applied sciences

MDPI

Article

Modular Robotic Design and Reconfiguring Path Planning

Ye Dai *, Chao-Fang Xiang, Zhao-Xu Liu, Zhao-Long Li, Wen-Yin Qu and Qi-Hao Zhang

Key Laboratory of Advanced Manufacturing Intelligent Technology of Ministry of Education, Harbin University of Science and Technology, Harbin 150080, China; xcf1197069305@163.com (C.-F.X.); yxx2314257928@163.com (Z.-X.L.); lizhaolong@hrbust.edu.cn (Z.-L.L.); qudafaer888@163.com (W.-Y.Q.); zhangqihao7777@163.com (Q.-H.Z.)

* Correspondence: daiye312@163.com

Abstract: The modular robot is becoming a prevalent research object in robots because of its unique configuration advantages and performance characteristics. It is possible to form robot configurations with different functions by reconfiguring functional modules. This paper focuses on studying the modular robot's configuration design and self-reconfiguration process and hopes to realize the industrial application of the modular self-reconfiguration robot to a certain extent. We design robotic configurations with different DOF based on the cellular module of the hexahedron and perform the kinematic analysis of the structure. An innovative design of a modular reconfiguration platform for conformational reorganization is presented, and the collaborative path planning between different modules in the reconfiguration platform is investigated. We propose an optimized ant colony algorithm for reconfiguration path planning and verify the superiority and rationality of this algorithm compared with the traditional ant colony algorithm for platform path planning through simulation experiments.

Keywords: modular robot; configuration design; kinematic analysis; self-reconfiguration platform; path planning; optimized ant colony algorithm

check for updates

Citation: Dai, Y.; Xiang, C.-F.; Liu, Z.-X.; Li, Z.-L.; Qu, W.-Y.; Zhang, Q.-H. Modular Robotic Design and Reconfiguring Path Planning. *Appl. Sci.* **2022**, *12*, 723. https://doi.org/10.3390/app12020723

Academic Editors: Augusto Ferrante, Mingcong Deng and Mihaiela Iliescu

Received: 15 December 2021
Accepted: 6 January 2022
Published: 12 January 2022

1. Introduction

In terms of the individual organization, existing robots may be divided into single-configuration robots and modular robots. The single-configuration robot is meant to do a single repeated duty. Still, the modular machine may modify its configuration automatically or artificially depending on the task or the working environment [1]. In terms of reconfigurability, expandability, and manufacturing simplicity, the modular robot outperforms the traditional robot.

At the present stage of the industry, welding robots are mainly used in the automotive manufacturing industry. They can perform all kinds of welding work manually to a high standard, with the advantages of high efficiency and good welding quality [2]. Most of the traditional robotic designs applied in industrial manufacturing are mechanical arms, which serve specific tasks and the corresponding applications are not flexible for other activities [3]. On the other hand, modern companies have a large space occupation of traditional robotic arms in the case of many devices. The self-reconfiguring robot can self-configure according to the task requirements and thus accomplish specific tasks independent of the environment. Due to its small space occupation, it can be better applied to assembly, equipment inspection, and automotive interior parts welding. The key technical areas of current research on self-reconfiguring robots are focused on the structural design of reconfiguration modules, robotic configuration analysis, and reconfiguration path planning [4,5].

In terms of research on self-reconfiguring robotic configurations, the concept of a dynamically reconfigurable robot system (Cellular robotic system), which can reconfigure and repair itself according to the task or work environment, was first proposed by Professor Toshio Fukuda in Japan in 1988 [6]. Subsequently, some modular chain robots were presented, such as Poly Bot [7] developed by Mark Yim and CONRO developed by Shen [8]. In the industrial application, Acaccia et al. [9] developed a generic modular robot system driving the development of self-reconfiguring modular robots.

The robotic reconfiguration strategy mainly includes conformational path planning and motion obstacle avoidance. Ant colony algorithm is a heuristic intelligent search algorithm that has been well used in solving path planning problems [10]. Compared with other intelligent algorithms, the ant colony algorithm has strong robustness and can easily achieve the fusion of different algorithms [11]. By optimizing and improving the traditional ant colony algorithm, we can effectively reduce the disadvantages of slow convergence and falling into local optimality [12]. The traditional ant colony algorithm is difficult to apply alone because of the disadvantages of easily falling into local optimum and difficulty in getting rid of dead points. Most of the studies have optimized the ACO (Ant colony optimization) or fused other algorithms for path planning. The following three improvement methods are commonly used: improvement based on the ACO's model, a combination of ACO and similar algorithms, and fusion of ACO and other algorithms [13]. Jiang et al. [14] proposed an improved ant colony optimization algorithm to improve the global nature of the algorithm by improving the quality ant pheromone principle, which effectively improved the efficiency of the robot. Deng et al. [15] fused the ant colony algorithm with a genetic algorithm to study the application of the TSP (Traveling salesman problem) and finally verified the advantages of the fusion algorithm. In addition, Tao et al. [16] fully integrated the ant colony algorithm with machine learning and control strategy algorithms for intelligent optimization. Niu et al. [17] proposed a pheromone volatile factor adaptive strategy to improve the global search capability of the algorithm. However, it lacks the local search optimization capability and the improvement of algorithm search efficiency is very limited. Li et al. [18] combined the artificial bee colony algorithm to propose the optimization-seeking ants and the reconnaissance ants, which were given different pheromone update weights to improve the convergence speed of the algorithm, avoid the algorithm to fall into local optimum. However, the inability to deal with deadlocked ants or replenish the number of ants causes the algorithm to fail easily in path planning in complex environments.

Obstacle avoidance is a fundamental problem of path planning has also achieved a lot of research results nowadays. Anh et al. [19] proposed a new global path planning method for a self-reconfiguring robot based on the A-Star algorithm, which aims to cover a narrow space by generating path points with maximum coverage area to ensure excellent area coverage performance. Sun et al. [20] proposed an improved artificial potential field method to consider the distance between the robot and obstacles in planning. Liang et al. [21] proposed an EABC (Efficient artificial bee colony) algorithm with an obstacle avoidance policy for solving online path planning of multiple mobile devices. Qi et al. [22,23] proposed a probabilistic-based obstacle avoidance trajectory planning method that uses Gaussian motion prior probability estimation to obtain the optimal path trajectory. Although the path planning method based on the instructional programming model is feasible, it isn't easy to get the optimal path, and path planning is too time-consuming.

As shown above, although a lot of researches have been conducted on the path planning problem, the above research still has the following characteristics: (1) There is no combination of path optimization and obstacle avoidance problems, which leads to the risk of collision or even damage when the robot is performing path planning and has limitations. (2) The collision-free path planning takes too long, which leads to multiple repetitions of the path when the robot performs path planning, and the movement is inefficient. (3) Lacking handling of lock-ups during the path planning, which makes the path planning prone to failure.

The main objective of this paper is to build a reconfiguration platform for functional modules and design an optimized algorithm so that functional modules can find an optimal path in the platform to reconfigure the target configuration. We improve the movement efficiency of the reconfigured robot from the initial position to the working target position by optimizing the ant colony algorithm, reducing the unnecessary movement distance, and rationalizing obstacle avoidance. The paper is organized as follows: the first part introduces the advantages of the reconfiguration robot and the real-world application in self-reconfiguration planning. The second part investigates the module design and configuration modeling of self-reconfiguring robots. In the third part, a reconfiguration platform for functional modules is established, and the types of obstacle avoidance in the process of modules reconfiguration are defined. Subsequently, an optimized ant colony algorithm is designed based on the principle of obstacle avoidance for the path planning of modules of the reconfiguration platform. In the fourth part, simulation experiments of the platform path planning are conducted to compare and analyze the parameters of reconstruction time, obstacle avoidance time, and path planning length with the traditional ant colony algorithm. The fifth part concludes the paper and gives an outlook on future development.

2. Robotic Design and Modeling

2.1. Design of the Robotic Function Module

The robot mentioned in this paper is a multi-level self-reconfiguring modular robot with high reliability, changeable configurations, and easily expandable functions. The first stage of a mechanical configuration design is from the perspective of the module cell assembly, considering the ease of installation, maintenance, and replacement [24]. The design process of the target robot follows the multilevel theory, which develops a design concept of a "cell-tissue-organ-robot" for a reconfigurable modular robot. The robot is composed of various basic cellular functional modules; the cellular modules can form tissues with different functions; then, the different tissues form specific organs to complete the conformational reorganization.

The functional modules set in this paper are designed in a uniform size of 200 mm in an ortho-hexahedron structure. Due to the ortho-hexahedron module's regular geometric shape and space-filling nature, they can be combined into a stable overall configuration [25]. Different functional modules are integrated into different functional robotic configurations using a combination of active-and-passive slotted connections, as shown in Figure 1. The function modules can be divided into the interstitial function modules, swing function modules, rotation function modules, clamping function modules, etc. In addition, some auxiliary modules realize specific tasks, such as sensor modules, compensation function modules, monitoring and identification modules, etc.

2.2. Conformation Design Scheme for the Reconfigurable Robot

From the perspective of bionics, the modular robot designed in this paper adopts a tandem [26] structural design, which enables the robot to achieve multi-attitude motion flexibly and is suitable for the machine assembly of the machine. In addition, the concept of symmetric design is used in the structure [27], which is widely used in mechanical engineering due to its advantages of balanced force and manufacturing efficiency. The symmetric design is well suited for the modular reconfigurable robot. Based on the above considerations, this paper mainly sets the reconfigured robotic configurations with different DOF (Degree of freedom) as shown in Table 1.

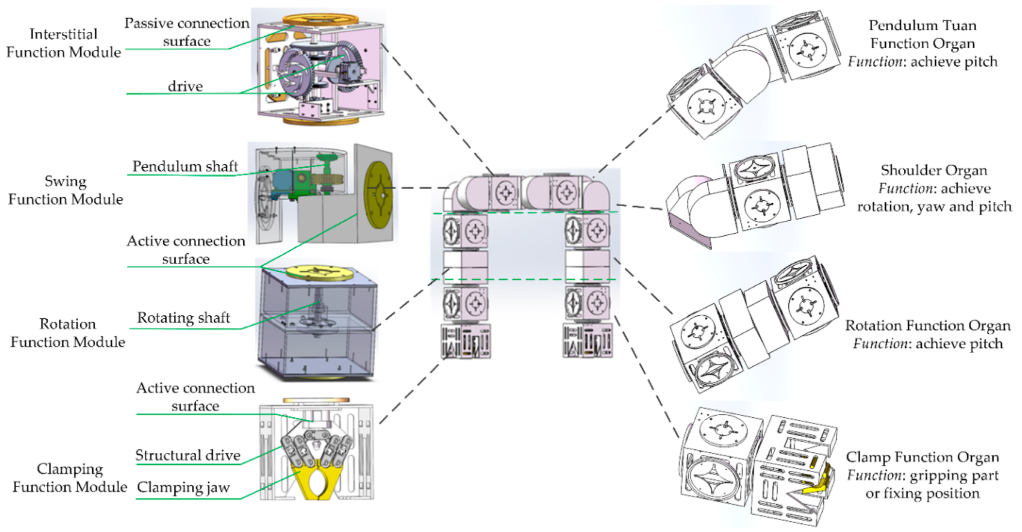
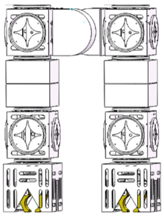
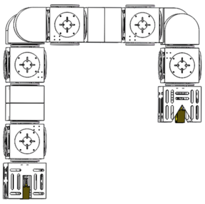
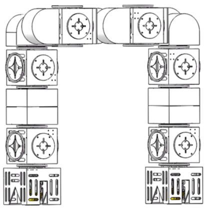
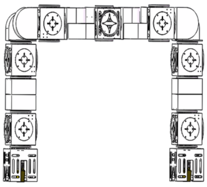
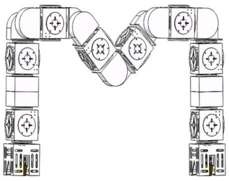

Interstitial Function Module
Passive connection surface
drive
Swing Function Module
Pendulum shaft
Active connection surface
Rotation Function Module
Rotating shaft
Active connection surface
Clamping Function Module
Structural drive
Clamping jaw
Pendulum Tuan Function Organ
Function: achieve pitch
Shoulder Organ
Function: achieve rotation, yaw and pitch
Rotation Function Organ
Function: achieve pitch
Clamp Function Organ
Function: gripping part or fixing position

Figure 1. Module composition of the self-reconfiguring robot.

Table 1. Self-reconfiguring robot configuration design scheme.

DOF	3-D Model	Components	Characteristics
Three		2 rotating modules 1 swing module 2 clamping modules 4 interstitial modules	It has low freedom for simple grasping or climbing.
Four		2 rotating modules 2 swing modules 2 clamping modules 5 interstitial modules	With a certain degree of flexibility, and it can complete the position adjustment.
Five		3 rotating modules 2 swing modules 2 clamping modules 6 interstitial modules	High flexibility for most basic tasks.
Six		2 rotating modules 4 swing modules 2 clamping modules 7 interstitial modules	Meet the needs of free space for installation in any position
Seven		2 rotating modules 5 swing modules 2 clamping modules 8 interstitial modules	Consider the advantages of redundant motility

2.3. Kinematic Analysis

Before a module is assembled, the robot's configuration needs to be described in the form of a mathematical description. It is used as the module's assembly target to guide the module through the reconfiguration tasks. The configuration description of a modular robot needs contains essential information about the module parameters and the inter-module connection relationships [28]. We select the five-DOF reconfigured robot for kinematic analysis. In comparison, it can meet most of the requirements without excessive corresponding loads on the module joints. It consists of three rotating modules, two swing modules, two clamping modules, and six interstitial modules. The coordinate system is established at the geometric center point of each functional module according to the robot configuration characteristics, as shown in Figure 2.

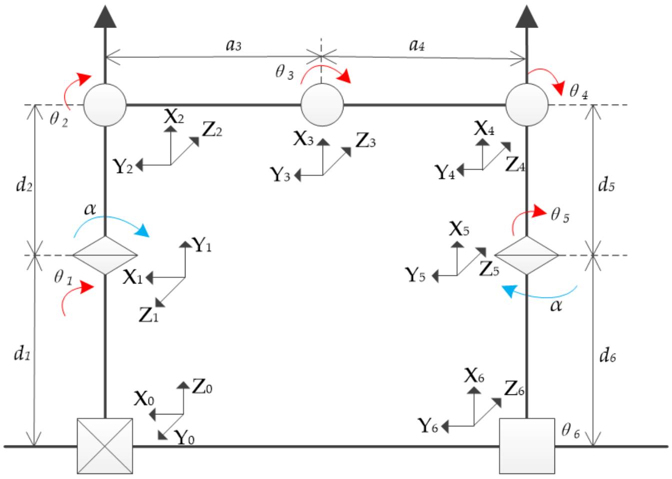


Figure 2. Schematic diagram of modular robotic D-H coordinate.

Set the clamping function module at the left end of the reconstructed robot as the fixed end, and the right end as the free gripping end. The D-H parameter table of the five-DOF configuration robot is analyzed, as shown in Table 2.

Table 2. Modular robotic D-H parameter table.

Joint	Turn α	Length a	Offset d	Angle θ	Range
Module 1	0°	0	d_1+d_2	θ_1	−180~180°
Module 2	90°	0	0	θ_2	−90~90°
Module 3	0°	a_3	0	θ_3	−90~90°
Module 4	0°	a_4	0	θ_4	−90~90°
Module 5	−90°	0	0	θ_5	−180~180°
Module 6	0°	0	d_5+d_6	θ_6	0°

The spatial positional information of the reconfigured robot is expressed by resolving the positional transformations between functional modules by the alignment transformation matrix. The alignment transformation matrix of the reconfigured robot is:

$$A_{n+1} = {}_{n+1}^{\ \ n}T = \begin{bmatrix} cos\theta_{n+1} & -sin\theta_{n+1}cos\alpha_{n+1} & sin\theta_{n+1}sin\alpha_{n+1} & a_{n+1}cos\theta_{n+1} \\ sin\theta_{n+1} & cos\theta_{n+1}cos\alpha_{n+1} & -cos\theta_{n+1}sin\alpha_{n+1} & a_{n+1}sin\theta_{n+1} \\ 0 & sin\alpha_{n+1} & cos\alpha_{n+1} & d_{n+1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

When the clamping module of the end of the reconfigured robot is clamped and fixed, the velocity of the end module of the reconfigured robot in the reference coordinate system can be known by the reference [29] and is expressed:

$$\dot{p} = [\begin{matrix} v_e \\ \omega_e \end{matrix}] = J \begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_5 \end{bmatrix} = J[\dot{\theta}] \tag{2}$$

where: "$\dot{p}$" is the velocity of the reconfigured robotic functional module; "v_e" is the linear velocity of the robotic end clamping module; "ω_e" is the angular velocity of the robotic end clamping module; "J" is the Jacobi matrix; "$[\dot{\theta}]$" is the angular velocity vector of the reconfigured robotic joint.

The Jacobi matrix "J" can be expressed by Equation (3).

$$J(\theta) = \begin{bmatrix} J_{v1} & J_{v2} & \cdots & J_{v5} \\ Q_1 & Q_2 & \cdots & Q_5 \end{bmatrix} \tag{3}$$

in the formula:

$$Q_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix};\ Q_2 = \begin{bmatrix} s_1 \\ c_1 \\ 0 \end{bmatrix}; Q_3 = \begin{bmatrix} s_1 \\ -c_1 \\ 0 \end{bmatrix}; Q_4 = \begin{bmatrix} s_1 \\ c_1 \\ 0 \end{bmatrix}; Q_5 = \begin{bmatrix} c_1c_{234} \\ s_1c_{234} \\ -s_{234} \end{bmatrix}$$

$$s_i = sin\theta_i;\ c_i = cos\theta_i\ ;$$

$$J_{v1} = \begin{bmatrix} s_1s_{234}+d_2c_2s_1+a_3c_{234} \\ (d_1+d_2)c_1s_{234}+d_2c_1s_2+a_3c_1c_2 \\ 0 \end{bmatrix}$$

$$J_{v2} = \begin{bmatrix} c_1(a_3(s_2+s_{23})+(d_1+d_2)c_{234}) \\ s_1(a_3(s_2+s_{23})+(d_1+d_2)c_{234}) \\ c_1((d_1+d_2)s_{234}+a_3c_2(c_2+c_{23})+s_1(d_1+d_2)c_1c_{234}+a_3c_1c_{234}) \end{bmatrix}$$

$$J_{v3} = \begin{bmatrix} c_1(a_3s_{23}+(d_1+d_2)c_{234}) \\ s_1(a_3s_{23}+(d_1+d_2)c_{234}) \\ c_1((d_1+d_2)c_1s_{23}+a_3c_1c_{234})+s_1((d_1+d_2)s_1s_{23}+a_3c_1s_{234}) \end{bmatrix}$$

$$J_{v4} = \begin{bmatrix} -(d_1+d_2)s_1s_{234} \\ -(d_1+d_2)c_1c_{234} \\ (d_1+d_2)s_{234} \end{bmatrix}; J_{v5} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

When the kinematics of the reconstructed robot is required, the coordinate system of the left or right clamping module is defaulted to the reference coordinate system according to the definition of robot motion. The joint angle "$\theta_i(i = 1, 2, \ldots, 5)$" is substituted into Equations (1)–(3) to complete the kinematics analysis of the robot.

3. Research on Path Planning Based on Module Reconfiguration Platform

The establishment of the target configuration of the reconfiguring robot requires the mobile splicing of different functional modules, which inevitably involves the application of path planning and related algorithms for the reconfiguration process. In this section, a platform for the module's self-reconfiguration is proposed and based on which the path planning algorithm is optimized.

3.1. Building of the Module Reconfiguration Platform

3.1.1. Reconfiguration Platform Introduction

The reconfiguration platform is mainly used to complete the mutual transformation between modules and configurations. Within the platform, modules are assembled into cellular robot configurations with different functions through reasonable obstacle avoidance strategies and path planning. The reconfiguration planning flow chart is shown in Figure 3. Firstly, the target configuration is set according to the task and a mathematical model is built. After determining the number of modules, connection interfaces, and connection relationships, the system transmits the information to the module warehouse. Then, the module warehouse performs the screening of modules and releases them to the module reconfiguration platform. Within the platform, the motion path map of the module is firstly constructed and the obstacle module is set up at the same time. Subsequently, a reasonable path planning of the 2D map is carried out, specifically including the development of obstacle avoidance guidelines and the design of innovative algorithms. Finally, the formation of the target configuration is completed within the platform. The completed reconfigured cell robot is transported to the robot's work area by the robotic transport system. In addition, the completed robots are again transported via the robotic transport system directly to the reconfiguration platform for disassembly of the conformation so that it can participate in the next stage of reconfiguration or be stored.

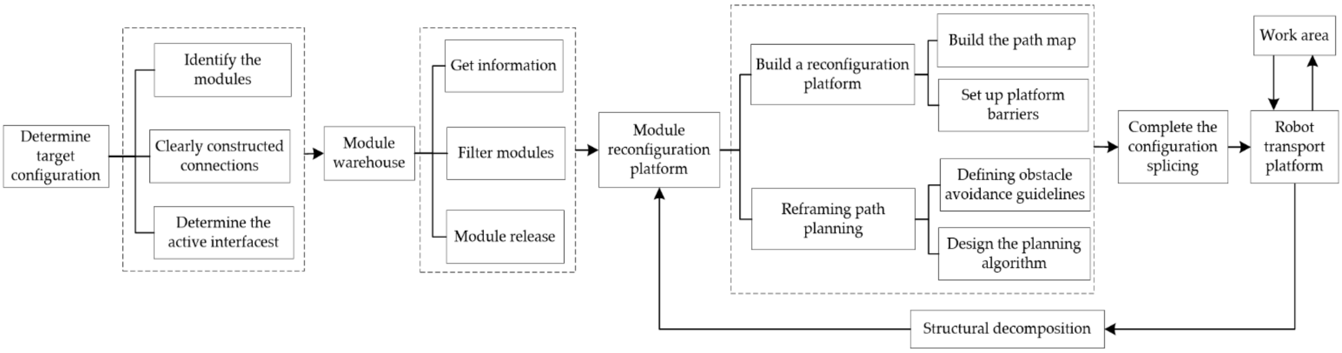

Determine target configuration
Identify the modules
Clearly constructed connections
Determine the active interfacest
Module warehouse
Get information
Filter modules
Module release
Module reconfiguration platform
Build a reconfiguration platform
Build the path map
Set up platform barriers
Reframing path planning
Defining obstacle avoidance guidelines
Design the planning algorithm
Complete the configuration splicing
Work area
Robot transport platform
Structural decomposition

Figure 3. Module reconstruction planning flowchart.

Based on the above analysis, we divide the functions of the platform. The overall distribution of the module reconfiguration system is shown in Figure 4. It consists of three parts: module warehouse, module reconfiguration platform, and robot transport platform. The module warehouse is mainly used for storing reconfiguration function modules. It screens and releases the corresponding number of modules to the reconfiguration platform according to the configuration requirements and transports the modules to the reconfiguration position following the configuration connection requirements through the three-DOF drive device to complete the docking between modules. The reconfigured robot that has completed its task can also be returned to the reconfiguration platform for de-configuration. The robot is deconstructed into separate functional modules and rejoined to the new robotic configuration tasks. The reconfigured robot on the platform is transported to the working position via a conveyor platform. The various parts of the platform are closely connected and work together to accomplish the task of reconstructing the platform through information communication and collaborative drive.

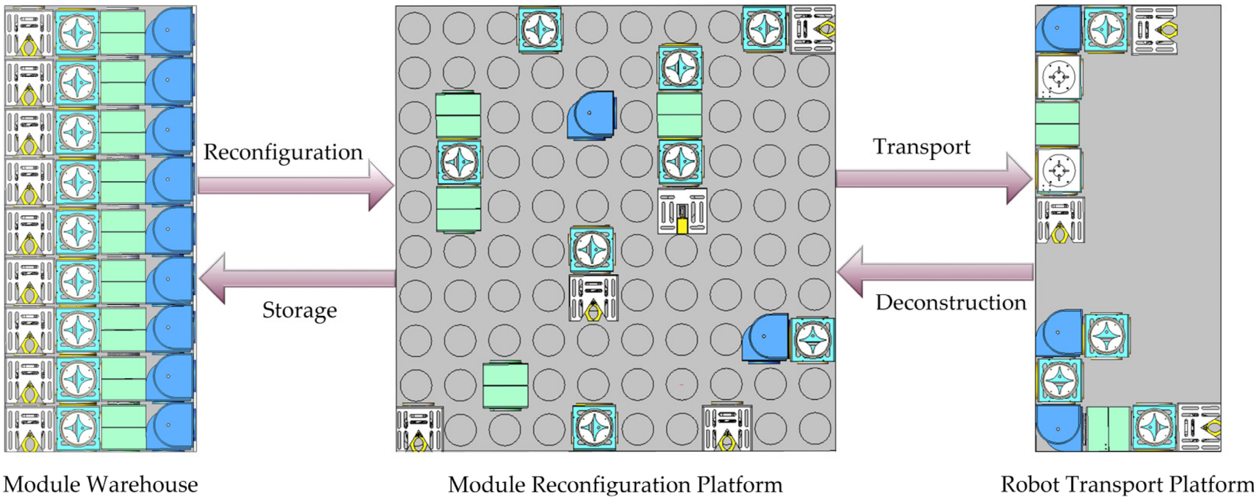


Figure 4. Distribution of the module reconfiguration system.

3.1.2. Mathematical Model of the Reconfiguration Platform

The raster method [30] is simple and easy to implement. It can describe the environment in the form of a grid. It simplifies the relationships between the adjacent grids to ensure the validity of the constructed environment. The modular reconfiguration platform is modeled using the raster method, which represents the platform as a "20 × 20" two-dimensional raster, and specifies that:

(1) The movement space of modules is a two-dimensional environment.
(2) Define the module's movement step; the module moves one grid distance at a time, sets the module's standard length to 200 mm, uniform motion for "v = 0.2 m/s".
(3) The coordinate of the module in the grid map is the center point of the grid.

When the reconfiguration platform receives the robotic reconfiguration signal, it finds a path that satisfies the planning conditions between the initial and target positions by the optimized ant colony algorithm. The global path planning is performed first, and then the local collision avoidance between modules is ensured by the module avoidance criterion to find the shortest path between the initial point and the target point of the modules.

The planning path "P_i" of the function module "i" in the reconfigured platform model is represented by the centroid vector coordinates in the raster model, $P_i = (P_{i1}, P_{i2}, \ldots, P_{il})$. The total length of the motion path of the functional module is:

$$min_{P_i} d = \sum_{i=1}^{i=n} \left|\left|P_i\right|\right| \tag{4}$$

where: "$||P_i||$" is the path length of the function module "i".

During the refactoring planning process, constraints need to be defined for the moving process of the functional modules, which includes the motion condition definition and the stop condition definition.

(1) Definition of the motion conditions: the reconfiguration function module is specified to carry out two-dimensional plane motion in the reconfiguration platform for the four directions of front, back, left, and right. The module moves in a certain direction if it meets the defined requirements.

$$|x_{a+1} - x_a| \le u, \forall\ u > 1,\ \ u \text{ is } a \text{ constant} \tag{5}$$

$$|y_{a+1} - y_a| \le u,\ \ \forall u > 1 \tag{6}$$

$$\sqrt{(x_{a+1} - x_a)^2 + (y_{a+1} - y_a)^2} \le u,\ \forall u > 1 \tag{7}$$

where: the *"a"* indicates the functional module moves to the *"step a"*; "x_a" and "y_a" indicate the horizontal and vertical coordinates of the module at the *"step a"*.

(2) Definition of the stop motion conditions: It is stipulated that the reconfiguration function module stops motion only when it moves to the target position. The condition for determining the module to stop is:

$$\left|\left(x_{a-1}+y_{a-1}\right)-\left(x_{a-2}+y_{a-2}\right)\right| \geq \left|\left(x_a+y_a\right)-\left(x_{a-1}+y_{a-1}\right)\right|, \forall\, a>2 \tag{8}$$

that is: when the functional module moves from step " $a-2$" to step " $a-1$" with 0 step, the functional module remains stationary and the module reaches the target position.

3.2. Module Reconfiguration Campaign Definition

3.2.1. Mobile Priority for Functional Modules

Since the transportation tasks of the module reconfiguration platform are complicated, to improve the efficiency and safety of the combined reconfiguration of the robot, this paper adopts multiple modules to transport simultaneously and complete the combined tasks collaboratively. Therefore, the priority of different modules in the reconfiguration process is set, which can ensure reasonable collision avoidance during the cooperative planning of modules.

When the module performs grid path search, if there is no module or obstacle in front of it, its state is defined as a free state; when other functional modules exist in front, the state is defined as a collision avoidance state. In case of collision, the module with a high priority level is defined to choose the path firstly. The module with a low priority level automatically avoids or waits for other modules to pass. The priority level is defined as:

(1) When different modules are selected, the priority order of the modules is clamping function module, rotation function module, swing function module, interstitial function module.

(2) If two modules are at the same priority level, the criteria for judging the priority level of the two modules is the distance for the module moves in the grid, and the module that moves farther away has a priority in choosing the path.

(3) If the two modules move the same distance and have the same priority level, the functional module that has participated in robotic deconstruction has a higher priority than the module in the reconfiguration bin.

During the module reconfiguration process, the two modules with higher priority carry out collaborative path planning, defining the other modules as obstacles. When the original modules complete the path planning, the subsequent modules with higher priority continue the path planning until the last module stops moving, the module path planning of the desired configuration is completed.

3.2.2. Establishment and Solution of the Functional Module Collision Avoidance Model

The reconfiguration platform plans the set of paths for n functional modules based on the number of modules of the reconfiguring robotic configuration, *"p_n = {p_1, p_2, . . . , p_n}"*, The new set of paths *"$p^*{}_n$ = {$p^*{}_1$, $p^*{}_2$, . . . , $p^*{}_n$}"* is obtained by the module collision avoidance criterion. The objective function is:

$$L=\sum\nolimits_{i=1}^{n} length\left(p_i^*\right)=l\cdot\tau \tag{9}$$

where: "L" is the length of the moving path of n functional modules; "τ" is the number of grids; "l" is the length of a grid.

When multiple modules are used for path planning in the reconfiguration platform, the platform defaults all modules that are not the priority to obstacles, thus ensuring the accuracy of the robot's confirmation order. The function module searches the surrounding grid according to the forward direction during the path planning and defines the next ready-to-move grid location as the waiting area. When two modules explore the same waiting area, the distance between modules should be greater than the minimum collision distance. Three module collision avoidance forms are defined according to the relative positions of modules in the reconfigured platform.

The first type is that after two modules search the same waiting area, the next movement direction between modules is opposite and at an angle of 0° to the original movement direction, as shown in Figure 5a–c, ("S_1" represents the higher priority module's path and "S_2" represents the low priority module's motion path.) "S_1" moves to the waiting area and waits one raster movement time, "S_2" moves to the blank raster position and then returns to the original position after "S_1" passes and continues in the original direction. If the situation in Figure 5d is encountered, "S_2" moves one grid in the opposite direction of the original movement, and searches whether there are obstacles above and below the grid. If there is no obstacle, it follows the first type of collision avoidance form; if there is still an obstacle, it continues to search the next position grid, and "S_1" still follows the original route movement.

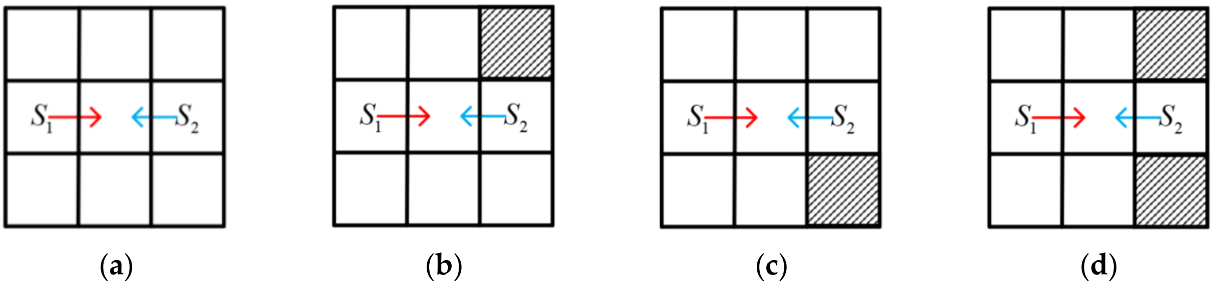


Figure 5. Schematic diagram of the first type of collision avoidance. (**a**) Situation 1; (**b**) Situation 2; (**c**) Situation 3; (**d**) Situation 4.

The second form is that two modules search to the same waiting area, and the subsequent movement direction between modules is opposite, and 90° from the original movement direction or the module movement direction remains the same, and the movement angle between modules is 90°. As shown in Figure 6, "S_2" waits for one grid distance at the previous grid position in the waiting area and continues to move after "S_1" passes through the waiting area.

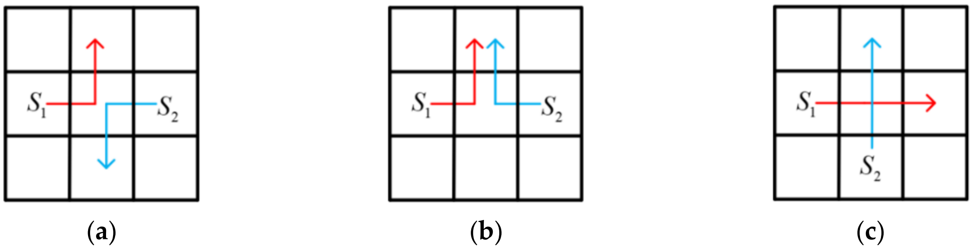


Figure 6. Schematic diagram of the second type of collision avoidance. (**a**) Situation 1; (**b**) Situation 2; (**c**) Situation 3.

The third type is that one of the two modules moves at 90° to the original direction or both modules move at 90° to the original direction after passing through the waiting area. As shown in Figure 7, after "S_1" enters the waiting area, "S_2" moves one grid in the opposite direction of the original motion or moves to a blank grid around the waiting area, and "S_2" continues to move in the original direction after "S_1" finishes moving the grid.

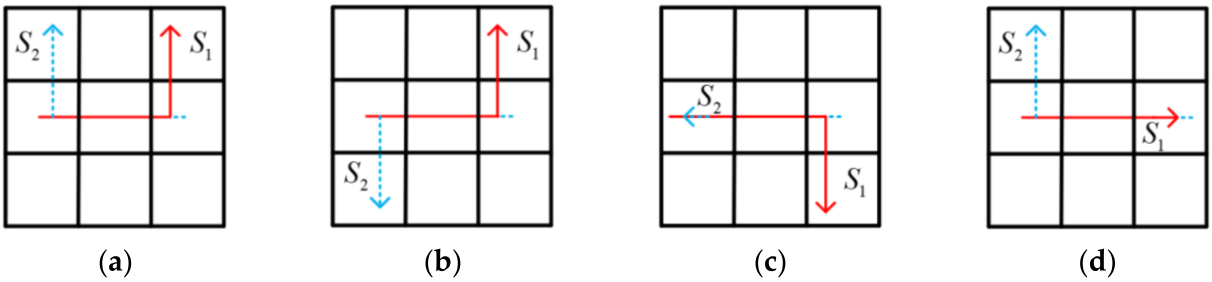


Figure 7. Schematic diagram of the third type of collision avoidance. (**a**) Situation 1; (**b**) Situation 2; (**c**) Situation 3; (**d**) Situation 4.

3.3. Modular Path Planning Based on Optimal Ant Colony Optimization

3.3.1. Ant Colony Optimization

The core idea of ant colony optimization is to simulate the working mechanism of a single ant colony foraging process to carry food to the anthill through mutual collaboration and information transfer, which is a kind of bionic intelligence algorithm. ACO belongs to a heuristic search, which can significantly reduce the difficulty of the problems and improve search efficiency. It also has the characteristics of distributed computing and positive feedback, which can achieve global search optimization [31].

When the traditional ant colony optimization is used for path planning, the key to its implementation mainly includes the calculation of transfer probability, updating of pheromone concentration, storing the taboo table of nodes that ants are not allowed to choose, etc. [32,33].

1. Probability of state transfer

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}\ [\eta_{ij}(t)]^{\beta}}{\sum_{s\in A}[\tau_{is}(t)]^{\alpha}\ [\eta_{is}(t)]^{\beta}}, & if\ j \in A \\ 0,\ otherwise \end{cases} \tag{10}$$

where “$p_{ij}^{k}(t)$” is the transfer probability of the ant “k” from node “i” to node “j” at time “t”. The larger the value, the greater the probability that the ant chooses path “$I \to j$”; “$\tau_{ij}(t)$” is the pheromone concentration of the path at a time “t”; “$\eta_{ij}(t)$” is the heuristic function; “α” and “β” are the pheromone and the desired heuristic factor, respectively; “$\tau_{is}(t)$” is the total amount of information from node “i” to target node “s”; “$\eta_{is}(t)$” is the value of the desired heuristic function from node “i” to target node “s”; “A” is the set of nodes that the ant “k” is allowed to select the next one.

For conventional path planning with the goal of distance minimization, the heuristic function can be described as the inverse of the distance of neighboring nodes.

2. Pheromone concentration

The ant colony algorithm achieves the selection of the shortest path based on a positive feedback mechanism. As time passes, the concentration of pheromone left on the shortest path gradually increases, and the probability of that path being selected increases. However, if the pheromone concentration is too high, the heuristic information will be lost and thus fall into a local optimum. Therefore, the algorithm simulates the mechanism of pheromone volatilization in nature and achieves the pheromone update.

3. Forbidden table

The forbidden table is a record to store the nodes that have been selected by ants during the search iteration of the ant colony algorithm, characterizing the ants are not allowed to select the node again in the subsequent path selection.

3.3.2. Optimal ant Colony Algorithm Design

For the shortcomings of the traditional ant colony algorithm such as slow convergence speed and fall into local optimum easily, in this paper, an improved ant colony algorithm is proposed. First, a non-uniform allocation of initial pheromones based on the traditional ant colony algorithm is adopted to avoid useless search behaviors by ants; Secondly, introducing heuristic factors for node angles to construct multivariate heuristic functions to speed up the search; Then, the deadlock phenomenon is solved by improving the transfer probability; Finally, the avoidance strategy is added to the pre-search process, and the ant backoff strategy is used to deal with the deadlock problem. The optimized ant colony algorithm has better guidance for the reconstruction planning, which can effectively guide the functional module to move along the direction closest to the shortest distance between the initial point and the target point and effectively improve the path planning efficiency.

1. Set non-uniformly distributed initial pheromones

Since the initial pheromone values are the same at the beginning of the ant colony algorithm, the ants select the next node by the shortest distance in the search path, ignoring the global information, which makes the search less efficient and increases the path length. To solve this problem, this paper delineates a rectangular region with the start and endpoints as diagonal vertices and increases the initial value of pheromone in this region. It makes the robotic module have the initiation to move forward toward the target point in the early movement process, which is helpful to reduce the blind search of ants and improve the convergence speed.

2. Design of the heuristic function

The heuristic function in the traditional ant colony algorithm usually only takes the inverse of the distance of neighboring nodes, which leads to the tendency of ants to select the nearest optional node to the current node when making path selection. It leads to deadlock and duplicate paths when ants search for and cannot find an optimal route. So we choose to introduce the heuristic factor of node angle to construct a multivariate heuristic function. In the raster environment, the initial point (x_0, y_0) is connected to the current point (x_i, y_i) to form a line "l_i", and the current point (x_i, y_i) is connected to the target point (x_{i+1}, y_{i+1}) to form a line "l_j". The new heuristic function is designed according to the angle "θ" between "l_i "and "l_j" as well as the distance between the current node and the target node.

(1) Distance heuristic factor

Based on the distance characteristics between two nodes in the two-dimensional space, giving an initial guiding direction for the ant colony algorithm, the attractiveness of the target point for path point selection is used to improve the fast convergence of the algorithm. A distance heuristic formula is established:

$$Q(x_i, y_i) = \frac{1}{1 + \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}} \tag{11}$$

(2) Target pinch angle heuristic factor

Set the angle between the ant's starting point "P_s", current point "P_w" and target point "P_E" as "θ_E", and the ant will find the shortest path when it moves in the shortest straight-line direction. The specific definition of the angle "θ_E" is as follows:

$$\theta_E = \arccos\left(\frac{\overrightarrow{P_s P_E} \cdot \overrightarrow{P_w P_E}}{|P_s P_E| \times |P_w P_E|} \right) \tag{12}$$

where: "θ_E" is the target angle between the vector "$\overrightarrow{P_s P_E}$" and the vector "$\overrightarrow{P_w P_E}$".

The nodes are selected in the explorable area so that "θ_E" tends to the target angle "φ_E", which is conducive to the path search direction tending to the target point "P_E", as shown in Figure 8.

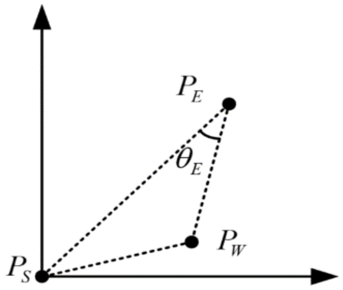


Figure 8. Schematic diagram of target angle.

The target pinch angle "φ_E":

$$\varphi_E = \theta_E + \varepsilon \,,\ \varepsilon > 0 \tag{13}$$

where: "ε" is the angle adjustment parameter greater than zero, which aims to prevent the "φ_E" from being zero; "θ_E" is the angle between the two vectors.

Target pinch angle heuristic factor:

$$R(x_i, y_i) = \frac{1}{\varphi_E + \xi} \,,\ \xi \in [0,\ 1] \tag{14}$$

where: "ξ" is the angle coefficient; "φ_E" is the target angle.

Since the initial point is far away from the target point at the beginning of the path planning, so the "ξ" is increased in the early stage to avoid the blind search due to the large range of the pinch angle; the "ξ" is decreased in the later stage of the path planning to enhance the accuracy of the path planning.

Combining the Equation (10) and the Equation (13) to construct the multivariate heuristic function:

$$H(x, y) = Q(x_i, y_i) \cdot R(x_i, y_i) \tag{15}$$

3. Update pheromone

The optimal solution is obtained by using the stronger global search capability and search speed of the gravitational search algorithm for path search of the whole region, and the obtained results are transformed into the initial pheromone in the ant colony algorithm. This effectively solves the problems caused by the uniform distribution of initial node pheromones in the traditional ant colony algorithm, which easily falls into local optimum and blind search, and also effectively accelerates the speed of path search. When the ant "k" completes a search, the pheromone content of the node on the better path in this iteration is selected for global update. Before the ant "k" shifts from the current point to the next point "$P_i(x_i, y_i)$", a real-time pheromone update is performed for the current point:

$$\tau_{i,j} = \tau_{i,j}^{G} + (1-\rho)\tau_{i,j} \tag{16}$$

$$\rho = \rho_0 + (1-\rho_0)\frac{N}{N_{max}} \tag{17}$$

where: "$\tau_{i,j}$" is the pheromone content of the current node "P_i"; "$\tau_{i,j}^{G}$" is the initial value of the pheromone content of each node, which takes the value of 0.5; "ρ" is the pheromone content decay coefficient; "ρ_0" indicates the initial value of the pheromone decay coefficient, which takes the value of 0.3; "N" is the current number of searches of the ant colony algorithm; “ $N_{\max}$” indicates the maximum number of searches of the ant colony algorithm, which takes the value of 100.

4. Deadlock avoidance strategies

When an ant faces a U-shaped obstacle, it falls into a “trap”, where there is no way around it and it cannot get out of the trap. This phenomenon causes the ants to die and eventually leads the algorithm to deadlock. To avoid getting into the deadlock region, which leads to path search stagnation, a new obstacle avoidance strategy needs to be designed to avoid getting into a draw. As shown in Figure 9, when the ant “k” moves to grid “r” and gets into a deadlock, the ant returns to grid “r’ “at step” $k-1$” and defines grid “r” as an obstacle, and removes grid “r” from the set of optional nodes and uses roulette wheel method to make the ant reselect the step “k” movement to “r'”. This obstacle avoidance strategy can effectively improve the algorithm convergence performance and reduce the search time in reconfiguration path planning.

5. Algorithmic steps

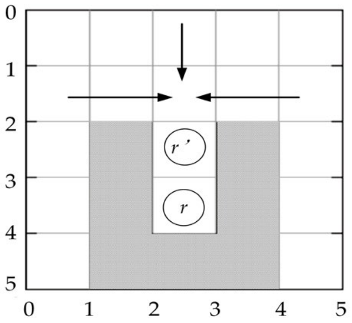


Figure 9. Raster map of U-shaped obstacles.

The steps of the optimized path planning algorithm are shown in Figure 10.

- *Step 1:* Parameter initialization: set $\alpha = 1$, $K = 50$, $\beta = 3$, $\mu = 0.5$;
- *Step 2:* Create the map environment and define the start and endpoints;
- *Step 3:* Set the number of cycles: $Nc = Nc + 1$;
- *Step 4:* Initialize the number of ants: $K = 1$;
- *Step 5:* Calculate the transfer probability of the ant ”k”, and select the next moveable grid point “j” of the ant using the roulette wheel method;
- *Step 6:* Calculate the next selectable grid point “n” of grid “j”;
- *Step 7:* Determine “$n = 1$?”. If yes, define raster point “j” as an obstacle raster and update the map obstacle and return to step 5; otherwise, it will determine if the deadlock condition can be reached;
- *Step 8:* Determine whether the deadlock condition is satisfied? If it satisfies, returning to the previous step “$r - 1$” and defining this step “r” as an obstacle, subsequently updating the map obstacle and returning to step 5; if it is not satisfied, determining whether ant “K ($K = 1, 2, \ldots, k$)” reaches the final target point;
- *Step 9:* Determine whether the ant reaches the endpoint? If the final target point is reached, recording the path length, otherwise returning to step 5;
- *Step 10*: Set the number of ants: “$K = k + 1$”;
- *Step 11:* Update the pheromone;
- *Step 12*: Determine “$N_c > N_{max}$?” if yes, choose the best path; otherwise returning to step 3;
- *Step 13*: Determine whether the path generates collision? If yes, output the path after dealing with obstacles; otherwise, outputting the path directly;
- *Step 14:* Finish.

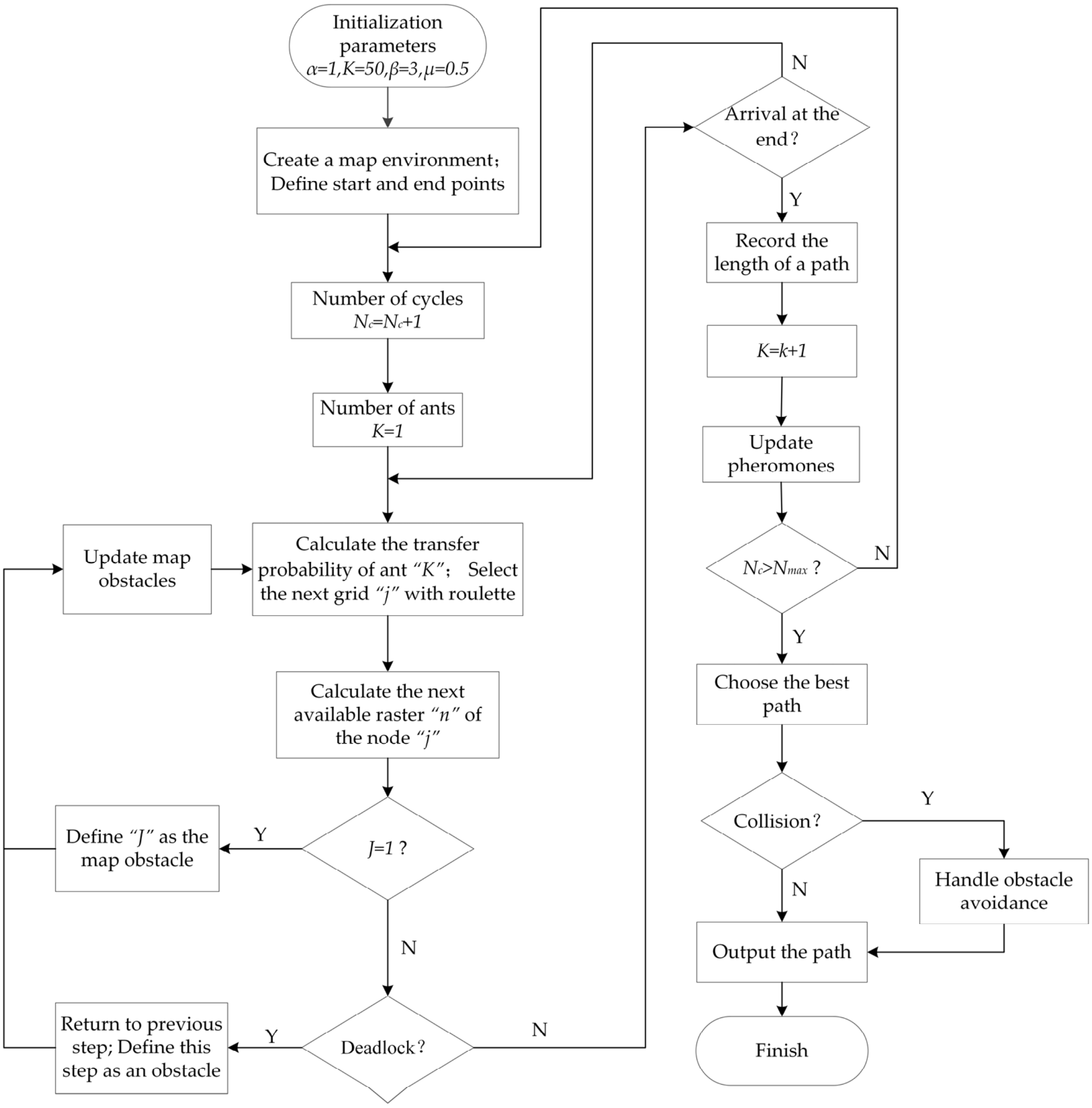

Initialization parameters
α=1,K=50,β=3,μ=0.5
Create a map environment; Define start and end points
Number of cycles N_c=N_c+1
Number of ants K=1
Calculate the transfer probability of ant "K"; Select the next grid "j" with roulette
Update map obstacles
Calculate the next available raster "n" of the node "j"
J=1 ?
Define "J" as the map obstacle
Deadlock?
Return to previous step; Define this step as an obstacle
Arrival at the end?
Record the length of a path
K=k+1
Update pheromones
N_c>N_{max} ?
Choose the best path
Collision?
Handle obstacle avoidance
Output the path
Finish
Y
N

Figure 10. Flow chart of the optimization algorithm.

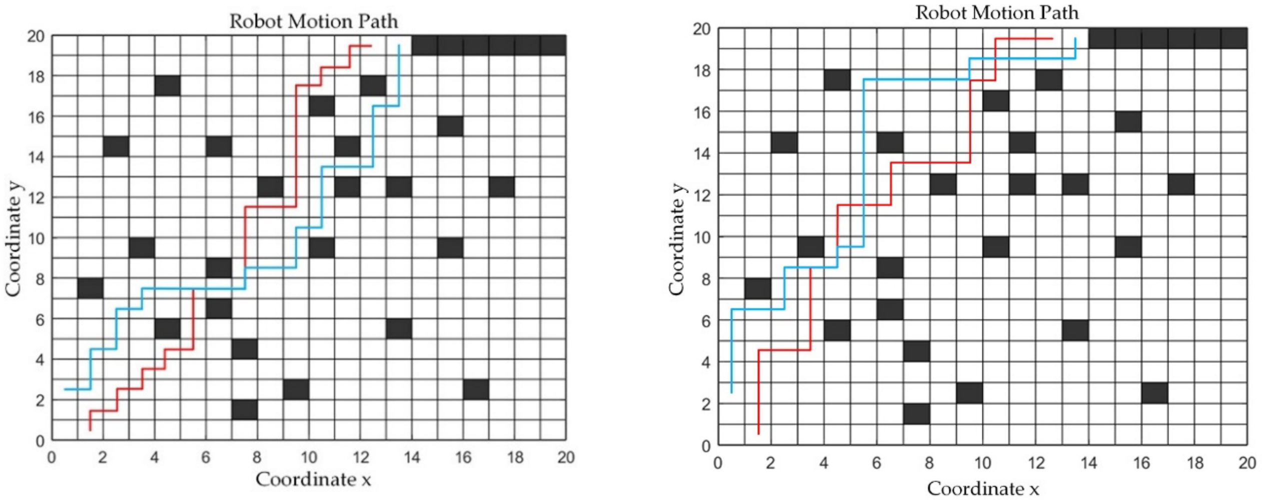
4. Simulation and Analysis of Path Planning for Module's Reconfiguration

4.1. Reconfiguration Path Simulation Experiment

MATLAB models the reconfiguration platform environment and path planning simulation to verify the algorithm's effectiveness. Using the raster method for environment modeling, the algorithm parameters are set as shown in Table 3. The cooperative path planning of the clamping module and the interstitial module is an example in the simulation. The initial position of the clamping module is "T_1" (2,1) and the target point is "E_1" (13,20); the initial position of the interstitial module is "T_2" (1,3) and the target point is "E_2" (14,20). The raster method is used for environment modeling, and the basic ACO path planning method and the improved ACO path planning method proposed in this paper are compared in two experimental environments, respectively. The simulation results are shown in Figures 11–14. The solid red line in Figures represents the moving path "L_1^*" for the clamping function module, and the dashed blue line represents the moving path "L_2^*" for the interstitial function module.

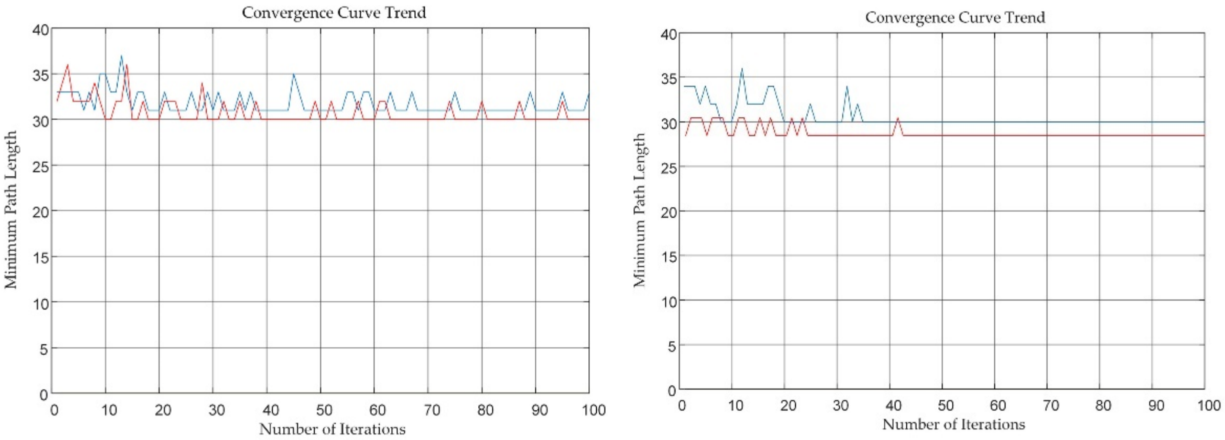
Table 3. Comparison of algorithm parameter settings.

Parameters	Traditional Ant Colony Optimization	Optimized Ant Colony Optimization
Ant population	50	50
Iteration number	100	100
Pheromone inspired factor	1	1
Expectation heuristic factor	3	3
Pheromone evaporation factor	ρ_0	$\rho_0+(1-\rho_0)\frac{N}{N_{\max}}$
Distance heuristic factor	$\frac{1}{d_{ij}}$	$\frac{1}{1+d_{ij}}$
Nodal perspective heuristic factor	/	$\frac{1}{\varphi_E+\xi}$
Initial pheromone	Uniform distribution	Non-uniform distribution
Deadlock handling	/	Fallback strategy



(**a**) (**b**)

Figure 11. Schematic diagram of the ant colony algorithm robot movement path (Environment 1). (**a**) Ant colony algorithm (**b**) Improved ant colony algorithm.



(**a**) (**b**)

Figure 12. Schematic diagram of the convergence curve change of the ant colony algorithm (Environment 1). (**a**) Ant colony algorithm (**b**) Improved ant colony algorithm.

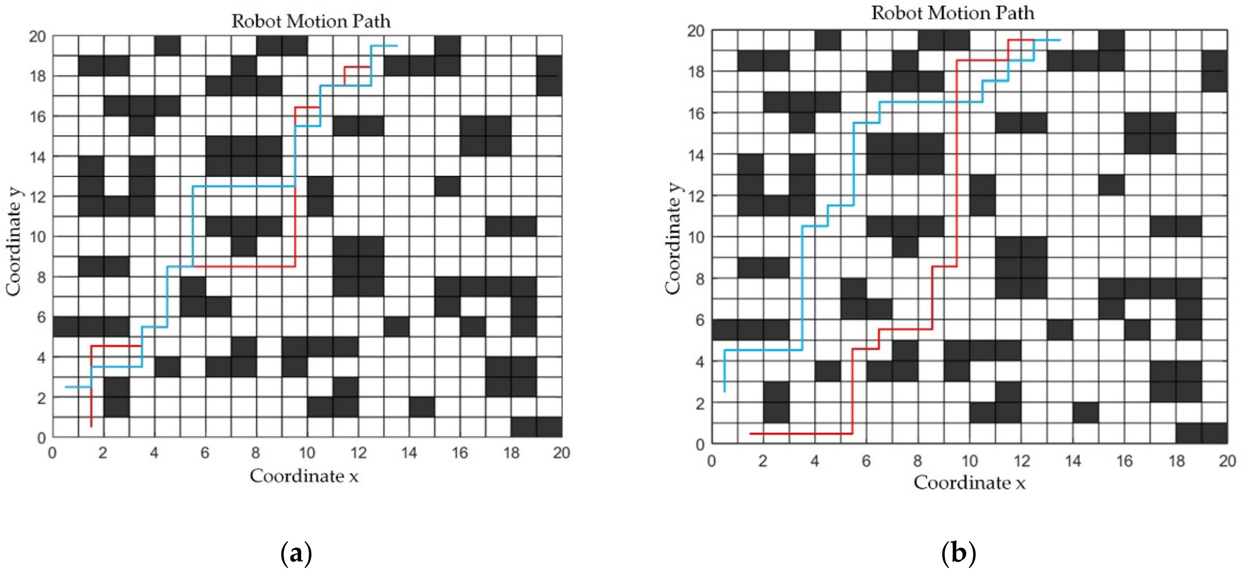


Figure 13. Schematic diagram of the ant colony algorithm robot movement path (Environment 2). (**a**) Ant colony algorithm (**b**) Improved ant colony algorithm.

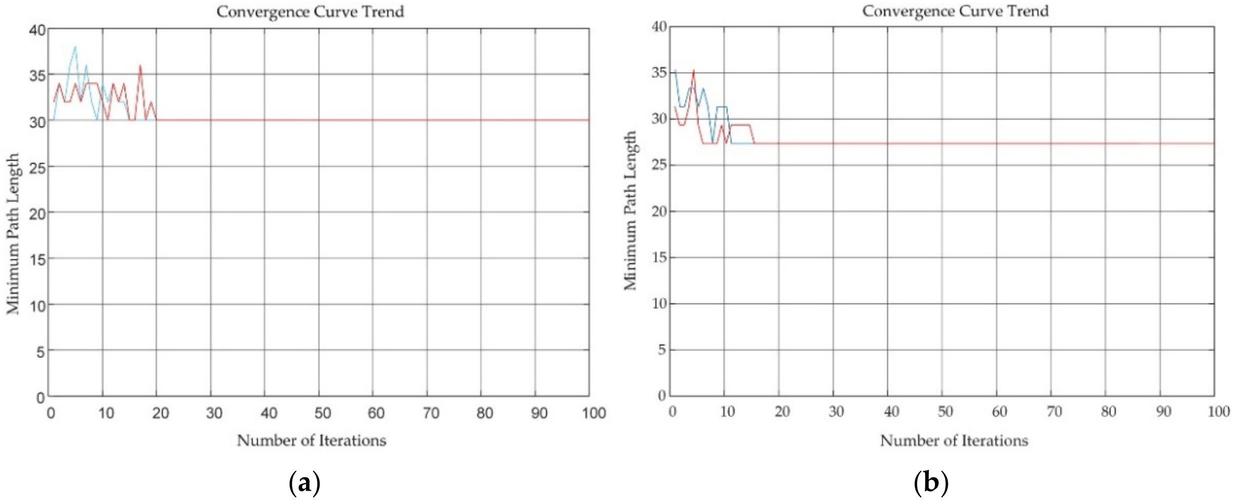


Figure 14. Schematic diagram of the convergence curve change of the ant colony algorithm (Environment 2). (**a**) Ant colony algorithm (**b**) Improved ant colony algorithm.

In the simulation results under the 20 × 20 raster environment, the path planning graphs and iterative convergence curves of the basic ant colony algorithm and the improved ant colony algorithm proposed in this paper are shown respectively. From Figure 12, it can be concluded that in the set range of 100 steps, the traditional ant colony algorithm is difficult to converge the algorithm and there are still small fluctuations. The simulation results of the optimized algorithm show that the two types of functional modules converge at about 42 steps and 35 steps, respectively. The convergence efficiency has been increased and improved substantially. Moreover, the minimum path length is reduced by about 6.7%. The simulation results in experiment 2 are shown in Figures 13 and 14. The optimized simulation structure shows that the convergence efficiency of the two modules is improved by 12.5% and 26.7% respectively. The path length is also reduced by about 8.3%. In addition, it can be seen from the simulation path diagrams that the optimized path inflection point has been significantly reduced. In summary, the path planning method proposed in this paper has an obvious improvement effect and better search performance.

4.2. Collaborative Collision Analysis of Modules

Take the simulation results of the improved ant colony algorithm under environment 1 as an example. As shown in Figure 15, four collision avoidance behaviors occur in the path during the collaborative path planning of the module. The first time the two modules have the third type of collision avoidance behavior, the direction of the clamping module is unchanged and the angle of motion is 90°, the two modules pass the waiting area and move in the same direction respectively, the clamping module represented by the moving path "L_1^*" moves to the waiting area (4,9), the interstitial module represented by the moving path "L_2^*" retreats to the grid (3,9), and continues to move in the original direction after the clamping module passes. The second time the second type of collision avoidance behavior occurs, the direction of motion of the module remains unchanged and the angle of motion is 90°, the interstitial module represented by the moving path "L_2^*" waits for a grid distance at the position of the grid (6,11), and the interstitial module continues to move after the clamping module passes the waiting area (6,12). The third and fourth time the second type of collision avoidance behavior occurs again, the interstitial module represented by the movement path "L_2^*" waits for a grid distance at the position of the grid (9,18) and (10,19), after the clamping module passes the waiting area (10,18), (11,19) the interstitial functional module continues to move.

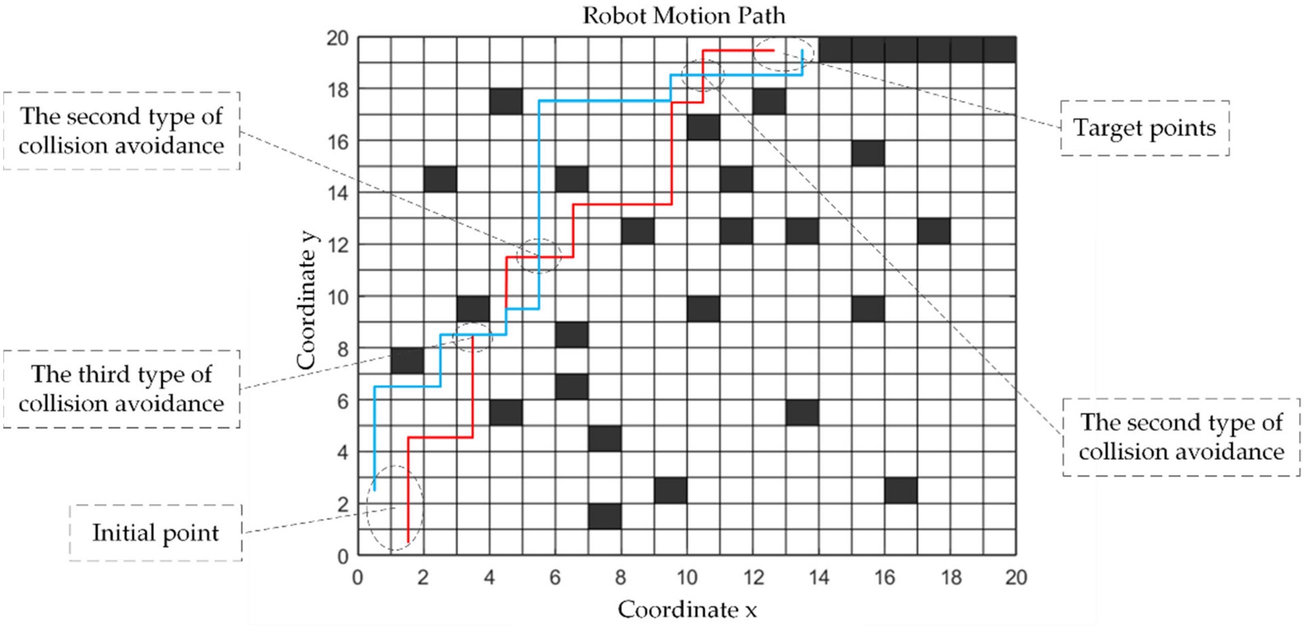

Robot Motion Path
The second type of collision avoidance
The third type of collision avoidance
Initial point
Target points
The second type of collision avoidance
Coordinate y
Coordinate x
0
2
4
6
8
10
12
14
16
18
20

Figure 15. Module collaborative path planning diagram.

When a group of modules completes the reconstruction path planning, the next priority module of the robot performs the inter-module collaborative path planning until the overall reconstruction tasks of the robot are completed. If there are remaining modules in the platform after completing the reconstruction tasks, the modules stop moving and wait for the following reconstruction tasks. Finally, the two modules work together to reach the reconfiguration target location and always maintain the collision-free optimal reconfiguration path.

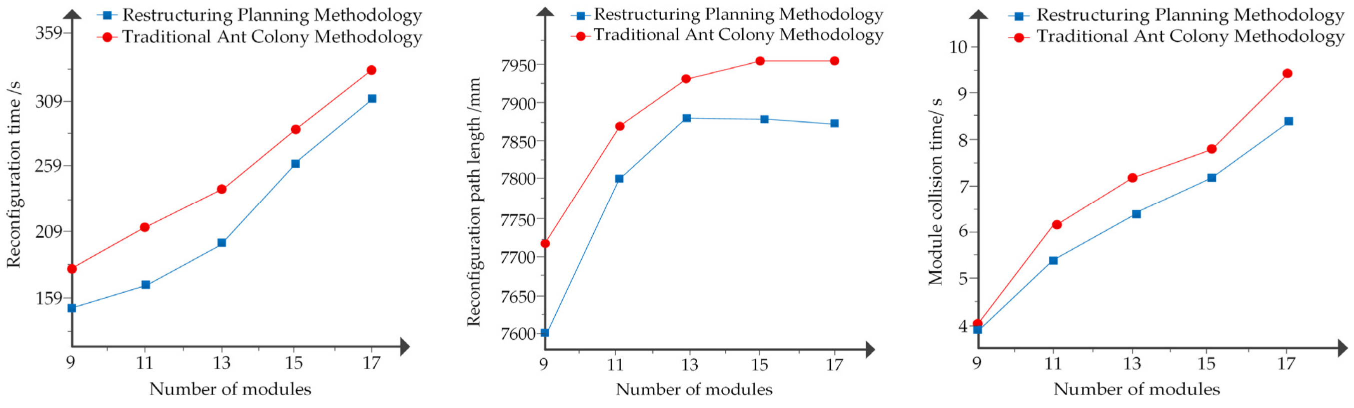
4.3. Module Reconfiguration Platform Planning Efficiency Analysis

The planning time of the module plays an essential role in the reconfiguration path planning, and the path planning time is the sum of the module's avoidance consumption time and the module's moving path consumption time. In Section 2.2, the configuration design of the robot with different DOF is carried out, and the module parameters corresponding to different configurations are shown in Table 4.

Table 4. Robotic configuration module parameters.

DOF	Three	Four	Five	Six	Seven
Total Modules	9	11	13	15	17
Motion Modules	5	6	7	8	9
Interstitial Modules	4	5	6	7	8

Figure 16a shows the actual time taken by the reconfigured robot with different DOF to complete the target configuration using different path planning methods in the reconfiguration platform. In terms of the overall time consumed by the robot, the overall time consumed increases with the number of modules. Compared with the traditional ant colony method, the reconfiguration planning algorithm can effectively reduce the overall reconfiguration planning time of the modules. However, when the number of modules reaches a certain level, the robot does not significantly reduce the overall modules' reconfiguration planning time. As the number of modules increases, the optional free area in the reconfiguration platform decreases accordingly. Therefore, the number of modules within the platform needs to be adjusted according to the specifications of the module reconfiguration platform to achieve the most efficient utilization of the reconfiguration platform.



(**a**) (**b**) (**c**)

Figure 16. Module collaborative path planning diagram. (**a**) Robot configuration reconstruction planning time, (**b**) Robot configuration reconstruction plan path length, (**c**) Collision avoidance time of robot configuration module.

Figure 16b shows the length of the reconfiguration path for the modules to complete the target configuration within the reconfiguration platform. The time required for the robot to complete all module reconfigurations increases with the number of modules. Compared with the traditional ant colony method, the module reconfiguration planning method effectively reduces the path length.

Figure 16c shows the required collision avoidance time when the module completes the target configurations. As the number of modules increases, the collision avoidance time required by the robot also increases. Compared with the traditional ant colony path planning method, the module reconfiguration planning method reduces the collision avoidance time when the module completes the reconfiguration planning.

Combined with the above simulation results, it can be seen that due to the priority definition of the module reconfiguration planning and the influence of heuristic functions, the unnecessary moving paths can be effectively reduced, and the path planning efficiency can be improved. The reconfiguration path planning method proposed in this paper finds the better reconfiguration path in the module reconfiguration platform with a relatively short time under different DOF configuration requirements. It executes the module collision avoidance criterion to complete the path planning between reconfigured modules in the shortest time.

5. Conclusions

The reconfiguring modular robot can reconfigure and combine modules according to different task requirements, and the robot is highly versatile and expandable. Self-reconfiguring robots are gaining more and more attention and applications due to their unique advantages, especially in the field of automotive manufacturing. Modular robots are applied to work properly in harsh environments, such as vacuum welding, high-temperature heat treatment, forging, and stamping, etc. The application of modular robots effectively reduces human involvement and fully ensures work safety. It is an important research direction for the future industrial field. The study of the reconfiguring robot is vital technical support for future intelligent manufacturing. In this paper, the module and configuration scheme design, robotic kinematics analysis, and module path planning in the reconfiguration platform of the reconfiguring robot are studied in depth.

(1) In this paper, the configuration design of different reconfigured robotic configurations is based on the multilevel theory of mechanical modules. A kinematic analysis based on a fully functional robot is performed to establish the D-H parameters of the robot, which lays the mathematical foundation for the later paper.
(2) We perform the construction and mathematical model description of the module reconfiguration platform and define the movement priority and collision avoidance guidelines for modules in the reconfiguration platform. An innovative path planning method based on the module reconfiguration platform is proposed to ensure that the robot can reconfigure its configuration according to the requirements of different operations.
(3) Simulation verification and analysis of the proposed module's reconfiguration path planning method in this paper. MATLAB simulates the cooperative path planning of two different functional modules to verify the rationality of the obstacle avoidance criterion and collision-free path planning. In addition, we analyze the collaborative planning efficiency of the reconfigured platform based on the configurations with different DOF and finally verify the advantages of the module reconfiguration planning method over the traditional ant colony algorithms through various aspects.

In this paper, the research on reconfiguring robots focuses on the module's structure and configuration scheme design, path planning method, etc. At this stage, the following aspects can be further investigated for reconfiguring robots.

1. The design of a robotic configuration scheme can also fully consider the advantages of parallel mechanisms or series-parallel hybrid mechanisms to improve the flexibility of the robotic configuration.
2. This article considers the reconfiguration path planning research in the two-dimensional plane, and additionally, the obstacle avoidance and path planning research in the three-dimensional environment should also be considered.
3. Research the module's control strategy and integrate multiple algorithms to fully use the advantages of different algorithms and effectively improve reconfiguration efficiency.

Author Contributions: Conceptualization, Y.D. and C.-F.X.; methodology, Y.D. and Z.-X.L.; software, Z.-X.L.; validation, Y.D. and Z.-L.L.; investigation, C.-F.X., W.-Y.Q. and Q.-H.Z.; writing—original draft preparation, C.-F.X.; writing—review and editing, Y.D. and Z.-L.L. All authors have read and agreed to the published version of the manuscript.

Funding: The study is supported by the Natural Science Foundation of China (Grant No. 52075134); Natural Science Foundation of Heilongjiang Province of China (Grant No. LH2019E062); Postdoctoral Scientific Research Developmental Fund of Heilongjiang Province of China (Grant No. LBH-Q20097); Fundamental Research Foundation for Universities of Heilongjiang Province, China (LGYC2018JC040).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Date sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, D.Y. Research on Reconfigurable Modular Robot Docking and Combination Strategy. Master's Thesis, Shandong University, Jinan, China, 2020. (In Chinese)
2. Gao, X.W. An analysis of industrial robots in intelligent manufacturing. *Intern. Combust. Engine Parts* **2021**, *20*, 199–200. [CrossRef]
3. Christensen, D.J.; Schultz, U.P.; Stoy, K. A distributed and morphology-independent strategy for adaptive locomotion in self-reconfigurable modular robots. *Robot. Auton. Syst.* **2013**, *61*, 1021–1035. [CrossRef]
4. Chu, K.D.; Hossain, S.G.M.; Nelson, C.A. Design of a four-DOF modular self-reconfigurable robot with novel gaits. In Proceedings of the ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Washington, DC, USA, 28–31 August 2011; pp. 1–8.
5. Gao, S.Q.; Guo, S.G.; Xing, D.D. Research on self-reconfiguring modular robot. *Comput. CD Softw. Appl.* **2013**, *1*, 63–64.
6. Fukuda, T. Approach to the Dynamically Reconfigurable Robotic System. *J. Intell. Robot. Syst.* **1988**, *1*, 55–72. [CrossRef]
7. Castano, A.; Behar, A.; Will, P.M. The Conro modules for reconfigurable robots. *IEEE/ASME Trans. Mechatron.* **2002**, *7*, 403–409. [CrossRef]
8. Rubenstein, M.; Payne, K.; Will, P. Docking among independent and autonomous CONRO self-reconfigurable robots. In Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans, LA, USA, 26 April–1 May 2004; pp. 2877–2882.
9. Acaccia, G.; Bruzzone, L.; Razzoli, R. A modular robotic system for industrial applications. *Assem. Autom.* **2008**, *28*, 151–162. [CrossRef]
10. Zhang, S.C.; Pu, J.X.; Si, Y.N.; Sun, L. Path planning for mobile robot using an enhanced ant colony optimization and path geometric optimization. *Int. J. Adv. Robot. Syst.* **2021**, *18*, 3. [CrossRef]
11. Guo, P.; Zhang, J.W.; Rong, F. Application of ant colony algorithm in unmanned park tour route. *Mod. Electron. Tech.* **2019**, *42*, 149–152.
12. Zhang, X.L.; Yang, Y.X.; Xie, Y.C. An improved ant colony algorithm for robot path planning. *Comput. Eng. Appl.* **2020**, *56*, 29–34.
13. Li, Y.; Ji, J.N.; Shen, J.L.; Su, R. A path planning method for mobile robots based on improved ant colony algorithm. *J. Nanjing Univ. Inf. Sci. Technol.* **2021**, *13*, 298–303. [CrossRef]
14. Jiang, M.; Wang, F.; Ge, Y.; Sun, L.L. Research on path planning of mobile robot based on improved ant colony algorithm. *Chin. J. Sci. Inst.* **2019**, *40*, 113–122. [CrossRef]
15. Deng, H.Y. Comparative Study of Ant Colony Algorithm, Genetic Algorithm and the Fusion of Both in TSP Application. Master's Thesis, Shanxi Normal University, Xi'an, China, 2017. (In Chinese)
16. Tao, Y.; Gao, H.; Ren, F.; Chen, C.Y.; Wang, T.M.; Xiong, H.G.; Jiang, S. A Mobile Service Robot Global Path Planning Method Based on Ant Colony Optimization and Fuzzy Control. *Appl. Sci.* **2021**, *11*, 3605. [CrossRef]
17. Niu, L.H.; Ji, N.P. Research on robot path planning problem based on improved ant colony algorithm. *Microprocessors* **2020**, *41*, 37–40.
18. Li, S.D.; You, X.M.; Liu, S. A dual ant-state ant colony algorithm combining the dynamic grading of ABC algorithm. *Comput. Eng. Appl.* **2020**, *56*, 37–46.
19. Anh, V.L.; Veerajagadheswar, P.; Vinu, S. Modified A-Star algorithm for efficient coverage path planning in tetris inspired self-reconfigurable robot with integrated laser sensor. *Sensors* **2018**, *18*, 2585. [CrossRef]
20. Sun, J.B.; Liu, G.L.; Tian, G.H. Smart obstacle avoidance using a danger index for a dynamic environment. *Appl. Sci.* **2019**, *9*, 1589. [CrossRef]
21. Liang, J.H.; Lee, C.H. Efficient collision-free path planning of multiple mobile robots system using efficient artificial bee colony algorithm. *Adv. Eng. Softw.* **2015**, *79*, 47–56. [CrossRef]
22. Qi, R.L.; Zhou, W.J.; Liu, J.G.; Zhang, W.; Lei, X. Obstacle avoidance trajectory planning for gaussian motion of robot based on probability theory. *J. Mech. Eng.* **2017**, *53*, 92–100. [CrossRef]
23. Su, J.B.; Xie, W.L. Motion planning and coordination for robot systems based on representation space, transactions on systems. *IEEE Trans. Syst. Man Cyb.* **2011**, *41*, 248–260. [CrossRef] [PubMed]
24. Zhang, H.B. *Research on the Design and Reconfiguration Strategy of Space Truss in-Orbit Assembly Robot*; Harbin Institute of Technology: Harbin, China, 2020.
25. Liu, K.Y.; Chen, M.; Fei, Y.Q. Reconfiguration strategy for tandem modular robots. *High Tech. Lett.* **2019**, *29*, 995–1002.
26. Lin, J.Z.; Ye, C.C.; Yang, J.X.; Zhao, H.; Ding, H.; Luo, M. Contour error-based optimization of the end-effector pose of a 6 degree-of-freedom serial robot in milling operation. *Robot. Comput.-Integr. Manuf.* **2022**, *73*, 102257. [CrossRef]
27. Ma, L.; Zhao, S.D. Analysis of combined symmetry concept system and application of mechanical structure. *Mod. Manuf. Technol. Equip.* **2018**, *4*, 45–47.

28. Zuo, Z.H. Research on Fast Modeling of Kinematics and Dynamics of Modular Robotic Arm. Master's Thesis, Beijing University of Posts and Telecommunications, Beijing, China, 2015. (In Chinese)
29. Song, W.G. *Fundamentals of Robotics*; Metallurgical Industry Press: Beijing, China, 2015; pp. 61–66.
30. Zhu, Y.; Zhu, L.C.; Long, Y.; Luo, J.Q. The Wolf pack assignment rule based on ant colony algorithm and the path planning of scout ants in complex raster diagra. *J. Phys. Conf. Ser.* **2021**, *1*, 1732. [CrossRef]
31. Li, H.; Zang, L.; Gan, L. Path search for guessing symbolic execution based on ant colony algorithm. *Comput. Sci.* **2018**, *45*, 145–150.
32. Wang, Y. Research on the optimization method of air logistics distribution path based on ant colony algorithm. *Inf. Technol.* **2021**, *11*, 76–80.
33. Liu, Y. Robot path planning method based on improved potential field and ant colony algorithm. *Comput. Simul.* **2021**, *38*, 355–360.