

# Assembly Sequence Planning for Robotic Automatic Operation

HuiLi<sup>1, a</sup>, Xu-tang Zhang<sup>2, b</sup> and Xi Wang<sup>2, c</sup>

<sup>1</sup>School of Mechatronics, Harbin Institute of Technology, Harbin 150001, China;

<sup>2</sup>School of Mechatronics, Harbin Institute of Technology, Harbin 150001, China;

<sup>a</sup>574774369@qq.com, <sup>b</sup>13840234756@163.com, <sup>c</sup> 939425550@qq.com

**Keywords:** assembly sequence planning, interference matrix, discrete particle swarm optimization algorithm, automatic operation.

**Abstract.** This paper has improved the traditional interference matrix, with joining the information of robotic assembly resource, which is used to search for the optimal assembly sequence. We have given a processing method of interference matrix containing the resource of the robotic assembly, which includes the decision method of geometry feasibility, method of judging the interference between two parts. By dealing with the proposed interference matrix, we define a fitness function to evaluate the value of the various assembly sequences. At last, the paper use Discrete Particle Swarm Optimization (DPSO) algorithm to search the optimal assembly sequence, which is more conformable to actually assemble by using robot.

## 1. Introduction

Assembly sequence planning is a hot research topic in recent years, the research method can be divided into 3 kinds: graph theoretic approach[1,2,3], heuristic search algorithm[4,5], planning method based on virtual reality. If we plan an assembly sequence of a complex assembly, the first method will appear the problem of combination explosion. Therefore, in recent years, many scholars have proposed many solutions for solving this problem by using heuristic search algorithms. Most of the research works are focused on the optimization of heuristic assembly sequence planning algorithm[6]. Bonneville F at al. use genetic algorithm to generate and evaluate assembly plan[7]. De Lit P at al. proposed a generated assembly tree, and use the expert system to find the optimal sequence[8]. Hong Yu at al. proposed a review of the assembly sequence planning method based on particle swarm optimization[9]. The most of proposed heuristic search algorithms are depended on the interference matrix. But until now, the assembly sequence planning are realized in the virtual environment without the resource of the tools, this is not realistic.

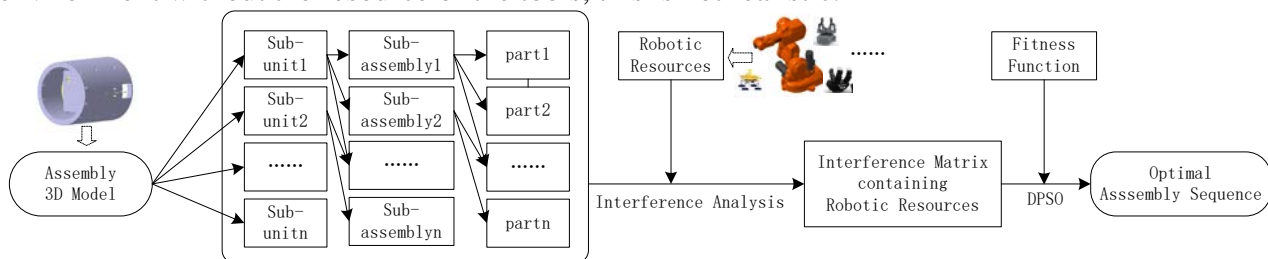


Fig. 1 Framework of the paper

So the The paper has improved the traditional interference matrix in the part 1, with joining the information of robotic assembly resources. Then, the paper has given the processing method of the interference matrix containing robotic resources, constructed fitness function and improved the Discrete Particle Swarm Optimization(DPSO) algorithm to search the optimal assembly sequence, which has high value. In the last, the paper gives an example to verify the rationality of the method proposed by this paper. The framework of the paper is show in figure.1, which explain the integrated thoughts of this paper.

## 2. Interference matrix

Before modeling, we put forward some hypothesis:

(1) Ignoring the micro deformation and accumulated tolerance of the parts in the assembly process;

(2) When assembling the threaded joint, for example: nut, bolt at al. we treat it as the linear assembling.

(3) The component ,sub-assembly in the assembly are defined by person.

### 2.1 Interference matrix of parts

We can analyze the interference relation between parts in the assembly from 6 directions(+X, -X, +Y, -Y, +Z, -Z). Because the interference relation between part i and part j in the +X direction is the same with the interference relation between part j and part i in the -X direction. So the Interference matrix in the -X direction is the transpose of the Interference matrix in the +X direction. Therefore we can only solve the Interference matrix in the +X, +Y, +Z directions.

The element in the interference matrix of parts represent the interference relation between parts in the assembly, when the parts are assembled or disassembled from the assembly.interference matrix of parts ( $I^2M$ ) is described as bellow:

$$I^2M_d = \begin{matrix} & P_1 & P_2 & \dots & P_n \\ \begin{matrix} P_1 \\ P_2 \\ \dots \\ P_n \end{matrix} & \begin{bmatrix} I_{11d} & I_{12d} & \dots & I_{1nd} \\ I_{21d} & I_{22d} & \dots & I_{2nd} \\ \dots & \dots & \dots & \dots \\ I_{n1d} & I_{n2d} & \dots & I_{ndd} \end{bmatrix} \end{matrix} \quad (1)$$

$I^2M_d$ ——interference matrix of parts in the d directions. (d=+X, -X, +Y, -Y, +Z, -Z)

$I_{ijd}$ ——interference relation between part i and part j, which is decided by the bellow

$$I_{ij} = \begin{cases} 0 & \text{interference occurs between part i and part j in d direction} \\ 1 & \text{interference doesn't occurs between part i and part j in d direction} \end{cases} \quad (2)$$

### 2.2 Interference matrix containing robotic resources

The resources of the robotic can be divided into two kinds by the functions of the resources :

(1) the resource for grabbing,  $T(P)$ ; (2) the resource for auxiliary assembly,  $AT(P)$ . The robotic resource of part i in the assembly for grabbing can be defined as:  $T(P_i) = \{t(p_i)_1, t(p_i)_2, \dots, t(p_i)_n\}$ .  $t(p_i)_j$  represent the j th robotic resource of part i for grabbing, n represent the total number of the robotic resource of part i for grabbing. The robotic resource of part i in the assembly for auxiliary assembly can be defined as:  $AT(P_i) = \{at(p_i)_1, at(p_i)_2, \dots, at(p_i)_m\}$ .  $at(p_i)_j$  represent the j th robotic resource of part i for auxiliary assembly, m represent the total number of the robotic resource of part i for auxiliary assembly.

The robotic assembly resource include both the resource for grabbing and the resource for auxiliary assembly, which can be defined as:  $R(P) = \{T(P), AT(P)\}$ ,  $R(P) = \{r(p_1), r(p_2), \dots, r(p_k)\}$ .  $r(p_i)$  represent the all robotic assembly resource of part i, which is the sum of the  $T(P_i)$  and  $AT(P_i)$ . The total number of elements in  $r(p_i)$  is (n+m).

So, the interference matrix containing robotic resources is described as expression (3):

$$I^2MT_d = \begin{matrix} & P & R(P) \\ \begin{matrix} P \\ R(P) \end{matrix} & \begin{bmatrix} I^2M_d & IPR(P)_d \\ IR(P)P_d & IR(P)R(P)_d \end{bmatrix} \end{matrix} \quad (3)$$

$I^2MT_d$ ——the interference matrix containing robotic resources in d direction;

$I^2M_d$ ——interference matrix of parts in d direction;

$IR(P)P_d$ ——interference relationship between parts and robotic assembly resources in d direction;

$IR(P)R(P)_d$ ——interference relationship among the robotic assembly resources in d direction.

Interference relationship between the robotic assembly resources in d direction can be described as expression (4):

$$IR(P)R(P)_d = r(p_1) \begin{bmatrix} r(p_1) & r(p_2) & \dots & r(p_n) \\ IR_{P_1}R_{P_1,d} & IR_{P_1}R_{P_2,d} & \dots & IR_{P_1}R_{P_n,d} \\ r(p_2) & IR_{P_2}R_{P_1,d} & IR_{P_2}R_{P_2,d} & \dots & IR_{P_2}R_{P_n,d} \\ \dots & \dots & \dots & \dots & \dots \\ r(p_n) & IR_{P_n}R_{P_1,d} & IR_{P_n}R_{P_2,d} & \dots & IR_{P_n}R_{P_n,d} \end{bmatrix} \quad (4)$$

The paper gives an example of an actual assembly part in fig.2, to explain the usage guidance of the interference matrix containing robotic resources.

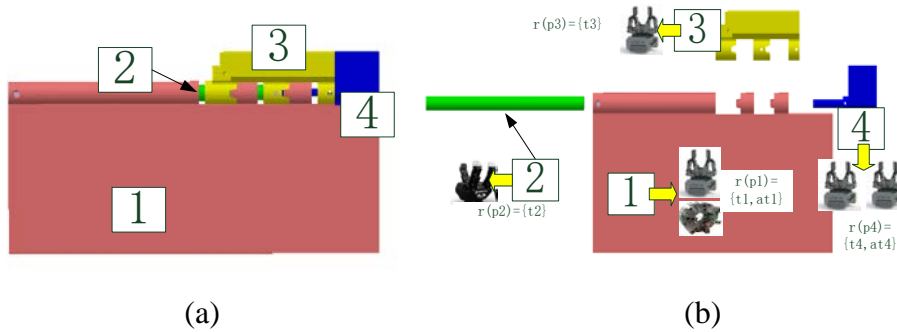


Fig. 2 (a) an actual assembly (b) an explosive view of the actual assembly with the robotic resource. The number of the parts in the actual assembly are shown in the fig 1 (a), we define part 1 as the basal part in the assembly, which is assembled firstly. The robotic resource of parts 1,2,3,4 for grabbing are defined as :  $T(P1)=\{t1\}$ ,  $T(P2)=\{t2\}$ ,  $T(P3)=\{t3\}$ ,  $T(P4)=\{t4\}$ . The parts 1,4 need the resource for auxiliary assembly, which are defined as follows:  $AT(P1)=\{at1\}$ ,  $AT(P4)=\{at4\}$ . The parts 2,3 don't need the resource for auxiliary assembly, which are defined as  $\emptyset$ . The robotic assembly resource  $R(P)$  during assembling process can be described as:

$$\begin{aligned} R(P) &= \{T(P), AT(P)\} \\ &= \{T(P1), T(P2), T(P3), T(P4), AT(P1), AT(P4)\} \\ &= \{t1, t2, t3, t4, at1, at4\} \end{aligned} \quad (5)$$

So, the interference matrix containing robotic resources in +X direction is described as the right in fig. 3 by change the left interference link relationship graph:

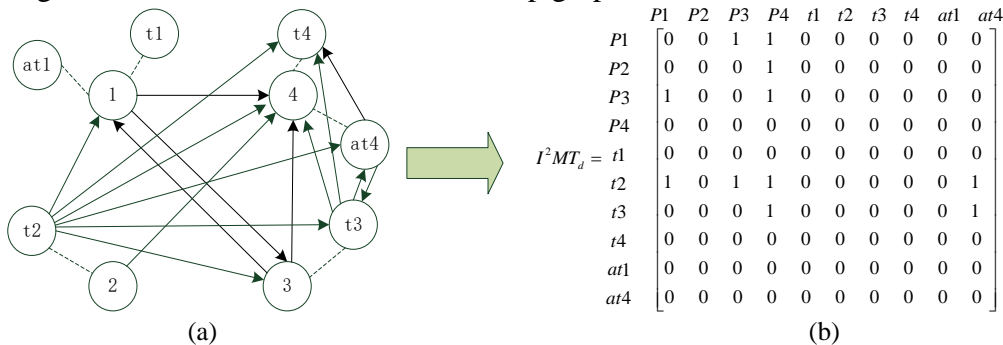


Fig. 3 (a) the actual assembly's interference link relationship graph in +X direction (b) the actual assembly's interference matrix containing robotic resources in +X direction

The interference matrix containing robotic resources in +Y, +Z directions are similar to the one in +X direction.

### 3. Assembly sequence planning

#### 3.1 Inference of geometric feasibility

The interference matrix containing robotic resources' handling method is different from the traditional interference matrix of parts' handling method. When a part is being assembled, the robotic

resource for grabbing must be used. When the part has been assembled, the robotic grab resource must exit from the assembly. However, the robotic resource for auxiliary assembly must exist in the assembly until all the parts connecting to this part have been assembled.

We explain the method of how to interference the parts' geometric feasibility with the example in fig.1. When a parts can be assembled in d direction, it can also be disassembled in the opposite direction., and the sequence of the disassemble is opposite to the sequence of the assemble. So, we use this method to plan the assemble sequence.

When there is no part starting to be disassembled, the robotic resources doesn't have any interference relationship with the parts in the assembly, so the matrix is shown in fig 4(a). When we judge a part whether it can be disassembled from the assembly, for example the part 3, we should do like this: Firstly, judging if the robotic resources for auxiliary assembly would be needed. Obviously, the part1 is defined as the basic part, so when judge whether the part3 can be disassembled or not, the corresponding row and column about at1 in  $I^2MT_d1$  should also be changed to the the corresponding row and column about at1 in  $I^2MT_d$ . The column and row about t4 should also change like it about at1 as well.

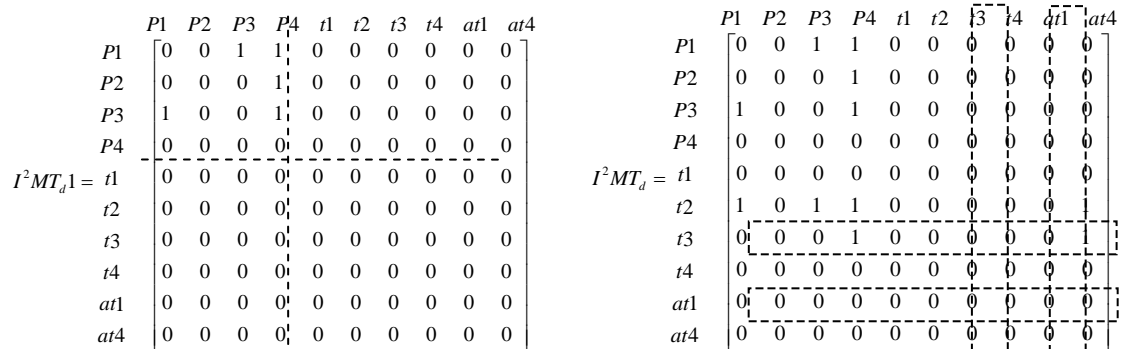


Fig. 4 (a) the interference matrix before disassemble (b) the interference matrix containing robotic resources

Next, we can use the equation (6) to judge whether a part can be disassembled or not. Define  $S(P_i)$  as the sum of elements which represent the interference relationship between a part which would be disassembled in d direction and other parts. If  $S(P_i) = 0$ , the part can be disassembled in d direction. But if  $S(P_i) \neq 0$ , the part can't be disassembled in d direction.

$$S(P_i)_d = S_p(P_i)_d + S_t(P_i)_d \tag{6}$$

$S_p(P_i)_d$  is the sum of the interference relationship between the parts and robotic resources.  $S_t(P_i)_d$  is the sum of the interference relationship among the robotic resources.

$$S_p(P_i)_d = \sum_{j=1}^n I_{ijd} + \sum_{j=n+1}^m IP_iR(P_j)_d \tag{7}$$

$$S_t(P_i)_d = \sum_{j=1}^n IR(P_i)P_{jd} + \sum_{j=1}^n IR(P_i)R(P_j)_d \tag{8}$$

### 3.2 Value of the assembly sequence

#### 3.2.1 Value of the geometric feasibility

Define the  $v_p$  as the value of the geometric feasibility, which represent the number of the parts that can be disassembled in the assembly sequence. Obviously, the greater the  $v_p$ , the better.

#### 3.2.2 Value of stability

Define the  $v_s$  as the value of the stability, which represent the stability of the assembly sequence. Define support matrix as  $S = (s_{ij})_{n \times n}$ , the of elements of it represent the support relations between part i and part j. The rules of evaluate the value of the stability between two parts are shown in the expression (9).

$$s_{ij} = \begin{cases} 2 & \text{there is special link relation between part } i \text{ and part } j \\ 1 & \text{part } j \text{ support part } i \text{ in direction of gravity} \\ 0 & \text{there is no supporting or link relation between part } i \text{ and part } j \end{cases} \quad (9)$$

So, the value of a assembly sequence's stability can be calculate by expression (10). n represent the total number of the parts in the assembly.

$$v_s = \sum_{i=1}^n \sum_{j=1}^i s_{ij} \quad 0 \leq v_s \leq 2n - 2 \quad (10)$$

Obviously, the bigger the  $v_s$ , the better.

### 3.2.3 Other value of the assembly sequence

Define  $v_d$  as the time of the direction changes when the part are disassembled, if the next part's disassemble direction is different from the before,  $v_d = v_d + 1$ . Obviously, the smaller the  $v_d$ , the better.

Define  $v_t$  as the time of the robotic resources changes when the part are disassembled, if the next part's robotic resources is different from the before,  $v_t = v_t + 1$ . Obviously, the smaller the  $v_t$ , the better.

### 3.2.4 Fitness function of assembly sequence

There are a lot of assembly sequences evaluation standards, the aims of different assembly or different assembling process are various. So, we define some weight coefficients to change the different values, thus we can change the fitness function by changing the weight coefficients. Define  $w_p, w_s, w_d, w_t$  as the weight coefficients of value  $v_p, v_s, v_d, v_t$ . Define fitness function as the bellow expression:

$$f = w_p(n - v_p) + w_s(2n - v_s) + w_d v_d + w_t v_t \quad (11)$$

## 3.3 Assembly sequence planning by DPSO

Particle Swarm Optimization(PSO) is an efficient algorithm to find the optimal solution. Until now many scholars have improved the Particle Swarm Optimization to adjust assembly sequence planning by discretize it, called Discrete Particle Swarm Optimization(DPSO). So, we improve an algorithm based on the algorithm that proposed by Hong Yu at al<sup>[9]</sup>. The algorithm he proposed may fall into local optimum usually, so ,we improve it. Adding a  $Gt=t/N$ . When the algorithm begin ,let the particle move to the location of  $P_g$ ,in the course of time let the particle move to the location of the  $P_b$ .The frame of this algorithm is shown in fig.5.

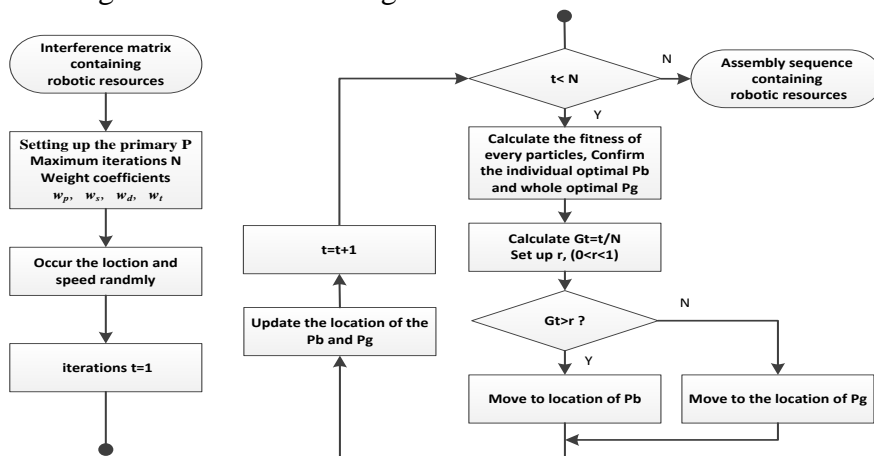


Fig. 5 The frame of Discrete Particle Swarm Optimization

## 4. Application

According to the method we proposed above, we give an example to test it. There is an radial bearing, containing 12 parts, as shown in fig.6. We can use our paper's method and traditional method to plan the assembly's sequence.

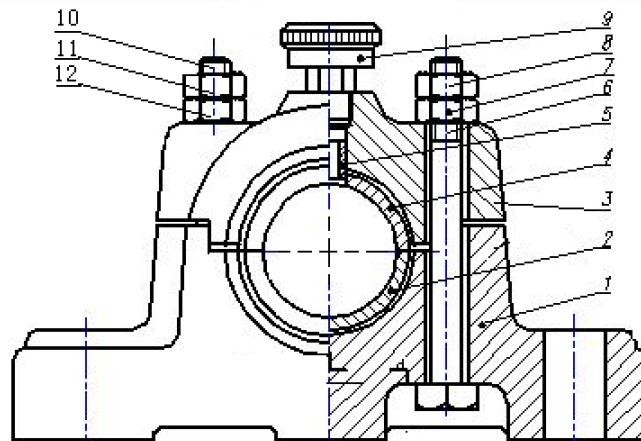


Fig. 6 radial bearing

Using the traditional interference matrix of part to plan the assembly sequence by DPSO, the optimal assembly sequence is:

Table 1 assembly sequence planning using interference matrix of parts

Assembly sequence	1	2	4	5	3	6	10	7	8	11	12	9
Assembly direction	-z	-z	-z	-z	-z	+z	+z	-z	-z	-z	-z	-z
Assembly tools	T1	T1	T1	T1	T1	T2	T2	T2	T2	T2	T2	T2
Value of assembly	vp=12; vs=14; vt=1; vd=2.											

Using the interference matrix containing robotic resources to plan the assembly sequence by DPSO, the optimal assembly sequence is (Atn represent the auxiliary tool is needed in the assembly, Atn represent the auxiliary tools should exit):

Table 2 assembly sequence planning using interference matrix containing robotic resources

Assembly sequence	6	10	1	2	4	5	3	7	8	11	12	9
Assembly direction	-z	-z	-z	-z	-z	-z	-z	-z	-z	-z	-z	-z
Assembly tools	T2	T2	T1	T1	T1	T1	T1	T2	T2	T2	T2	T2
Auxiliary tools	At1	At1	At1	—	—	—	—	—	—	—	—	At1
Value of assembly	vp=12; vs=18; vt=2; vd=0.											

The two method's convergence curve and distribution of the optimal fitness value in the process of searching the optimal assembly sequence are shown in fig.7 and fig.8.

From the fig.7 and fig.8, we can see that using the DPSO algorithm to deal with the both two matrix could also find the optimal assembly sequence. But, from table 1 and table 2, we can see the assembly sequence which come from by dealing with the traditional matrix is not so consistent with the fact. If we use this assembly sequence we should manufacture the tools specially to fix the part 1. And, when we use the assembly sequence shown in table 1, the part 6 and part 10 can not stay at their own assembly position as planned, they will drop because of gravity.

However, if using the method proposed by our paper we don't need to manufacture any special tool. And the assembly process in virtual environment is more aligned with it in the actual environment. The accident in assembly process like part 6 and part 10 will be avoided

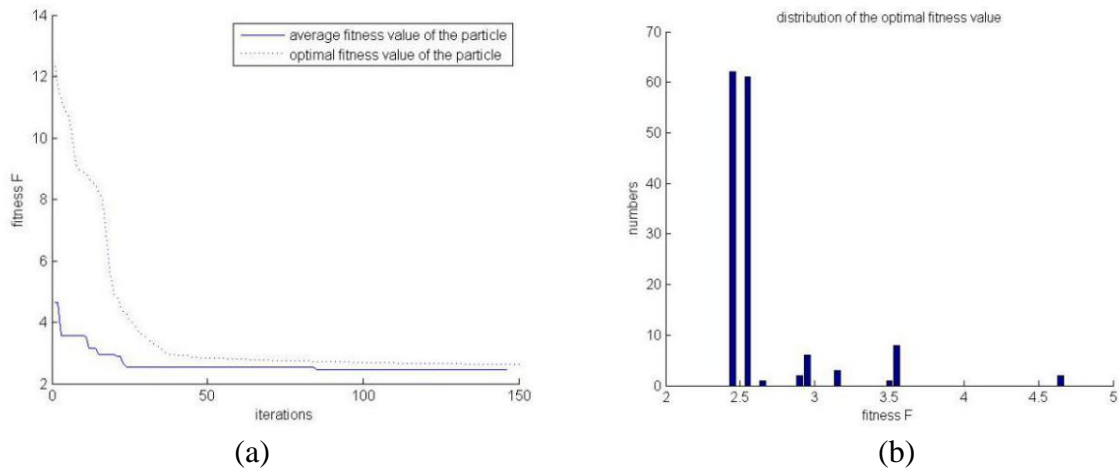


Fig. 7 Traditional method's convergence curve (a) and distribution of the optimal fitness value (b)

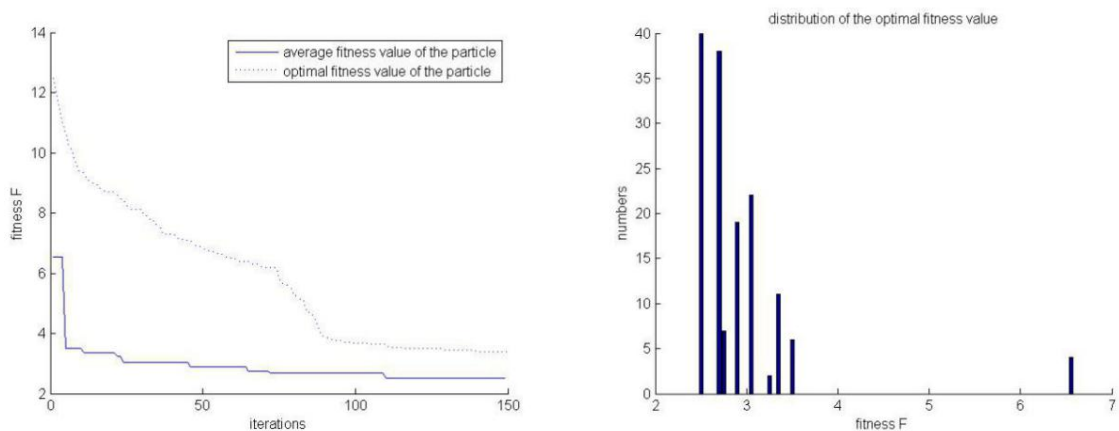


Fig. 8 Method proposed by our paper's convergence curve (a) and distribution of the optimal fitness value (b)

## 5. Conclusions

This paper proposed a method to structure a matrix containing robotic resource, which is used to plan assembly sequence for robot automatic operation. This method is more aligned with the actual assembly environment, and can give us much more information about the tools which would be used in assemble process. Meanwhile, the paper improve the Discrete Particle Swarm Optimization algorithm, to avoid the algorithm falling into local optimum usually, in the process of searching the optimal assembly sequence. From the application, we can see that the DPSO algorithm can find the optimal assembly sequence effectively by dealing with the interference matrix containing robotic resource.

## 6. References

- [1] Demoly F, Yan X T, Eynard B, et al. An assembly oriented design framework for product structure engineering and assembly sequence planning[J]. *Robotics and Computer-Integrated Manufacturing*, 2011, 27(1):33-46.
- [2] Papakostas N, Michalos G, Makris S, Zouzas D, ChryssolourisG. Industrial applications with cooperating robots for the flexibleassembly. *International Journal of Computer IntegratedManufacturing* 2011; 247:650-660.
- [3] Papakostas N, Pintzos G, Giannoulis C, Nikolakis N. Multi-criteria assembly line design underdemand uncertainty. (DET 2014), 8th International Conferenceon Digital Enterprise Technology 2014, 25-28 March, Stuttgart,Germany

- [4] Fan J, Ye Y, Cai J M. Multi-level intelligent assembly sequence planning algorithm supporting virtual assembly[J]. 2004, 4(4):3494-3499 vol.4.
- [5] Zhang J, Yang Y S, Feng S W. Method of Assembly Sequence Planning Based on Simulated EvolutionAlgorithm[J]. *Advanced Materials Research*, 2012, 490-495:1171-1175.
- [6] Wang L, Keshavarzmanesh S, Feng H-Y, Buchal RO. Assembly processplanning and its future in collaborative manufacturing: a review. *Int J AdvManuf Tech* 2009;41:132–44.
- [7] Bonneville F, Perrard C, Henrioud J-M. A genetic algorithm to generateand evaluate assembly plans. *IEEE Symp Emerg Tech Fact Autom* 1996;2:231–9.
- [8] De Lit P, Latinne P, Rekiek B, Delchambre A. Assembly planning with anordering genetic algorithm. *Int J Prod Res* 2001;39(16):3623–40.
- [9] YU Hong, WANG Cheng-en, YU Jia-peng. Assembly Sequence Planning Based on Particle Swarm Optimization Algorithm for Complex Product [J]. *Journal of Northeastern University (Natural Science)*, 2010, 31(2):261-264.