

# Green Product Module Partition Method Based on Improved Multi-objective Artificial Bee Colony Algorithm

Wei Wei<sup>1,\*</sup>, Yang Zhan<sup>1</sup>

<sup>1</sup>Beihang University, School of Mechanical Engineering and Automation, 100191 Beijing Haidian district, China;

**Abstract.** Modular design is an important design method in the mass customization for manufacturing industry. The purpose of this paper is to meet diverse market demands while reducing the impact of products on the ecological environment. Firstly, aiming at the product life cycle process, this paper summarizes the problems encountered in each stage of the product, and introduces five green product module partition principles. Then, through the component correlation matrix, the resource greenness objective function based on the whole life cycle and the polymerization degree objective function based on the component correlation matrix are established respectively by the axiomatic design theory which makes the product mapping from functional domain to structural domain. Next, an improved artificial bee colony algorithm is proposed. Based on the artificial bee colony algorithm, the algorithm applies congestion strategy and fast non-dominated sorting strategy to solve the module partition problem of product platform with multi-objective optimization, and a uniformly distributed pareto solution set is generated. Through above steps, the optimization results of module partition are obtained. Finally, an application example of aircraft tail horizontal stabilizer parts is given, and the advantages of the algorithm are proved by comparing with other algorithms.

## 1. Introduction

Mass customization (MC), as the most efficient and green mode of production in the current information age, not only meets the user's pursuit of product individualization, but also achieves the effective control of product cost [1]. This also makes the future market competition gradually expand from personalization to the whole product life cycle. Axiomatic design theory holds that design is a top-down process, which can be gradually developed from the high level of design abstract concept to the low level of detailed detail and solve the design problem in each domain. This theory provides a guiding framework for the product design and development process [2-4]. This method enables product mapping from functional domain to structural domain. Some experts and scholars have applied axiomatic design theory in the actual product design process [5-6]. Module design is a kind of product design. It aggregates the key features of the product and forms a subsystem with specific functions. By combining and configuring different subsystems, a parent system with certain functions can be formed. The system can produce a variety of products with similar functions but different performance and price according to different combinations [7]. Green design is to extend the design of products to the whole life cycle of the production, manufacture, use, maintenance, scrap and recovery without affecting the basic functions of products. It can reduce the burden on the environment caused by the

production, optimize the profits brought by products to enterprises, improve the utilization of resources as much as possible and indirectly enhance the market competitiveness of products. If we combine the theory of green design with the concept of module design, we can meet the functional and environmental attributes of products at the same time. From one aspect, we can use module design to build a product platform, which can quickly derive a wide range of products from product platform. Ultimately, the product development and renewal cycle will be significantly shortened. Under another aspect, based on the idea of green design, products can decrease even dismiss negative impacts on the environment. It is convenient to repurpose, upgrade, repair, disassemble, recycle and dispose of products. Therefore, green design and module design are combined to analyse the environmental impact of products on the basis of meeting the diversified and personalized needs of customers.

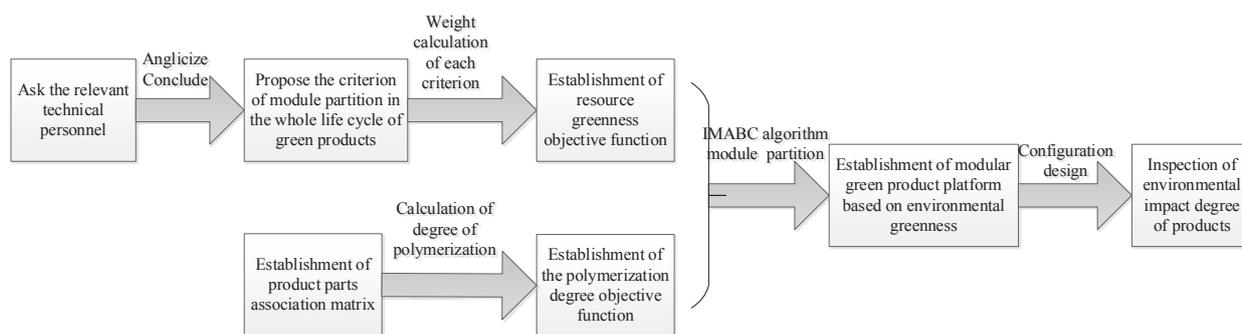
For the moment, many experts and scholars have studied module partition. GU et al. [8] pointed out the life cycle factors related to design objectives, and proposed a method about the integrated and module for product life. Cycle Han et al. [9] made a research on component model and its model repurposing, proposing a new method about module division for mechanical CAD component model. B.M.Li et al. [10] came up with a module partition method of CAD assembly model. N Janakiraman et al. [11] proposed a innovative algorithm for module partition

\* Corresponding author: [weiwei@buaa.edu.cn](mailto:weiwei@buaa.edu.cn)

problems by reconfigurable computing in the progress of VLSI design automation. H Li et al. [12] studied the relationship between ISP and physical and service modules, and proposed an interactive module design method. Guowei et al. [13] considered the functional and green performance, merged the basic units obtained from functional analysis, and put forward eight principles of product life cycle. Guoxi et al. [14] established the process model of functional module partition, introduced the key processes of acquiring customer requirements, and studied the method of module partition based on functional analysis. The axiomatic design concept is used in the above-mentioned module partition process.

At every point in the product lifecycle, it is usually difficult to quantify the product green attributes, and the weights between the quantified partition principles are difficult to determine effectively. Ultimately, the result of module partition is not accurate enough. Therefore, aiming at the whole product life cycle process, this paper summarizes the green problems in each stage by inquiring relevant technical personnel. Five principles of dividing green product modules throughout the product life cycle are introduced. Through the component correlation

matrix, the resource greenness objective function based on the whole life cycle and the polymerization degree objective function based on the component correlation matrix are established respectively by the axiomatic design theory which makes the product mapping from functional domain to structural domain. Aiming at the problem that basic artificial bee colony algorithm easily stuck in local optimal solution, this paper applies extensive learning strategy to ensure the multiplicity of the algorithm and avoid premature convergence. Next, through the improved congestion distance strategy, the running speed is improved, and the diversity and convergence of the population are guaranteed. Finally, an improved multi-objective artificial bee colony algorithm is proposed. Artificial Bee Colony algorithms provide optimized partitioning. The improved artificial ant colony algorithm for optimize the multi-objective problem of module division, and the result of module division is obtained. While fulfilling customers' functional requirements, it also improves the product's environmental friendliness. The specific module partition process is shown in the following figure.

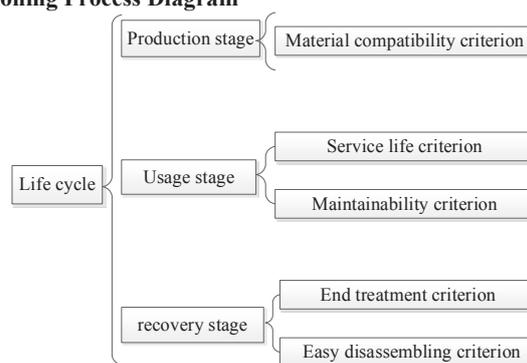


**Fig. 1. Module Partitioning Process Diagram**

## 2. Establishment of Module Product Platform Model

### 2.1 Determination of the Criteria for the green product module partition

In order to make the results of module partition for green products more reliable, the criteria for module partition should be specific to the product life cycle. It is divided into three stages, which be called the production stage, the usage stage and the recovery stage. The corresponding criteria for each stage of the product are shown in Figure 2.



**Fig. 2. Classification Chart of Module Partition Criteria**

#### 2.1.1 Material compatibility criteria

When the product enters the life stage of waste recovery, in order to recover part with low cost and high efficiency, we can divide the parts with the same or similar material into the same module as far as possible when dividing the product modules. If the compatibility of module components is better, the module can be directly integrated into the recovery process as a whole, and the disassembly process of the module can be omitted. It not

only reduces the difficulty of disassembly and recycling, but also controls the cost of recycling.

For any part  $a$  and part  $b$ , the component material compatibility correlation factor of criterion 1 is defined as follows:

$$R_1(a,b) = \begin{cases} 1, & a = b \\ r_1, & a \neq b \end{cases} \quad (1)$$

In the formula (1),  $0 < r_1 < 1$ ;  $Z_1$  represents that part  $a$  uses the same material as part  $b$ ;  $Z_2$  represents that part  $a$  uses the different material as part  $b$  and the  $r_1$  is the material compatibility value of part  $a$  and part  $b$ ;

### 2.1.2 Service Life Criteria

Basic parts with the same or similar service life should be divided into the same module. The service life of products is determined by the shortest service life in internal modules. If the failure of some modules leads to the overall scrap, it will cause a great waste of resources and increase the cost of production. The interaction factors that define the principle of service life between basic components are as follows:

$$R_2(a,b) = \frac{\min(L_a, L_b)}{\max(L_a, L_b)} \quad (2)$$

In the formula (2),  $L_a$ ,  $L_b$  are the service life of part  $a$  and part  $b$  respectively.

### 2.1.3 Maintainability criteria

Basic parts with the same or similar maintenance requirements should be divided into the same module. Good maintenance performance can reduce waste emissions, improve the integration of external maintenance resources, reduce the complexity of maintenance strategy, simplify maintenance work and reduce costs. Maintainability is shown in two aspects: fault correlation and maintenance correlation. Fault correlation refers to the state transition relationship between modules. When a module fails, another module that acts with it may be destroyed and new faults may occur. In this case, the correlation interaction value can be expressed in terms of the probability of the associated fault. Maintenance correlation means that different modules can share maintenance resources in case of failure, or they need to cooperate in the maintenance.

The interactive value of maintenance correlation can be expressed as the degree to which maintenance resources are shared among modules. Define maintainability interactivity factors between basic modules:

$$R_3(a,b) = \eta_{31}R_{31}(a,b) + \eta_{32}R_{32}(a,b)$$

(3)

In the formula (3),  $\eta_{31}, \eta_{32}$  are the weights of different maintenance relationships, and  $\eta_{31} + \eta_{32} = 1$ .

### 2.1.4 Easy disassembly criteria

At the end of product life cycle, disassembly process is often not smooth because that disassembly is not considered in initial design. Ultimately, the parts that have been disassembled can not be recycled well because they are damaged or can not be disassembled during disassembly.

To solve this problem, the easy disassembly criterion is proposed, and the disassembly process factor  $R_4(a,b)$  is as follows:

$$R_4(a,b) = W_{dc} \frac{\sum_{h=1}^n I_{dc}(a,b)}{n} + W_{db} I_{db}(a,b)$$

(4)

In the formula (4),  $W_{db}$  is the weight of spatial constraint relationship;  $W_{dc}$  is the weight of disassembly relationship,  $W_{dc} + W_{db} = 1$ ;  $I_{db}(a,b)$  is the value of spatial constraint relationship;  $I_{dc}(a,b)$  is the value of assembly relationship;  $n$  indicates the number of connection relations;  $h$  is the serial of connection points.

### 2.1.5 Terminal Processing Criteria

The disposal methods of scrap products are generally divided into recovery, waste and remanufacture. Recovery can be divided into material recovery and energy recovery according to different ways of recycling. Material recovery generally uses mechanical methods to deal with the material, just as tire rubber recovery usually uses a grinder to grind rubber into particles, and then recycle; Energy recovery refers to the extraction of energy from materials by incineration. Waste refers to the direct discarding of materials without any treatment. This method not only wastes materials, but also destroys the environment. Remanufacture is usually to disassemble and clean the parts, and then repair the parts through secondary processing. Processes vary greatly between different treatment methods. Therefore, when dividing the modules, the parts with the same processing mode are divided into the same modules, which can reduce the difficulty in product recovery.

For any part  $a$  and part  $b$ , the correlation factor of the terminal processing mode in criteria 1 is defined as follows:

$$R_5(a,b) = \begin{cases} 0, & a = b \\ r_5, & a \neq b \end{cases} \quad (5)$$

In formula (5),  $r_5$  is the quantitative value of the correlation factor for the end-processing mode;  $a, b$  are related parts;  $a=b$  means that  $a$  and  $b$  are handled in the same way.

## 2.2 Establishment of objective function of resource greenness

The calculation of resource greenness objective function is based on material compatibility criterion, service life criterion, maintenance criterion, easy disassembly criterion and terminal processing criteria. By adjusting the module partition results of each component, the green attribute of resources for the whole product can be better.

Through the above resource greenness attributes, the resource greenness matrix of components in the module cluster can be established.

### 2.2.1 Calculation of resource greenness attribute values

The resource greenness between two arbitrary parts is calculated by the method in chapter 2.2. The attribute values of resource greenness are as follows:

$$D_1 = \begin{bmatrix} R(1,1) & R(1,2) & R(1,3) & R(1,4) & R(1,5) \\ R(2,1) & R(2,2) & R(2,3) & R(2,4) & R(2,5) \\ \dots & \dots & \dots & \dots & \dots \\ R(n,1) & R(n,2) & R(n,3) & R(n,4) & R(n,5) \end{bmatrix} \quad (7)$$

In the formula (7),  $R(a,b)$  is the resource greenness attribute value of the resource for the part  $a$  and the part  $b$ .  $R(a,a)=1$  represents attribute value of part  $a$  is 1.  $R(a,b)=R(b,a)$  represents that the comprehensive attribute values of part  $a$  and part  $b$  are constant and  $a,b=1,2,3\dots n$ .

### 2.2.2 Construction of mathematical optimizing model for resource greenness among modules

The objective function for resource greenness of integrated relative resources among modules is:

$$f_1(x) = \sum_{a=1}^{k_1} \sum_{b=1}^{k_2} \left[ \frac{R(a,b)}{k_1 k_2} \right] \quad (8)$$

In the formula (8),  $R(a,b)$  is the resource greenness attribute value between the part  $a$  and the part  $b$ ;  $K_1$  and  $K_2$  are the number of parts in the two modules;  $x$  is the independent variable, that is, the module partition scheme.

## 2.3 Establishment of Function-Structure Object Function

In order to achieve green product module partition, besides considering the resource greenness, the functional structure relationship between product parts is also an important part of module partition. When the functional structure of each component in the module is aggregated and the functional structure between modules is independent, the final result of module partition will be more reasonable. In this paper, the function-structure

$$D_2 = \begin{bmatrix} K(1,1) & K(1,2) & K(1,3) & K(1,4) & K(1,5) \\ K(2,1) & K(2,2) & K(2,3) & K(2,4) & K(2,5) \\ \dots & \dots & \dots & \dots & \dots \\ K(n,1) & K(n,2) & K(n,3) & K(n,4) & K(n,5) \end{bmatrix} \quad (9)$$

In the formula (9),  $K(a,b)$  is the polymerization degree between the part  $a$  and the part  $b$ .

$$R(a,b) = \sum_{l=1}^5 \lambda_l \times R_l(a,b) \quad (6)$$

In the formula (6),  $R(a,b)$  is the greenness attribute value of part  $a$  and part  $b$ ;  $\lambda_l$  is the weight of criterion  $l$ ;  $R_l(a,b)$  is the correlation factor value of part  $a$  and part  $b$  in the module partition criterion  $l$ , such as  $R_1(a,b)$  is the material compatibility factor value of part  $a$  and part  $b$ .

Construct the resource greenness matrix of components in the module cluster:

objective function of components is introduced to calculate and analyse the internal polymerization degree and external polymerization degree. Finally, the reasonable module partition result of components is obtained. The process of establishing objective function is shown in the following figure.

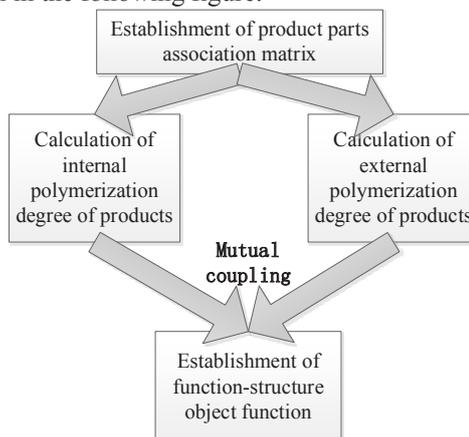


Fig. 3. Establishment of function-structure object function

### 2.3.1 Establishment of components association matrix

In order to express the interrelated value between parts, the relationship between two parts is expressed in the form of fuzzy relation, in which the values of each element are limited to  $[0,1]$ . By calculating, the component matrix is obtained as follows:

### 2.3.2 Calculation of internal polymerization degree and external polymerization degree

According to the correlation matrix of product parts, internal polymerization degree of module  $k$  is as follows:

$$F_1^k = \sum_{i=1}^{P^k-1} \sum_{j=i+1}^{P^k} \left[ \frac{K(a,b)}{P^k(P^k+1)/2} \right] \quad (10)$$

In the formula (10),  $K(a,b)$  is the attribute value of internal polymerization degree between part  $a$  and part  $b$ , and  $P^k$  represents the number of parts in module  $k$ .

External polymerization degree between two modules is as follows:

$$F_2^k = \sum_{i=1}^l \sum_{j=1}^k \left[ \frac{K(a,b)}{P^l P^k} \right] \quad (11)$$

In the formula (11),  $P^l$  and  $P^k$  is the number of parts in module  $l$  and module  $k$ ;  $K(a,b)$  is the attribute value of internal polymerization degree between part  $a$  and part  $b$ ;

### 2.3.3 Establishment of Function-Structure Object Function

According to the previous step, the internal polymerization degree of all modules and the external polymerization degree among modules are calculated.

The internal polymerization degree of all modules is as follows:

$$F_1 = \sum_{g=1}^U F_1^g \quad (12)$$

External polymerization degree among modules is as follows:

$$F_2 = \sum_{s=g+1}^U \sum_{g=1}^{U-1} F_2^{gs} \quad (13)$$

In the formula (12) and formula (13),  $U$  represents the number of all modules;  $g$  is the module  $g$ ;  $s$  is the module  $s$ .

The product function-result polymerization function is as follows:

$$f_2(x) = S \sqrt{F_1 / F_2} \quad (14)$$

In the formula (14),  $S$  is the contribution degree of related parts to product function;  $F_1$  is the internal polymerization degree of all modules;  $F_2$  is the external polymerization degree among modules.

## 2.4 Multi-objective optimization model for module partition

In product module design, the module partition of product platform needs to take into account the resource greenness of product resources and the related functions. In this process, resource greenness objective function and function-structure objective function are the optimization objectives, and product module partition results are independent variables. After summing up, the results are shown as follows:

$$f(x) = \begin{cases} [f_1(x), f_2(x)] \\ st: h_k(x) = 0, k=1, 2, 3, \dots, K \\ g_j(x) \leq 0, j=1, 2, 3, \dots, J \\ x = (\delta_1, \delta_2, \dots, \delta_N) \\ \delta_i \in (0, M) \end{cases} \quad (15)$$

In the formula (15), objective function  $f(x)$  defines two objective functions  $f_1(x)$ ,  $f_2(x)$ , which need to be optimized;  $h_k=0$  defines equality constraints;  $g_j(x) \leq 0$  defines inequality constraints.

## 3. Improved algorithm analysis

### 3.1 Basic ABC algorithm

The algorithm is an intelligent optimization algorithm [15]. It's basic idea of this algorithm derived from the process of collecting honey by bees in nature. In order to extract more high-quality honey, bee populations in nature often exchange information in the process of honey harvesting, and transmit high-quality food source information to their peers. In this algorithm, the search for high quality food source is to find the optimal solution. The bee population is divided into three types: employed bees, scout bees and follower bees. Their role is to discover and abandon ineffective food sources (no improvement in continuous limit times) and find new ones [16].

In this algorithm, the initialization of the solution is carried out as follows:

$$x_{i,j} = x_j^{\min} + \phi_{i,j} \times (x_j^{\max} - x_j^{\min}) \quad (16)$$

In the formula (16),  $[x_j^{\min}, x_j^{\max}]$  is the value range of the solution and  $\phi_{i,j}$  is between  $[0,1]$ .

In ABC algorithm, employed bees and follower bees update the solution (looking for food sources) in the same way. The search strategy is as follows:

$$v_{i,j} = x_{i,j} + \phi_{i,j} \times (x_{i,j} - x_{k,j}) \quad (17)$$

In the formula (17),  $i = 1, 2, \dots, P$  and  $j = 1, 2, \dots, V$ .  $P$  represents the number of solutions and  $V$  represents the dimension of solutions. In ABC, only one dimension selected randomly in  $V$  is updated at a time.  $x_{i,j}$  is the  $j$ -th dimension.  $x_k$  is a neighbour solution that is randomly selected in a population and is not equal to  $x_i$ .  $V_i$  is the solution after the update.  $\phi_{i,j}$  is uniform random numbers between  $[-1,1]$ . When  $V_i$  is better than the  $x_i$ , greedy selection is adopted and  $V_i$  directly replaces  $x_i$ .

When the employed bees finish their work by formula (16), the employed bees transmit the information (fitness) of the searched solution to the follower bees. In term of the selection probability, the follower bees select the solution by roulette to update. The selection probability of each solution is determined by formula (18) and formula (19):

$$fit_i = \begin{cases} 1 / (f(x_i) + 1), f(x_i) \geq 0 \\ 1 + abs(f(x_i)), f(x_i) < 0 \end{cases} \quad (18)$$

$$P_i = \text{fit}_i / \sum_{i=1}^{SN} \text{fit}_i \quad (19)$$

In the formula (18) and formula (19),  $\text{fit}_i$ ,  $f(x_i)$  and  $P_i$  refer to fitness, objective function and selection probability of the first food source, respectively. From formula (18) and formula (19), it can be seen that the higher the fitness, the greater the probability that the follower bees choose the food for solving the minimum problem. In the stage of scout bee, it is judged whether the maximum number of successive update failures in all solutions reaches the limit value. If it is reached, the scout bees are started and the corresponding solution is initialized by formula (16).

### 3.2 The improved multi-objective ABC algorithm

When using basic ABC algorithm for single objective optimization, the solve will be fell into the local optimum easily [17]. Multi-objective optimization problems need to find a series of pareto optimal solutions satisfying certain constraints, which is more difficult. Multi-objective optimization generally involves two performance indicators: convergence and diversity. The two performance indicators make the searched non-dominated solution evenly distributed in the solution space. The NSGA-II [18] uses the congestion distance instead of the fitness sharing strategy which needs to specify the sharing radius to keep the individual distribution uniformly. NSGA-II algorithm is used as an elite selection criterion in the same-level comparison after rapid sorting, which enables individuals in pareto domain to extend to the whole pareto domain. Based on the above work, we propose a multi-objective ABC algorithm with improved congestion strategy and fast non-dominated sorting.

#### 3.2.1 Artificial bee colony operator

##### i. Initialization stage

In the initialization phase, set the relevant parameters of the population, including population number, food source number, control parameters, maximum cycle number and  $D$ -th dimensional solution space. When all parameters are determined, the population is initialized.

In this paper, we give a food source  $x_i=(x_{i1}, x_{i2}, x_{i3}, \dots, x_{iD})$  and the location of  $x_i$  as the  $i$ -th food source in the population. In the solution space, the location of each  $x_i$  is initialized as follows:

$$x_{id} = L_d + \text{rand}_{i,d} (U_d - L_d) \quad (20)$$

In the formula,  $\text{rand}$  is a random number with uniform distribution in  $(0,1)$ ,  $L_d$  and  $U_d$  are upper and lower bounds of the  $D$ -th dimension for the whole search space respectively;  $\text{rand}$  is a random number with uniform distribution in  $(0,1)$ ;  $d = 1, 2, \dots, D$ .

##### ii. Employed bee stage

The employed bees search for the corresponding food source, record the relevant information (the location of the honey source and the amount of honey source), and share the information with the follower bees. For each food source, the feeding route for the employed bees is based on a random selection of neighbour  $k$  to explore a new

food source. Specifically, the calculation method of its temporary position is as follows:

$$V_{i,j} = x_{i,j} + \varphi_{i,j} (x_{i,j} - x_{k,j}) \quad (21)$$

In the formula (21):  $V_{ij}$  is a new position of  $i$ -th individual;  $i$  is the individual currently being searched;  $j$  is the dimension of random selection;  $k$  is a neighbour of random selection which is not equal to  $i$ ;  $\varphi_{ij}$  is the random number in  $[0,1]$ .

After obtaining a new temporary food source, the fitness of the source is assessed by the greed rule. Then the new individual is considered to be better and replaces the original individual.

##### iii. Follower bee stage

According to the information of honey sources provided by employed bees, the follower bees determine the better food sources for further exploitation. At this stage, the follower bees calculate the probability that the individual is selected to evolve in term of the fitness of the individual, and then use roulette method to select the individual for further exploration. The calculation method of the  $P_i$  for the  $i$ -th individual is as follows:

$$P_i = 1 - \text{fit}(x_i) / \sum_{j=1}^N \text{fit}(x_j) \quad (22)$$

In the formula (22),  $N$  is population size;  $\text{fit}(x_i)$  is the fitness value of  $x_i$ ; The larger the  $P_i$ , the more times the following bee searches near a better food source (with a smaller fitness value).

##### iv. Scout bee stage

Employed bees will be scouts as food is run out and cannot be improved. Next, a new food source is randomly generated by equation (20).

#### 3.2.2 Fast non-dominated sorting.

This paper refers to the fast non-dominated sorting method of NSGA-II. Set the population size to  $N$ . The new population  $Q_t$  produced in the  $T$  generation is combined with the parent population  $P_t$  to form  $R_t$ , and the population size is  $2N$ . Then  $R_t$  obtains a series of non-dominant pareto solutions based on dominance relations:  $F_1, F_2, F_3, \dots$  (The level of disassembly decreases in turn). When  $N$  is greater than the number of non-dominant solutions in  $F_1$ , then  $x$  solutions of  $F_1$  are selected to population  $P_{t+1}$ . The residual solution in  $P_{t+1}$  will continue to be selected in the remaining layers. When the residual solution is selected to  $F_3$  and the number of solutions for population  $P_{t+1}$  exceeds  $N$ , the non-dominant solutions in  $F_3$  need to be ranked by the congestion distance, and the better  $R$ -th individuals are preferred to enter  $P_{t+1}$ .

#### 3.2.3 Index-based congestion distance selection method

After non-dominated sorting, the congestion distance is calculated and sorted for the same level of non-dominated decomposition. Firstly, the congestion distance of the extreme solution (the maximum or minimum solution on each target) is set to infinity. For the other solution  $i$ , the congestion distance is calculated as follows:

$$D_i = \sum_{j=1}^M (|f_{i+1,j} - f_{i-1,j}|) \quad (23)$$

In the formula (23):  $D$  represents the congestion distance;  $M$  represents the amount of objective functions;  $f$  is the objective function value. As the objective functions increases, the proportion of non-dominant solutions to all solutions increases, and it becomes harder to identify the variations. The above method will delete the important individuals in the dense area and damage the diversity of pareto frontier. To this end, the index-based congestion distance strategy is defined as follows:

$$DI_i = \sum_{j=1}^M (|F_{I_{\varepsilon+}}(x_{i+1,j}) - F_{I_{\varepsilon+}}(x_{i-1,j})|) \quad (24)$$

In the formula (24),  $M$  is the target number;  $F_{I_{\varepsilon+}}(x_i)$  is the index-based fitness of  $i$ -th individual. The calculation is as follows:

$$\begin{cases} F_{I_{\varepsilon+}}(x_i) = \sum_{x_j \in P} e^{-I_{\varepsilon+}(x_j - x_i)/(c-s)} \\ c = \max_{x_i, x_j \in P} |I_{\varepsilon+}(x_j - x_i)| \end{cases} \quad (25)$$

In the formula (25), the parameter  $s$  is the proportional factor. Based on this definition, the calculation process of congestion distance is as follows:

Step 1: Initializes the congestion distance of the non-dominated decomposition solution, i.e.  $D_{i1} = 0$ .

Step 2: The individual is ranked by calculating the value of each objective function; the boundary individual is pre-set to infinity to make sure that it is available in the later selection process;

Step 3: The congestion distance for the non-dominated solution is calculated by formula (24).

Step 4: Determines the solution  $D_Y$  with the smallest congestion distance in the set and deletes it.

Step 5: Recalculates congestion distances of  $D_{Y+1}$  and  $D_{Y-1}$  by the formulas (26), respectively.

$$DI_{DY+1} = \sum_{j=1}^M (|f_{DY+2,j} - f_{DY-1,j}|) \quad (26)$$

$$DI_{DY-1} = \sum_{j=1}^M (|f_{DY+1,j} - f_{DY-2,j}|)$$

Step 6: If the file size is greater than the maximum default size, come back to step 3; else, end the method.

### 3.2.4 Solution flow

The final solution process is shown in the Figure 4.

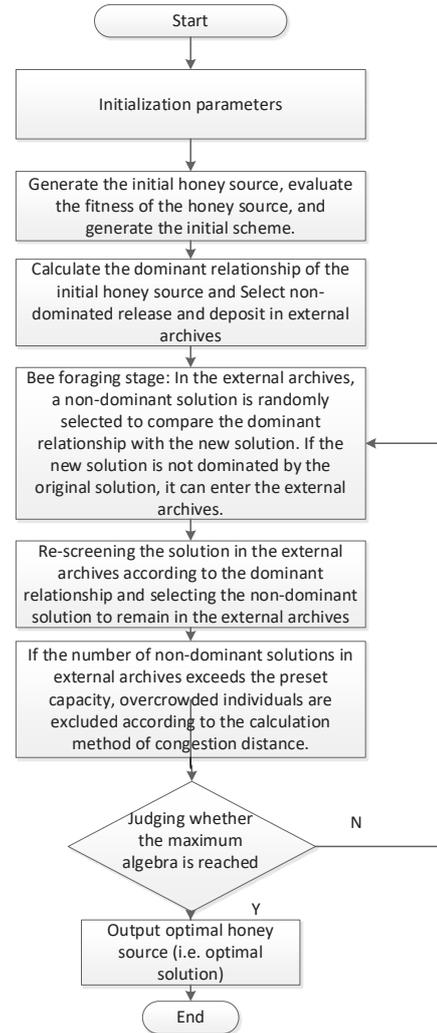
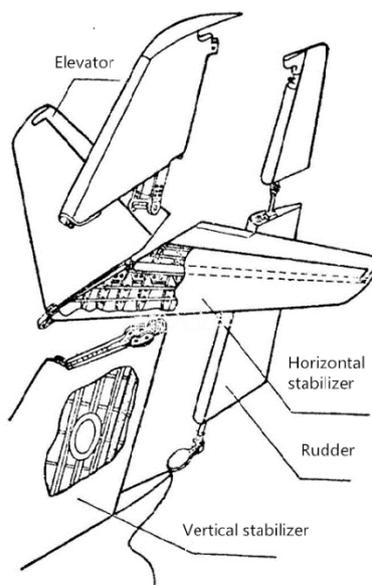


Fig. 4. Improved flow chart of ABC algorithm

## 4. Case verification

### 4.1 Module partition analysis of aircraft tail horizontal stabilizer parts

The aircraft tail is the key function module in the aircraft [19]. The horizontal stabilizer is taken as an application example of module partition. Figure 5 is a schematic view of the aircraft tail. The horizontal stabilizer is preliminarily decomposed into 25 parts as shown in Table 1:



**Fig. 5.** Aircraft tail structure schematic diagram.

**Table 1.** List of Horizontal Stabilizer Parts for Aircraft Tail

number	name	number	name	number	name
1	Fiberglass honeycomb	12	Hinge rib	23	Bearing lug
2	Elevator hinge	13	Rear spar	24	Elevator front spar
3	Elevator inboard	14	Skin panel	25	Horizontal stabilizer rear
4	Fiberglass honeycomb trailine edge	15	Actuator fitting		
5	Skin splice plates	16	Front spar		
6	Hinge ribsee	17	Open access hole		
7	Fixed trailine edge	18	Removable leading edge		
8	Aluminium spars and ribs	19	Auxiliary spar		
9	Removable panel	20	Sheet metal ribs		
10	Aluminium honeycomb	21	Access hole		
11	Stiffeners	22	Splice fitting		

**4.2 Module partition based on improved multi-objective ABC algorithm**

**4.2.1 Determination of module partition criteria among product parts**

From Formula (1) to Formula (5) and the relationships among the components of the aircraft tail horizontal stabilizer, five relevant criterion factors  $R_k(a,b)$  are computed.

**4.2.2 Determination of the weight of resource green degree criterion**

In this paper, according to the green evaluation criterion, the weight of the module partition criterion for the

horizontal stabilizer is determined. The weights of five criteria based on resource greenness are respectively as follows:  $\lambda = \{0.08, 0.31, 0.28, 0.16, 0.17\}$

**4.2.3 Establishment of resource greenness objective function**

By formula (6) and  $R_k(a,b)$ , the attribute values of resource greenness between any two parts are calculated, and the resource greenness matrix is established. The matrix is shown as follows:

**Table 2.** Green Degree Matrix of Parts and Components Resources.

1	0.06	0.09	0.44	0.08	0.22	0.06	0.02	0.07	0.14	...	0.11	0.03	0.04	0.84
0.06	1	0.24	0.35	0.35	0.68	0.14	0.46	0.83	0.13	...	0.48	0.33	0.14	0.03
0.09	0.24	1	0.71	0.71	0.48	0.73	0.32	0.32	0.02	...	0	0.32	0.05	0.17
0.44	0.35	0.71	1	0.87	0.12	0.49	0.33	0.22	0.17	...	0.11	0.18	0.02	0.18
0.08	0.35	0.71	0.87	1	0.22	0.73	0.32	0.17	0	...	0.21	0.26	0.02	0.08
0.22	0.68	0.48	0.12	0.22	1	0.19	0.64	0.76	0.01	...	0.33	0.27	0.04	0.09
0.06	0.14	0.73	0.49	0.73	0.19	1	0.33	0.33	0	...	0.02	0.32	0.16	0.18
0.02	0.46	0.32	0.33	0.32	0.64	0.33	1	0.67	0.03	...	0.17	0.19	0.04	0.03
0.07	0.83	0.32	0.22	0.17	0.76	0.33	0.67	1	0	...	0.17	0.19	0.04	0.03
0.14	0.13	0.02	0.17	0	0.01	0	0.03	0	1	...	0.60	0.02	0.34	0.02
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0.11	0.48	0	0.11	0.21	0.33	0.02	0.17	0.17	0.60	...	1	0.32	0.49	0.25
0.03	0.33	0.32	0.18	0.26	0.27	0.32	0.19	0.19	0.02	...	0.32	1	0.01	0.23
0.04	0.14	0.05	0.02	0.02	0.04	0.16	0.04	0.04	0.34	...	0.48	0.01	1	0.01
0.24	0.03	0.17	0.18	0.08	0.09	0.18	0.03	0.03	0.02	...	0.25	0.23	0.01	1

From the formula (8), the objective function  $f_1(x)$  of comprehensive relative resource greenness among modules is obtained.

form of fuzzy relation, in which the values of each element are limited to [0,1]. By calculating, the component matrix is obtained as Table 3.

**4.2.4 Establishment of product function-structure object function**

In order to express the interrelated value between parts, the relationship between two parts is expressed in the

**Table 3.** Parts correlation matrix.

1	0	0.07	0.07	0.07	0	0.08	0	0	0	...	0	0	0	0.84
0	1	0.25	0.25	0.25	0.75	0.23	0.74	0.69	0	...	0.29	0.26	0	0
0.07	0.25	1	0.68	0.68	0.23	0.84	0.24	0.22	0	...	0	0.19	0	0.09
0.07	0.25	0.68	1	0.77	0.24	0.67	0.25	0.18	0	...	0	0.22	0	0.09
0.07	0.25	0.68	0.77	1	0.22	0.66	0.24	0.25	0	...	0	0.22	0	0.09
0	0.75	0.23	0.24	0.22	1	0.22	0.72	0.86	0	...	0.27	0.29	0	0
0.08	0.23	0.84	0.67	0.66	0.24	1	0.22	0.22	0	...	0	0.22	0	0.09
0	0.74	0.24	0.25	0.24	0.72	0.22	1	0.72	0	...	0.27	0.29	0	0
0	0.69	0.24	0.18	0.25	0.86	0.22	0.72	1	0	...	0.27	0.29	0	0
0	0	0	0	0	0	0	0	0	1	...	0.55	0	0.56	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	0.29	0	0	0	0.27	0	0.27	0.27	0.55	...	1	0.29	0.55	0
0	0.26	0.19	0.22	0.22	0.29	0.22	0.29	0.29	0	...	0.29	1	0	0
0	0	0	0	0	0	0	0	0	0.56	...	0.55	0	1	0
0.84	0	0.09	0.09	0.09	0	0.09	0	0	0	...	0	0	0	1

**Table 4.** "Components-Functions" Contribution Matrix.

	External protection function	Internal support function	Lifting function	Assembly assistance function	Connection function
1	0.25	0	0	0	0.75
2	0	0	1	0	0
3	0	0	1	0	0
4	0.86	0	0	0	0.14
5	0.68	0	0	0	0.32
6	0	0.81	0	0	0.19
7	0.34	0	0	0.1	0.56
8	0	1	0	0	0
...	...	...	...	...	...
25	0	0.85	0.15	0	0

In addition, the idea of robustness is introduced into the functional matrix. According to the actual needs, five main functions of the module to be divided for the example product of aircraft tail horizontal stabilizer are determined. Considering the area of components and the portability of universality, the five main functions of the module are as follows: external protection function, internal support function, lifting function, Assembly assistance function and connection function. The contribution matrix of "parts-functions" is obtained as shown in Table 4, where the axiomatic design concept is used to make the product mapping from functional domain to structural domain. Formula (10) and formula (11) are used to calculate internal polymerization degree and external polymerization degree of the module k. Formulas (12) and (13) are used to calculate the sum of the internal polymerization degree of all modules and the external polymerization degree of each module. Finally, the product function-result polymerization function  $f_2(x)$  is obtained from the contribution matrix shown in formula (14) and Table 4.

**4.2.5 Establishment of mathematical model**

A multi-objective optimization mathematical model to the aircraft tail horizontal stabilizer is established. Its objective function is as follows:

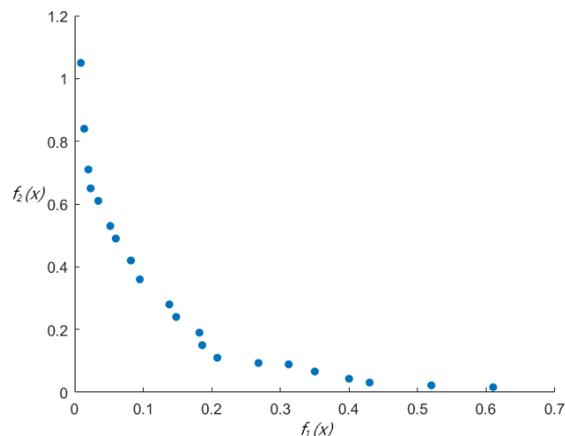
$$f(x) = \begin{cases} f_1(x) = \sum_{a=1}^{k_1} \sum_{b=1}^{k_2} \left[ \frac{R(a,b)}{k_1 k_2} \right] \\ f_2(x) = S\sqrt{F_1 / F_2} \end{cases} \quad (22)$$

When the number of modules is the square root of the total number for parts, the result of module partition is the best. Therefore, the paper sets the number of modules as the square root of the total number for parts  $N$ . The constraints are shown in formula (23).

$$\begin{cases} k_1, k_2 < \sqrt{N} + 1 \\ k_1, k_2 > 0 \\ k_1, k_2 \in 1, 2, \dots, n \end{cases} \quad (23)$$

**4.2.6 Multi-objective optimization based on improved multi-objective ABC algorithm**

Let the population size  $SN = 50$  and the external file size  $A = 100$ , and get a reasonable pareto solution of 20 solutions. The pareto solution is as follows:



**Fig. 6.** IMOABC-pareto solution set.

As shown in the Figure (6), the abscissa is the objective function  $f_1(x)$ ; the ordinate is the objective function  $f_2(x)$ ; the optimal pareto-front is convex; the fitting curve has an inflection point  $(f_0, f_1)$  near  $f_1(x) = 0.2$ . In  $(0, f_0)$  interval, the curve is steep, but in  $(f_0, f_{max})$  interval, the curve is smooth. When partitioning modules, it is necessary to ensure that the target noise function has the least impact on the partitioning results, and the sensitivity of the target criterion function is low. Therefore, considering the two-objective optimization and the robust design of module partition, when  $f_1(x)=0.221$  and  $f_2(x)=0.124$  (the nearest to the right of the inflection point), the corresponding module partition results are the best. The results are shown in Table 5.

**Table 5.** Optimized results of module partition.

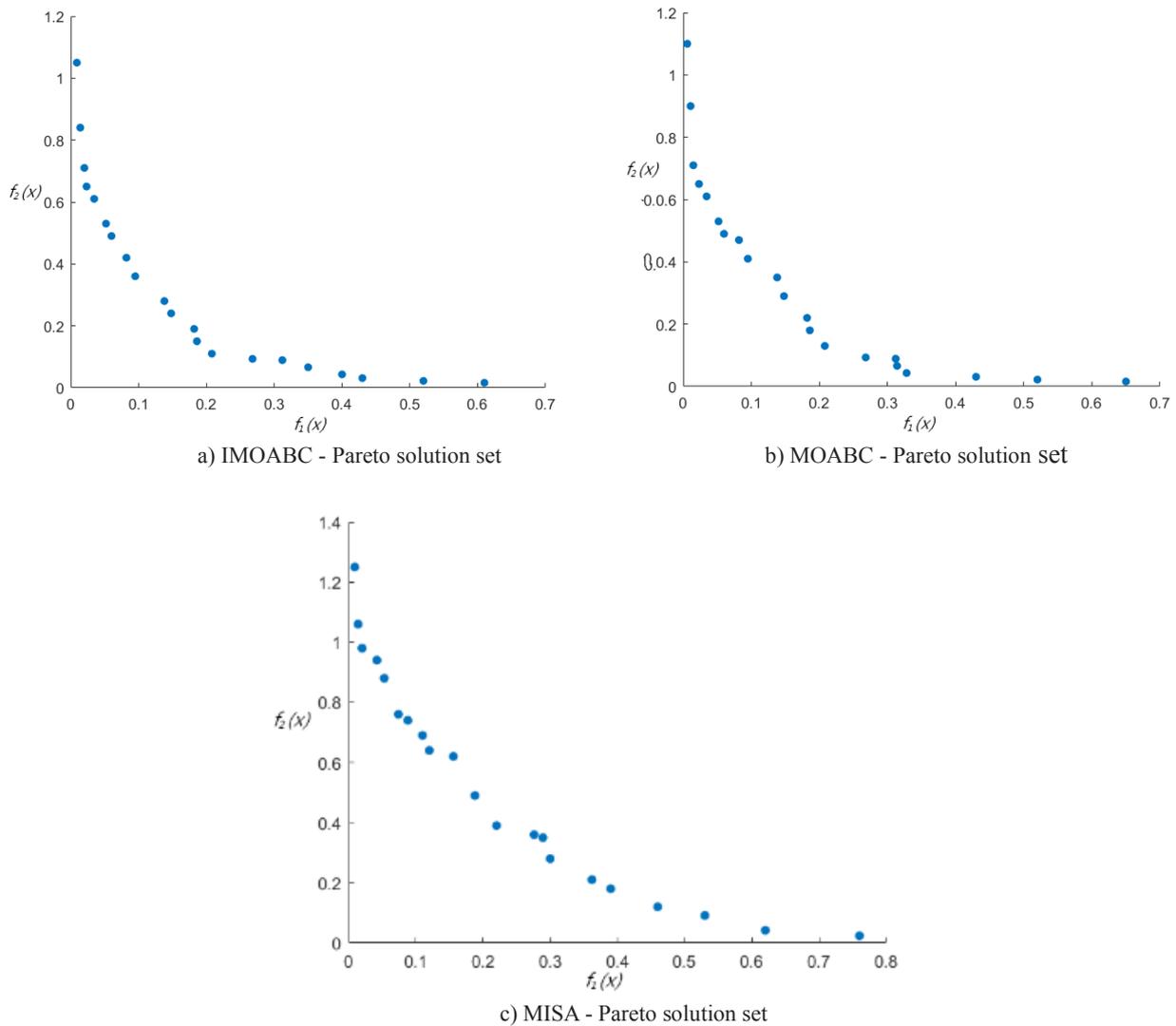
Module	Part	Function
1	[5,10,11,14]	External protection function
2	[6,8,9,12,13,16,19,20,23,25]	Internal support function
3	[2,3,24]	Lifting function
4	[15,17,21]	Assembly assistance function
5	[1,4,7,18,22]	Auxiliary connection function

The data shows that the module partition result in this paper is more reasonable in terms of green and function than the traditional design method. It is proved that the method works in the module partition.

**4.3 Comparison and analysis of algorithms**

The IMOABC used in this paper is compared with the MOABC and the MISA. The analysis is made from three

aspects: the wide distribution, the uniformity and the approximation of the optimal solution. The third item tests the convergence of the algorithm, and the other two items show the diversity of pareto-optimal solutions on the pareto front. Each algorithm is computed 20 times. And the pareto front is shown in Figure 7.



**Fig. 7.** Comparison of pareto solution set between IMOABC algorithm and other algorithms.

From Figure 7, we can see that the algorithm in this paper has obvious advantages over other methods in terms of breadth and uniformity, and its approximation is better. The results of the three algorithms are compared by box chart statistics, and the comparison results of each index

can be obtained from Table 6. Comparing with the other two algorithms, the convergence of the optimal solution obtained by IMOABC algorithm is better. Moreover, it has obvious advantages in the uniformity and breadth of distribution on the pareto front.

**Table 6.** Comparison of evaluation indexes of three algorithms.

Evaluation index	Algorithm for optimization		
	IMOABC	MOABC	MISA
Uniformity of optimal solution distribution	good	medium	poor
Approximation of optimal solutions to pareto front	medium	medium	poor
The width of the distribution of optimal solutions to pareto front	medium	medium	medium

## 5. Conclusion

There are three important points of this paper: a modularization method for green products with full life cycle, a mathematical model based on green resource and product polymerization, and an improved multi-objective Artificial Bee Colony algorithm. A well known problem in modular design is the partitioning of the modules. We use this method to solve the problem in modular design. This paper is oriented to product life cycle process. By visiting and inquiring the relevant technical personnel, it summarizes the green problems in each stage. Five green product module partition principles throughout product life cycle are introduced. Through the component correlation matrix, the resource greenness objective function based on the whole life cycle and the polymerization degree objective function based on the component correlation matrix are established respectively. Finally, a multi-objective optimization model of green product module partition is formed. A non-dominated sorting strategy is introduced on the basis of ABC algorithm for module partition. Based on the index calculation, the individual congestion assessment strategy is improved, and an improved multi-objective ABC algorithm is proposed. The data shows that the module partition result in this paper is more reasonable in terms of green and function than the traditional design method. The improved multi-objective Artificial Bee Colony algorithms provide fairly optimized partitioning. In order to verify the validity of the algorithm, IMOABC algorithm, MOABC algorithm and MISA algorithm are used to solve the module partition problem. By comparing the results, it is found that the improved multi-objective ABC algorithm has great advantages in terms of breadth, uniformity and the optimal solution approximation. The improved multi-objective Artificial Bee Colony algorithms provide fairly optimized partitioning.

## 6. Acknowledge

This research is supported by the National Natural Science Foundation of China (Grant number 51675028), The National Key Research and Development Program of China (Grant number 2018YFB1701703, 2017YFB1104201), Aeronautical Science Foundation of China (Grant number 20171651015) and State Key Lab of Digital Manufacturing Equipment & Technology (Grant number: DMETKF2017020).

## 7. Reference

1. D. Gabriel, Y. Reich. *Computer-Aided Design* ,35.9, 791-806 (2003)
2. S. Nam P. *The principles of design (oxford series on advanced manufacturing)* (1990)
3. S. Nam P. *Axiomatic Design: Advances and Applications*. New York: Oxford University Press. (2001)

4. S. Nam P. *Complexity: Theory and Applications*. New York: Oxford University Press. (2005)
5. Kannan, Devika, et al. *Journal of Cleaner Production* 96, 194-208 (2015)
6. Kulak, Osman, et al. *Expert Systems with Applications* 37.9, : 6705-6717 (2010)
7. Peien, Pan Shuangxia Gao Fei Feng. *Chinese Journal of Mechanical Engineering* ,7 (2003)
8. G. P., Hashemian, M. Hashemian, S. Sosale., & E. Rivin. *CIRP Annals*, 46.1, 71-74. (1997)
9. H. Zhoupeng, et al. *Journal of Advanced Mechanical Design, Systems, and Manufacturing* ,12.1 (2018)
10. L. B. M., and S. Q. Xie. *International Journal of Production Research* 53.17, 5224-5240 (2015)
11. J. Nithiyantham, P. Nirmal Kumar. *Journal of Systems Architecture* 60.1, 119-1390 (2014)
12. L. Hao, et al. *Computers in Industry* 63.4,298-308 (2012)
13. G. Wei, G. Liu, L. Zhang. *Journal of Hefei University of Technology (Natural Science)* **10** (2010)
14. G. ZHANG, B. GONG J. LI, W. Baozhong C. Yuanzhuo. *Modular Machine Tool & Automatic Manufacturing Technique* ,9 (2004)
15. K. Dervis, and B. Basturk. *Journal of global optimization*,39.3, 459-471 (2007)
16. K. Dervis, and B. Akay. *Applied mathematics and computation* ,214.1, 108-132 (2009)
17. D. Kalyanmoy, et al. *IEEE transactions on evolutionary computation*,6.2, 182-197 (2002)
18. K. Dervis, et al. *Artificial Intelligence Review* 42.1, 21-57 (2014)
19. G. Matthew G. *Doctoral dissertation, Virginia Tech* (2004)