. Moreover, the decreasing rate of fitness function is far less than the increasing rate of simulation time. In addition, when the sampling quantity changes from 400 to 500, both the decrement of the fitness function and the increment of the simulation time are very small, which are only 1.2294 and 120.31 s. The decrement and increment reach the peak value when the sampling quantity is 400.

TABLE 4. Sensitivity analysis of number of path points.

Number of path points	Fitness function	Cycle time	Simulation time
50	343.6783	63	548 s
100	356.3418	42	469.67 s
150	370.1987	86	1476.32 s
200	397.2386	36	683.09 s
250	408.6388	41	1343.39s

Table 4 lists the simulation results with the different number of the path point. From Table 4, the fitness function increases with the increase of number of path points. That is because the fitness function is the length of the path, which is obtained from the superposition of the straight distance from point to point. When the path point is much fewer, the path is much closer to the straight line. Consequently, the fewer the path point is, the smaller the fitness function is. In addition, the cycle time and the simulation time with number of path points as 100 is relatively few compared to others.

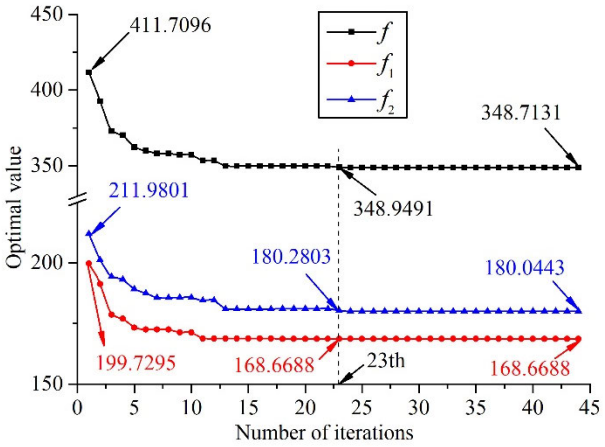

FIGURE 9. Iteration history.

Through the above analysis, 90, 400 and 100 are selecting as these three parameters in this example. The predetermined minimum iteration times of satisfied convergence precision is set as $[N_\varepsilon] = 20$. The iteration history is shown in Fig. 9.

As shown in Fig. 9, the fitness function (f, f_1, f_2) decreased from the initial (411.7096, 199.7295, 211.9801) to (348.9491, 168.6688, 180.2803) in the first 23 cycles. After the 23th cycle, the fitness function is tending towards stability. When the cycle is 44, the convergence precision is satisfied for 20 consecutive times. The length of path L_1 and L_2 is 180.0443 and 168.6688, respectively, and the total length is 348.7131. Fig. 10 gives the path L_1 under B-spline interpolation. The number of position control points from starting point to target point is set 3, which is C_1 (31.1006, 38.6076, 37.6437), C_2 (46.5727, 58.4784, 51.0669) and C_3 (60.196, 71.6803, 67.7921) shown in Fig. 10. The broken blue line is the path L_1 by position control points, which is the path before B-spline interpolation. The solid red line is the path L_1 by B-spline interpolation. The path by B-spline interpolation is much smoother than that by position control points.

Fig. 11 shows the obtained dual paths. In order to better observe whether the path collides with obstacles, three observe direction (D1, D2 and D3) are also given in Fig. 11. As shown in Fig. 11, both two paths do not collide with four obstacles.

The problem of this validation example is also solved by four other methods in order to much better verify the proposed method. Table 5 gives the comparison results of ten runs by different methods.

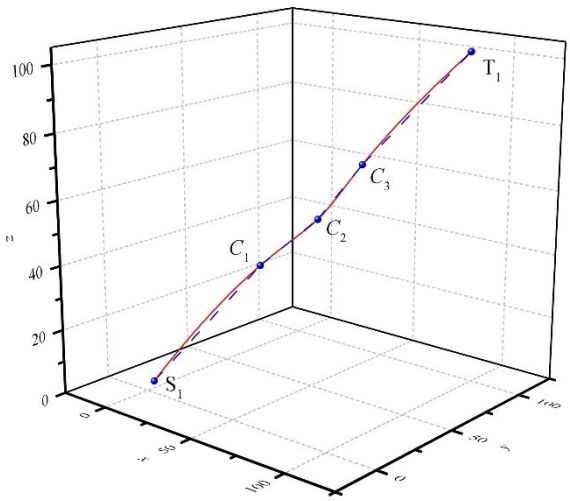

FIGURE 10. Path L_1 under B-spline interpolation.

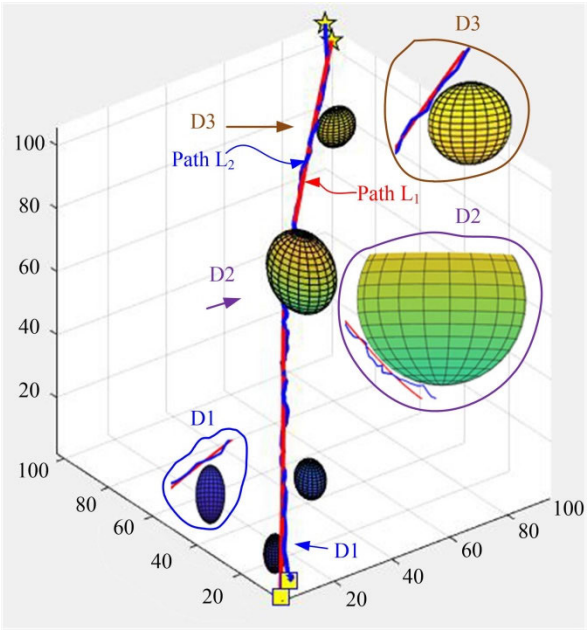

FIGURE 11. Graph of dual path.

TABLE 5. Comparison of different methods.

Method	BPL	AST (s)	SR (%)
A* with CCM	350.0261	870.93	60
RRT with CCM	347.8254	1250.15	70
RRT-RS	349.3356	1529.87	100
GA-RS	349.5782	1287.36	100
RSPSO	348.7131	1097.75	100

In Table 5, the BPL (which is the abbreviation of Best Path Length) is the shortest path within ten runs by different methods. The AST (which is the abbreviation of Average Simulation Time) is the average time of ten runs. The SR (which is the abbreviation of Success Rate) is the probability of obtaining feasible path planning in ten runs. A* with CCM or RRT with CCM is the method in which CCM is adopted to obtain the second path after the first path is planned by A* or RRT algorithm. RRT-RS or GA-RS is the method in which RS is adopted to obtain the second path after the first path is planned by RRT algorithm or GA (genetic algorithm). Firstly, through the comparison of the proposed RSPSO with A* with CCM and RRT with CCM, the BPL obtained by the proposed RSPSO is shorter than that by A* with CCM, while the BPL obtained by RSPSO is longer than that by RRT with CCM. For AST, the proposed RSPSO is bigger than A* with CCM and smaller than RRT with CCM. Secondly, through the comparison of the proposed RSPSO with RRT-RS and GA-RS, the BPL and the AST by RSPSO are best in these three methods. Besides, the ANI by RSPSO is less than that by GA-RS. For RSPSO and GA-RS, each iteration needs to perform 90 times of fitness calculation for path L_1 and perform 100 times of sampling operations for path L_2. The ANL (Average Number of iterations, which is the average times of iteration in ten runs) for RSPSO is 47, while the ANL for GA-RS is 58. Besides, through the comparison of these five methods, the SR by the methods of RRT-RS, GA-RS and RSPSO is 100%, while the SR by the methods of A* with CCM and RRT with CCM is 60% and 70%, respectively. This demonstrates that the SR by the RS-based methods is higher than that by the CCM-based methods. This is due to that a feasible path can be always found to match the first path when RS method is adopted to plan the second path after the first path is obtained (except the very special case, see Section II). However, for the CCM-based method, the CCM is used to get the second path after the first path is obtained. The second path obtained by CCM may collide with the obstacle, and therefore the failure of path planning will be occurred. The above comparisons of five methods indicate that the proposed RSPSO method can effectively realize the dual path planning with the distance constraint.

B. PATH PLANNING OF INSTALLATION OF REDUCER SHAFT

This example, which is unrelated to the first example, is mainly to verify the effectiveness of the proposed two stage path planning method. In this example, the path planning of installation of reducer shaft is adopted to complete the overall simulation verification from the path planning for the end-effectors to the joint space planning for robotic arms.

The gear reducer is a commonly used component in many kinds of mechanical equipment, which generally has a stable performance and a precise transmission ratio. The parts mounted on the shaft, including bearing, gear and shaft sleeve, always needs to keep sufficient stability during installing operation, and therefore, the co-worked double industrial robots can be employed to conduct the installation of reducer shaft. Since the shaft is a rigid body, the distance between the two end-effectors will be fixed once the shaft is caught by the industrial robots. The installation task of reducer shaft is shown in Fig. 12.

TABLE 6. Inverse kinematics solution.

No	Marker	Joint angle (rad)					
		θ_1^1, θ_1^2	θ_2^1, θ_2^2	θ_3^1, θ_3^2	θ_4^1, θ_4^2	θ_5^1, θ_5^2	θ_6^1, θ_6^2
1	(P_{10}, P_{20})	−0.5617, 0.5619	0.8052, 0.8057	0.2177, 0.2167	0, 0	−1.0229, −1.0224	−0.5617, 0.5619
2	(P_{11}, P_{21})	−0.5358, 0.5016	0.8759, 0.9069	0.1965, 0.167	0, 0	−1.0725, −1.0739	−0.5358, 0.5016
3	(P_{12}, P_{22})	−0.5098, 0.4853	0.9359, 0.9533	0.1824, 0.1538	0, 0	−1.1183, −1.1071	−0.5098, 0.4853
4	(P_{13}, P_{23})	−0.4912, 0.4718	0.9841, 0.9992	0.1889, 0.1669	0, 0	−1.1730, −1.1661	−0.4912, 0.4718
5	(P_{14}, P_{24})	−0.4771, 0.4773	1.0187, 1.0193	0.2006, 0.1997	0, 0	−1.2193, −1.2190	−0.4771, 0.4773
6	(P_{15}, P_{25})	−0.3676, 0.3572	1.0995, 1.1145	0.0744, 0.0843	0, 0	−1.1739, −1.1988	−0.3676, 0.3572
7	(P_{16}, P_{26})	−0.3046, 0.3000	1.1336, 1.1500	0.0228, 0.0591	0, 0	−1.1565, −1.2091	−0.3046, 0.3000
8	(P_{17}, P_{27})	−0.1267, 0.1255	1.1918, 1.1970	−0.0624, −0.0569	0, 0	−1.1294, −1.1401	−0.1267, 0.1255
9	(P_{18}, P_{28})	0, 0	1.2034, 1.2040	−0.0790, −0.0799	0, 0	−1.1244, −1.1241	0, 0
10	(P_{19}, P_{29})	0, 0	1.1771, 1.1778	−0.1097, −0.1106	0, 0	−1.0674, −1.0671	0, 0

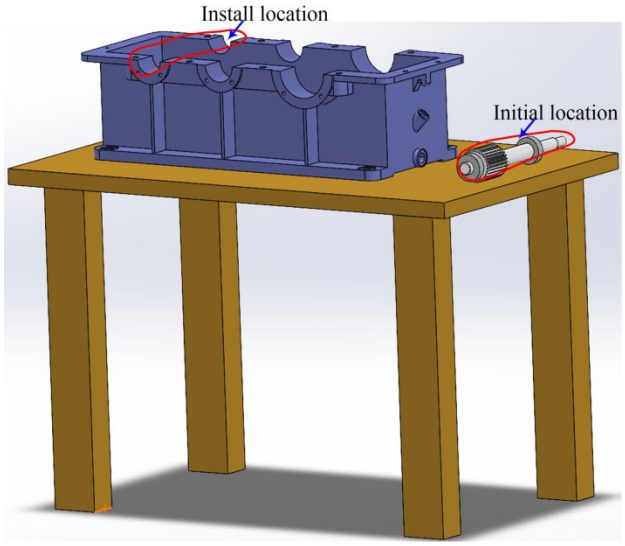

FIGURE 12. Installation task of reducer shaft.

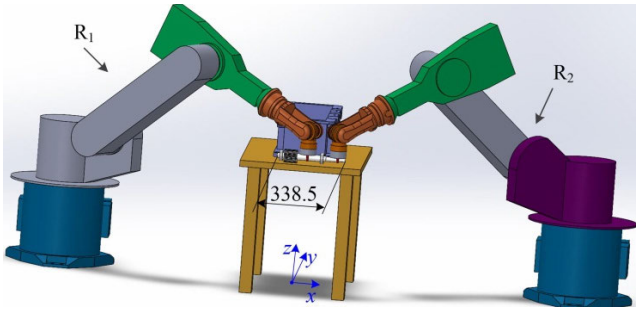

FIGURE 13. Installation scenario of double robots.

Fig. 13 shows the installation scenario of double robots. Two homogeneous KUKA industrial robots are employed to accomplish this installation process. The global coordinate system is established on the projection point of the worktop center as the original point. The total length of the shaft is 338.5mm. The workspace is $\mathbf{W}_{\min} = (-300, -500, 661)$ and $\mathbf{W}_{\max} = (300, 500, 1000)$. The starting point and target point of R_1 and R_2 are $S_1 = (25.25, -430, 693)$, $T_1 = (25.25, 270, 860)$ and $S_2 = (175.25, -430, 693)$, $T_2 = (175.25, 270, 860)$. The starting point and target point are the axis position points corresponding to the grasping position.

The gearbox is regarded as an obstacle during the shaft moving, which can be modelled as cuboid box. The diagonal coordinates of the cuboid box are $(-150.25, -271.5, 860)$ and $(150.25, 441, 661)$. In this engineering example, the collision detection is that the endpoints of the shaft on the straight line of two corresponding path points are not in the area of the obstacle box. However, the target point is located in the cuboid box of obstacle envelope. Thus, the collision detection is relatively complex. Based on the preliminary analysis, the collision free path planning can be divided into three parts according to the actual environment of shaft installation. Moreover, the size of the bearing, gear and shaft sleeve installed on the shaft need to be considered, and the maximum diameter is 32 mm. Consequently, the spatial scope of the three sections can be defined by

$$\begin{aligned} &1\text{st} : x \in \forall, y \in [-430, -305], z \in [693, 1000] \\ &2\text{st} : x \in \forall, y \in [-305, 270], z \in [892, 1000] \\ &3\text{st} : x \in [25.25, 175.25], y = 270, z \in [860, 1000] \end{aligned} \quad (33)$$

In the 1st path, the starting point and the target point of are set as $S_{11} = (25.25, -430, 693)$, $T_{11} = (25.25, -305, 900)$ and $S_{12} = (175.25, -430, 693)$, $T_{12} = (175.25, -305, 900)$. Since the z-value and y-value of path L_2 changes along the positive direction of the z-axis and y-axis, the sampling parameters are defined by $\theta \in [0, \frac{\pi}{2}]$ and $\varphi \in [0, \pi]$. In the 2st path, the starting point and the target point are set as $S_{21} = (25.25, -305, 900)$, $T_{21} = (25.25, 270, 900)$ and $S_{22} = (175.25, -305, 900)$, $T_{22} = (175.25, 270, 900)$. The z-value and y-value of path L_2 also change only along the positive direction of the z-axis and y-axis, so the sampling parameters are also defined by $\theta \in [0, \frac{\pi}{2}]$ and $\varphi \in [0, \pi]$. In the 3st path, the starting point and the target point are set as $S_{31} = (25.25, 270, 900)$, $T_{31} = (25.25, 270, 860)$ and $S_{32} = (175.25, 270, 900)$, $T_{32} = (175.25, 270, 860)$. In this part, the x-value and y-value of path L_2 are only taken as 175.25 and 270, respectively, and the z-value of path L_2 changes along the negative direction of z-axis. While the x-value and y-value of path L_1 are only taken as 25.25 and

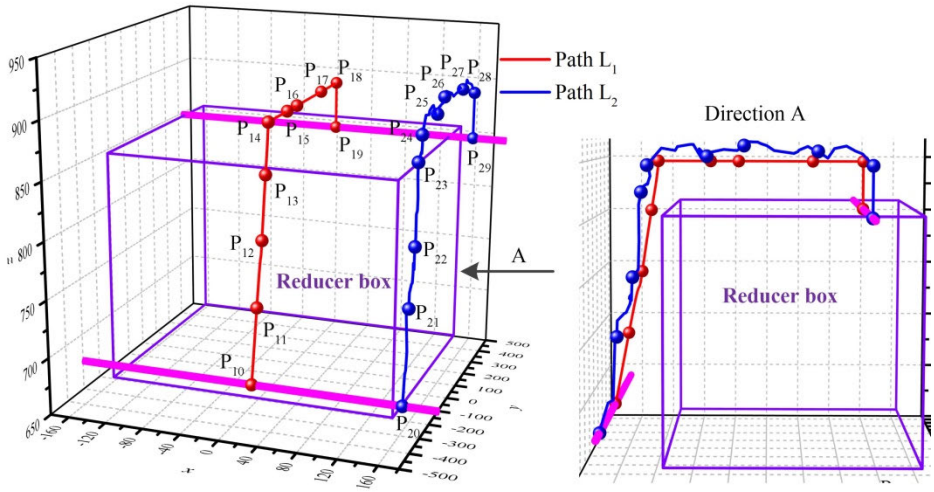

FIGURE 14. Dual path of installation of reducer shaft.

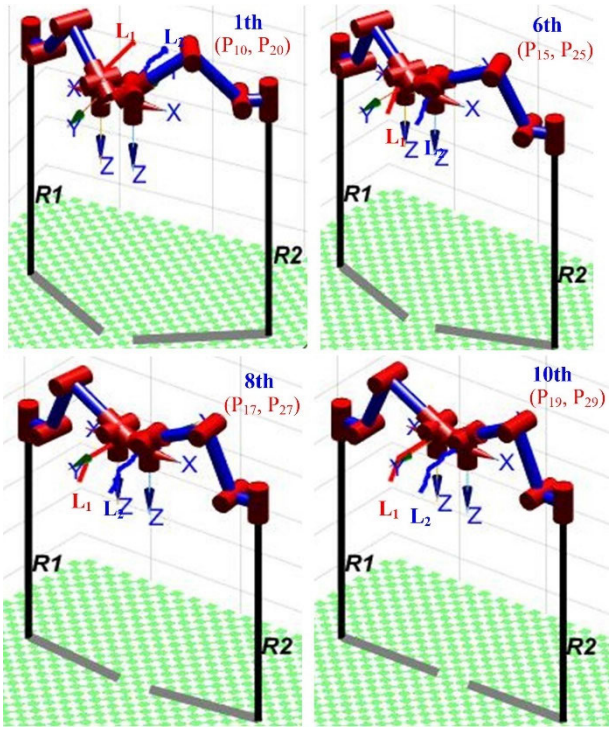

FIGURE 15. Installation process by MATLAB Robotics Toolbox.

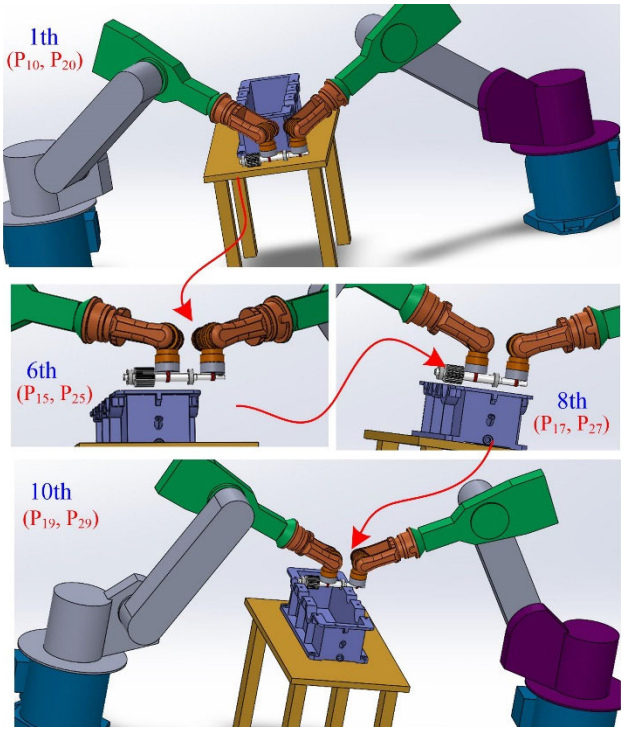

FIGURE 16. Installation process by Adams.

270, and the z-value of path L_1 also changes along the negative direction of z-axis. Consequently, the path of this part is a straight path moving along the negative direction of the z-axis. The dual path of installation of reducer shaft obtained by the proposed RSPSO is shown in Fig. 14.

As shown in Fig. 14, the reducer box (the rectangular box surrounded by purple straight lines) is the obstacle. The red line is the path L_1 planned for the first robot, while the blue line is the path L_2 planned for the second robot. The shaft marked by the magenta straight line is moved from the position of $P_{10}P_{20}$ to the position of $P_{19}P_{29}$ along two planned paths.

After the paths L_1 and L_2 are obtained, the joint space planning is performed. In this stage, the collision detections between robotic arms and gearbox, as well as the two robotic arms, need to be performed to prevent collisions. During the simulation by MATLAB, the size of the robots is added to the size of obstacles. Then, the collision detection can be completed. The joint angles by the second stages corresponding to the path points marked in Fig. 14 are shown in Table 6.

The inverse kinematics solution in Table 6 is used to complete the simulation of the dynamic installation process by MATLAB Robotics Toolbox and Adams, respectively. The simulations corresponding to the 1th, 6th, 8th and 10th markers are shown in Fig. 15 and Fig. 16.

From Fig. 15, the two robots both move from the starting point to the target point along the planned path L_1 and L_2. Moreover, neither the shaft nor the robots collided with the gearbox as shown in Fig. 16. The shaft is successfully installed to the designated location.

C. EXPERIMENT VALIDATION

According to the installation scenario shown in Part B, the experiment in the real environment is designed to further verify the practicality of the proposed algorithm. However, limited by our local experiment condition, the actual implementation makes some minor changes to the installation scenario, which is not exactly the same as the installation scenario shown in Fig. 12. Meanwhile, two different robots (that is, a Kuka robot and an ABB robot) are adopted to together realize the experiment. The experiment results are shown in Fig. 17 and Table 7.

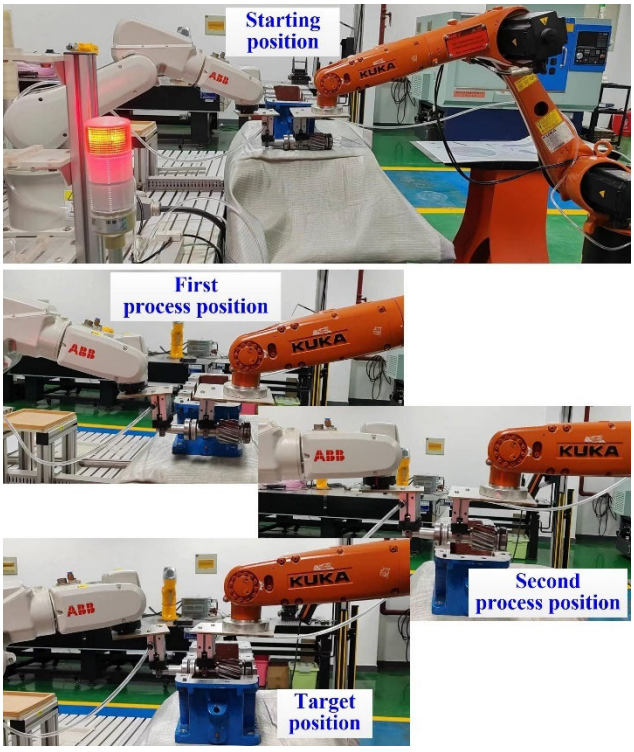

FIGURE 17. Installation corresponding to four positions.

In Fig. 17 and Table 7, the states of robot in the starting position, target position and two process positions are displayed. In Table 7, the first value is the joint angle of ABB robot, while the second value is the joint angle of Kuka robot. The path and the joint angle obtained by the proposed method are verified by the actual installation process of two robots. The planned two paths are verified to be collision-free, and the two robots can successfully achieve collaborative installation adopting the planned joint angles. The experiment results demonstrate that this scenario is successfully solved by the proposed method. Compared with the path obtained from manual teaching, the proposed method can be more flexible to obtain the dual paths for two co-worked robots.

TABLE 7. Joint angles corresponding to four positions.

Position	Joint angle (deg)					
	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
Starting	−164.35,	61.41,	−39.3,	3.12,	75.49,	0.48,
	−92.36	−55.74	69.97	3.34	−14.74	−30.51
First	−164.35,	53.14,	−35.95,	3.12,	75.85,	0.48,
	−92.92	−58.01	67.51	3.34	−13.36	−30.51
Second	−158.32,	50.84,	−53.51,	3.13,	89.46,	12.42,
	−91.06	−54.29	52.31	3.34	−3.16	−30.51
Target	−149.04,	70.03,	−76.37,	3.13,	89.45,	12.42,
	−85.3	−53.1	54.75	3.34	−5.26	−30.51

V. CONCLUSION

This paper presents a two stage path planning method that includes the dual path planning with the distance constraint of first stage and the joint space planning of second stage. The advantages of the proposed method include the following three aspects. (1) Both the optimal path of double end-effectors and the angular displacements of robotic arms corresponding to end-effectors can be simultaneously obtained by suing the two stage path planning method. (2) The optimal paths can guarantee the constant distance between the two end-effectors, which is due to the sampling operation on a spherical surface in the developed RSPSO. (3) The avoidance of obstacle collision and the avoidance of inter-robot collision are both taken into account in joint space planning, and therefore, the angular displacements with collision avoidance for dual robotic arms can be obtained.

Two simulation examples and an experiment are used to verify the proposed method, in which the one example is used to test the developed RSPSO of the first stage and the others are used to examine the effectiveness of the two stage path planning method in engineering application. The examples demonstrate that RSPSO algorithm can effectively obtain the dual path planning under the fixed distance constraints, and the proposed two stage path planning method can effectively realize the path planning task for engineering application of co-worked double industrial robots. Therefore, the proposed method can be regarded as an effective alternative in path planning for the co-worked double industrial robots.

However, the proposed method still exists the following drawbacks. (1) For the first stage, the distance constraint of two paths is ensured by fixed the distance of corresponding path points from one path to another in the proposed dual path planning method. When the path is longer and the number of path points is smaller, the distance from the previous path point to the next path point is larger. In this case, the distance

of the end-effectors may change in the process of the robotic arms moving from the previous path point to the next path point. This will cause the co-worked task to fail. However, when the number of path points is much more, the time of the dual path planning and the time of the joint space planning increase significantly. (2) For the second stage, the joint space planning only considers the final state angles of robotic arms in current proposed method. The motion planning of two robots and the simultaneous control of two robots will affect the success rate of the collaborative work. (3) The selection of the PSO performance parameters has significant impact on the application of the proposed method in robot path planning. It is necessary to propose an algorithm to select the most suitable parameters to improve PSO algorithm.

These drawbacks will restrict the in-depth application of the proposed method in practical engineering. Therefore, the proposed method requires further research around these drawbacks in order to better serve engineering applications, which are also our next work in the future.

REFERENCES

[1] M. Javaid, A. Haleem, R. P. Singh, and R. Suman, "Substantial capabilities of robotics in enhancing industry 4.0 implementation," *Cognit. Robot.*, vol. 1, pp. 58–75, Oct. 2021.

[2] Z. Feng, G. Hu, Y. Sun, and J. Soon, "An overview of collaborative robotic manipulation in multi-robot systems," *Annu. Rev. Control*, vol. 49, pp. 113–127, Jan. 2020.

[3] H. Wang, H. Wang, J. Huang, B. Zhao, and L. Quan, "Smooth point-to-point trajectory planning for industrial robots with kinematical constraints based on high-order polynomial curve," *Mechanism Mach. Theory*, vol. 139, pp. 284–293, Sep. 2019.

[4] K. Kaltsoukalas, S. Makris, and G. Chryssolouris, "On generating the motion of industrial robot manipulators," *Robot. Comput.-Integr. Manuf.*, vol. 32, pp. 65–71, Apr. 2015.

[5] I. Solis, J. Motes, R. Sandström, and N. M. Amato, "Representation-optimal multi-robot motion planning using conflict-based search," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 4608–4615, Jul. 2021.

[6] G. Wagner and H. Choset, "Subdimensional expansion for multirobot path planning," *Artif. Intell.*, vol. 219, pp. 1–24, Feb. 2015.

[7] G. Wagner, M. Kang, and H. Choset, "Probabilistic path planning for multiple robots with subdimensional expansion," in *Proc. IEEE Int. Conf. Robot. Autom.*, Saint Paul, MN, USA, May 2012, pp. 2886–2892.

[8] R. Shome, "Roadmaps for robot motion planning with groups of robots," *Current Robot. Rep.*, vol. 2, no. 1, pp. 85–94, Mar. 2021.

[9] M. Čáp, P. Novák, J. Vokrínek, and M. Pěchouček, "Multi-agent RRT: Sampling-based cooperative pathfinding," in *Proc. 12th Int. Conf. Auto. Agents Multiagent Syst.*, Saint Paul, MN, USA, May 2013, pp. 480–481.

[10] R. Shome, K. Solovey, A. Dobson, D. Halperin, and K. E. Bekris, "DRRT: Scalable and informed asymptotically-optimal multi-robot motion planning," *Auto. Robots*, vol. 44, nos. 3–4, pp. 443–467, Mar. 2020.

[11] F. A. Raheem and I. I. Gorial, "Comparative study between joint space and Cartesian space path planning for two-link robot manipulator using fuzzy logic," *Iraqi J. Comput., Commun. Control Syst. Eng.*, vol. 13, no. 2, pp. 1–10, 2013.

[12] Y.-A. Lu, K. Tang, and C.-Y. Wang, "Collision-free and smooth joint motion planning for six-axis industrial robots by redundancy optimization," *Robot. Comput.-Integr. Manuf.*, vol. 68, Apr. 2021, Art. no. 102091.

[13] B. Su and L. Zou, "Manipulator trajectory planning based on the algebraic-trigonometric Hermite blended interpolation spline," *Proc. Eng.*, vol. 29, pp. 2093–2097, Jan. 2012.

[14] Y. Pu, Y. Shi, X. Lin, W. Zhang, and P. Zhao, "Joint motion planning of industrial robot based on modified cubic Hermite interpolation with velocity constraint," *Appl. Sci.*, vol. 11, no. 19, p. 8879, Sep. 2021.

[15] X. Liu, C. Qiu, Q. Zeng, A. Li, and N. Xie, "Time-energy optimal trajectory planning for collaborative welding robot with multiple manipulators," *Proc. Manuf.*, vol. 43, pp. 527–534, Jan. 2020.

[16] E. Glorieux, S. Riazi, and B. Lennartson, "Productivity/energy optimisation of trajectories and coordination for cyclic multi-robot systems," *Robot. Comput.-Integr. Manuf.*, vol. 49, pp. 152–161, Feb. 2018.

[17] J. Shu, W. Li, and Y. Gao, "Collision-free trajectory planning for robotic assembly of lightweight structures," *Autom. Construct.*, vol. 142, Oct. 2022, Art. no. 104520.

[18] J.-H. Chen and K.-T. Song, "Collision-free motion planning for human–robot collaborative safety under Cartesian constraint," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Brisbane, QLD, Australia, May 2018, pp. 4348–4354.

[19] Y. Liu, F. Zha, M. Li, W. Guo, Y. Jia, P. Wang, Y. Zang, and L. Sun, "Creating better collision-free trajectory for robot motion planning by linearly constrained quadratic programming," *Frontiers Neurorobot.*, vol. 15, Aug. 2021, Art. no. 724116.

[20] B. Tang, K. Xiang, M. Pang, and Z. Zhanxia, "Multi-robot path planning using an improved self-adaptive particle swarm optimization," *Int. J. Adv. Robot. Syst.*, vol. 17, no. 5, Sep. 2020, Art. no. 172988142093615.

[21] L. Larsen and J. Kim, "Path planning of cooperating industrial robots using evolutionary algorithms," *Robot. Comput. Integr. Manuf.*, vol. 67, Feb. 2021, Art. no. 102053.

[22] X. Zhou, X. Wang, Z. Xie, F. Li, and X. Gu, "Online obstacle avoidance path planning and application for arc welding robot," *Robot. Comput.-Integr. Manuf.*, vol. 78, Dec. 2022, Art. no. 102413.

[23] X. Wang, B. Tang, Y. Yan, and X. Gu, "Time-optimal path planning for dual-welding robots based on intelligent optimization strategy," in *Transactions on Intelligent Welding Manufacturing*. Singapore: Springer, 2018, pp. 47–59.

[24] J. Lian, W. Yu, K. Xiao, and W. Liu, "Cubic spline interpolation-based robot path planning using a chaotic adaptive particle swarm optimization algorithm," *Math. Problems Eng.*, vol. 2020, pp. 1–20, Feb. 2020.

[25] T. Zhang and F. OuYang, "Kinematics analysis and path planning of dual-robot coordinated couple motion," *J. Shanghai Jiaotong Univ.*, vol. 47, no. 8, pp. 1251–1256, 2013.

[26] Y. Tang, J. Hou, T. Wu, S. Gong, J. Zhang, and L. Zhong, "Hybrid collision detection algorithm based on particle conversion and bounding box," *J. Harbin Eng. Univ.*, vol. 39, no. 10, pp. 1695–1701, 2018.

[27] C. Villaseñor, A. A. Gallegos, G. Lopez-Gonzalez, J. Gomez-Avila, J. Hernandez-Barragan, and N. Arana-Daniel, "Ellipsoidal path planning for unmanned aerial vehicles," *Appl. Sci.*, vol. 11, no. 17, p. 7997, Aug. 2021.

[28] Y. Song, L. Zhang, R. Tian, and X. Wang, "Research on obstacle avoidance motion planning method of manipulator in complex multi scene," *J. Northwestern Polytech. Univ.*, vol. 41, no. 3, pp. 500–509, Jun. 2023.

LI-XIANG ZHANG was born in Baoding, Hebei, China, in 1982. She received the B.S. and M.S. degrees in mechanical engineering from Yanshan University, Qinhuangdao, China, in 2006 and 2009, respectively, and the Ph.D. degree in mechanical manufacturing and automation from China Agricultural University, Beijing, China, in 2014. Since 2016, she has been a Teacher with the School of Mechanical and Equipment Engineering, Hebei University of Engineering, where she is currently an Associate Professor. Her research interests include machine design, reliability-based design optimization, industrial robot technology, and artificial intelligence.

XIN-JIA MENG was born in Baoding, Hebei, China, in 1982. He received the B.S. degree from the Hebei University of Architecture, in 2006, the M.S. degree from Yanshan University, in 2009, and the Ph.D. degree from the School of Mechanical Engineering, Beijing Institute of Technology, in 2016. Since 2016, he has been a Teacher with the School of Mechanical and Equipment Engineering, Hebei University of Engineering, where he is currently an Associate Professor. His research interests include machine design, design optimization, reliability-based design, and industrial robot technology.

ZHI-JIE DING was born in Fengcheng, Jiangxi, China, in 2001. He received the B.S. degree from the Hebei University of Engineering, in 2022, where he is currently pursuing the M.S. degree in mechanical engineering. His primary research interests include machine design, computer simulation, and industrial robot technology.

TIAN-SHU WANG was born in Baoding, Hebei, China, in 2002. She is currently pursuing the B.S. degree with the Beijing University of Chinese Medicine. Her current research interests include medical treatment techniques, computer simulation, and artificial intelligence.