Academic Journal of Engineering and Technology Science
ISSN 2616-5767 Vol.5, Issue 1: 4-9, DOI: 10.25236/AJETS.2022.050102

Research on the Selection and Matching of Complex Mechanical Products Based on Genetic Algorithms

Xu Wei

Zaozhuang University, Zaozhuang, Shandong, 266700, China

Abstract: *Assembly, the key to the entire product production, has a direct impact on product quality. The parts of various products are optimized and matched according to the prescribed rules by using computer technology, and the best matching scheme is finally obtained. This method can effectively ensure the assembly accuracy of the parts and reduce the residual rate of the parts. This paper proposes an optimized grouping and matching method based on genetic algorithm for the selection of complex mechanical products, and redefines the structure of dyeing, and provides technical support for the intelligent development of mechanical manufacturing.*

Keywords: *Selective Assembly, Surplus Parts, Genetic Algorithm*

1. Introduction

In the manufacture of mechanical products, the selective assembly method is usually used to meet the requirements of high assembly accuracy, and suitable parts are selected for assembly from parts with low machining accuracy [1]. The traditional method generally adopts the strategy of random selection and sequential assembly, and the deviation of the parts will accumulate in the direction of the dimensional chain, resulting in lower assembly quality. Selective assembly can solve this problem by offsetting the dimensional deviation of the parts and improve the quality of the assembly. For more complex selective assembly problems, computer-aided assembly has become a hot issue in current research. This paper studies the problem of group selection and assembly of complex parts based on genetic algorithm. The algorithm takes obtaining the optimal assembly combination and reducing the remaining parts as the optimization goal. Differently,this method takes a new grouping mode,and constructs a new chromosome structure to make full use of the grouping data information of the parts. After researching and analyzing the control parameters of the algorithm, the fitness function is improved for more complex assembly problems, so that it can solve multiple dimensional chain problems.The method is verified by an example of the assembly of an automobile engine cylinder consisting of three parts and three dimensional chains in different directions.this study has more practical reference significance for actual production.

2. Genetic algorithm of single dimensional chain

2.1 Description of single dimension chain problem

In this paper, the selective assembly of ball bearings in a complex assembly (more than two mating parts) is the object of analysis and research. The ball bearing is composed of three parts: inner ring, ball and outer ring. These parts， processed by different processing machines and processing procedures, have different size deviations, and the processing size deviations do not conform to the normal distribution. Therefore, the problem to be solved is how to assemble a component that meets the design requirements of dimensional accuracy through three parts with lower dimensional accuracy, so as to improve the assembly success rate and reduce the remaining parts.

2.2 Initialization of genetic algorithm

Since the distribution of part size deviations does not always obey the normal distribution, a large number of remaining parts will still be produced according to the above method. This article proposes a new chromosome encoding method, see formula (1).

$$\text{Chro}_{\,i} = \begin{bmatrix} 2 & 4 & 1 & \cdots & 2 \\ 5 & 3 & 2 & \cdots & 4 \\ 6 & 5 & 4 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & 2 & \cdots & 1 \end{bmatrix}_{m\times n} \tag{1}$$

Among them, *Chro* $_i$is the i-th chromosome;n is he number of parts in each batch;m The types of parts that make up the assembly;$Chro_{i[3][2]}$ is the number of groups the part is selected from. This means that the third part of the second assembly comes from the fifth group of the third part.

Since randomly generated individuals are not guaranteed to be feasible, in order to ensure the feasibility of the solution, the structure of the chromosome needs to meet the following constraints, see formula (2).

$$\sum_{b=1}^{n} (\text{ Chro }_i[a][b] == k) = G_{a,k}; a = 1,2\cdots m; k = 1,2\cdots \tag{2}$$

Among them, the [a][b] subscript represents the ath row and the bth column of the chromosome;The k-th group of k part, k is taken from 1 to the number of groups of the part in turn; $G_{a,k}$is a the number of parts contained in the kth group of parts; Chro $_i[a][b] == k$ judgment expression. If the value of Chro $_i[a][b]$is equal to k, then the value of the expression is 1, otherwise the value of the expression is 0.

The initial population is generated in a random manner. The random generation method can ensure that the initial solutions are evenly distributed in the solution space, expand the search range of the genetic algorithm, and increase the possibility of searching for the optimal solution. In order to ensure that the initial solution meets the constraints defined by formula (2), the initial solution is generated through the following three steps:Step 1: Generate a random integer sequence from 1 to 50 (the number of parts in each batch) ;Step 2: Insert the grouping data of the part according to the random integer sequence generated in the first step ;Step 3: Repeat step 1 to generate a random number sequence, and then enter step 2 to mark the next part grouping data.

Repeat the above process until all the parts are marked, at this time the coding of a chromosome is completed.According to the above steps, twenty sets of possible fitting assembly combinations are generated as the initial population.

2.3 Fitness function of genetic algorithm

The fitness function in this article aims to improve the success rate of assembly, that is, to increase the number of products that can be assembled to meet the design accuracy requirements, which is more in line with the actual requirements of the enterprise. The calculation method of the dimensional deviation of the assembly is given by the formula (3), and the calculation method of the closed loop is adopted. Formula (4) is a function of the success rate of assembly.

$$\begin{aligned} &T_{min} = X_{1min} + X_{2min} + \cdots + X_{k-1min} - X_{kmax} - X_{k+1max} - \cdots - X_{mmax} \\ &T_{max} = X_{1max} + X_{2max} + \cdots + X_{k-1max} - X_{kmin} - X_{k+1min} - \cdots - X_{mmin} \\ &t_\sigma = [T_{min}, T_{max}] \end{aligned} \tag{3}$$

Among them, t_σ is the upper and lower limits of the dimensional deviation of the assembly. 1 ~ k - 1 are increasing loops, and k~ m are decreasing loops. X$_{min}$ is the smallest size in a group, X*max* is the largest size.

$$f_{i,x} = \begin{cases} 0 & t_\sigma \not\subset T_\sigma \\ 1 & t_\sigma \subset T_\sigma \\ p & t_\sigma \infty T_\sigma \end{cases} \tag{4}$$

After grouping, the parts in each group can be approximately considered to obey the average distribution, so p can be calculated by formula (5).

$$p = \frac{t_\sigma \cap T_\sigma}{t}; 0 < p < 1 \tag{5}$$

Formula (6) gives the calculation formula of individual fitness value. The fitness value of an individual is the sum of the assembly success rate of each assembly in the chromosome divided by the number of parts in each batch.

$$\text{Fit}_{\,i} = \frac{\sum_{x=1}^{N} f_{i,x}}{N} \tag{6}$$

2.4 Generation of offspring of genetic algorithm

Use the sorting method to compare the proportion method to select the operator[2].

Step 1: Evaluate each individual in the population according to the fitness function to obtain the individual fitness value.The individuals are sorted in ascending order according to the size of the fitness value. The lower the individual, the higher the fitness value and the greater the possibility of being selected.Here, suppose the size of the population is N.

$$\begin{gathered} p_i = \frac{i}{\sum_{i=1}^{N} i} = \frac{2i}{N(N+1)} \\ F_i = \sum_{i=1}^{I} p_i = \frac{i(i+1)}{N(N+1)} \end{gathered} \tag{7}$$

Step 2:A random number r is generated through the random number generation function, and the generated random number r is greater than 0 and less than 1. If the i-th individual is selected, then formula (8) needs to be satisfied.

$$F_{i-1} < r \leq F_i \tag{8}$$

Because i is an integer, the number i of the selected individual can be calculated by formula (9).

$$i = \mathrm{ceil}\left(\sqrt{2 \times \mathrm{ceil}\left(\left(\frac{1}{2} \times (N+1) \times N\right) \times r\right) + \frac{1}{4}} - \frac{1}{2}\right) \tag{9}$$

In order to satisfy the constraints defined by formula (2) during the crossover process and ensure the feasibility of the solution, it is necessary to adjust the chromosomes during the crossover operation. The specific method is as follows:

Step 1: Make a copy of the gene before the crossover position of the selected two parents (denoted as I and II respectively), respectively (denoted as Sub_I and Sub_II respectively)[3].It is used to record the gene data after the cross bit on the original substring.

Step 2: Compare the locus in the paternal body II with the locus in Sub_I from the last bit forward, and if they are the same, remove the locus from the paternal body II.In the end, there are only II-Sub_I loci left in the parent body II. After splicing the remaining loci to Sub_I, a new substring I is obtained.The parent I uses the same operation to get the new substring II.

In order to maintain the diversity of the population and effectively prevent the algorithm from prematurely or converging to the local optimal solution, mutation processing is adopted.

Step 1: In each substring, for each gene position, first generate a random number, the range of which is between 0 and 1.If the random number is less than the mutation probability pm, then the mutation operation is performed on the locus. Otherwise, the gene locus will not be mutated and enter the judgment of the next gene locus.

Step 2: Find the nearest locus that differs from the gene value of the variant locus, and exchange the gene values of these two loci. When looking for an exchange locus, the gene before the variant locus is searched first.

2.5 Termination condition and result output of genetic algorithm

In this paper, the maximum number of iterations is set to 1000 generations. When the maximum number of iterations of the algorithm reaches this set value, the operation is terminated and the searched optimization results are output. The final output results of the algorithm proposed in this paper are: the optimal assembly combination searched, see Table 1.

Table1: Optimal assembly combination of output

Components	Assembly combination
A	5,2,5,5,4,4,3,3,4,5,5,5,3,3,5,3,5,5,5,3,4,5,5,4,4,3,3,3,5,6,5,5,6,4,4,7,2,1,6,4,2,4,2
B	4,3,3,3,3,3,3,4,3,3,3,3,2,3,3,3,3,4,5,3,2,2,2,2,3,4,4,4,4,3,5,4,2,2,2,4,5,3,1,6,4,3,2
C	1,1,1,1,2,2,2,1,1,2,2,2,2,2,2,1,1,1,2,3,1,2,1,2,2,2,2,2,3,3,2,2,1,1,1,2,2,2,2,1,1,1,2

3. Analysis of genetic algorithm control parameters

3.1 The effect of the number of groups on the results

In the traditional grouping selection assembly method,regardless of the size of the part's dimensional deviation zone and the gap accuracy required by the assembly, all parts are uniformly divided into 6 groups according to the process capability, and the dimensional accuracy of the processed parts is assumed to obey the normal distribution. However, in actual situations, the dimensional accuracy of parts is often non-normally distributed. For these parts, the number of groups can be appropriately reduced to reduce the complexity of calculation and the workload of grouping. In this experiment, fix the grouping number of two of the three parts, change the grouping number of the remaining part, and calculate the success rate of assembly through the group selection genetic algorithm. After grouping, the tolerance ranges of the groups of the three parts are similar, all around 2 microns.

3.2 The impact of population size on results

In the experiment, the crossover probability pc and mutation probability pm are set to 0.8 and 0.05, respectively. For each different population number, the program runs 10 times at a time to find the average success rate and average running time. The results are shown in Table 2.

Table 2: Average success rate and average running time under different population sizes

Item	Results					
Population size	10	20	30	40	60	80
Success rate (%)	75.3	78.6	78.5	78.7	78.7	79.1
Running time (s)	17	23	32	52	108	184

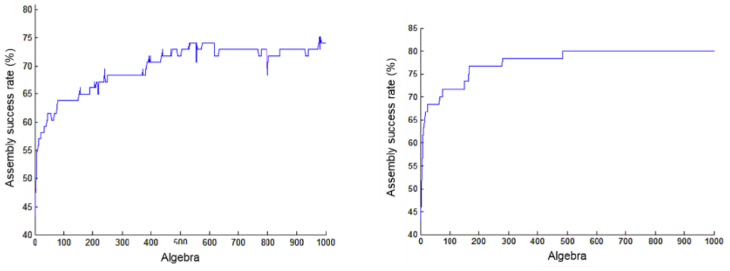
It can be seen from Table 2 that with the increase of the initial population, the average assembly success rate after 10 program runs shows an upward trend. When n=20, the optimization success rate can already meet the requirements, and the calculation takes less time.

3.3 The impact of crossover probability and mutation probability on the results

The selection of crossover probability and mutation probability in genetic algorithm is related to the performance of the algorithm. This paper uses Adaptive GA, the crossover probability p_c and mutation probability p_m can automatically change with the size of the individual fitness value in the population[4]. While enriching the diversity of the population, it ensures that the algorithm can eventually converge to the optimal solution.

3.4 The impact of elite retention strategy on results

In genetic algorithm, if the optimal individual that has evolved to the current search is not protected, it is likely that the structure of the optimal individual will be destroyed and lost in the next iteration. This phenomenon will occur repeatedly in the evolution of genetic algorithms, and the global convergence of the algorithm cannot be guaranteed. In order to avoid this phenomenon, the "elite retention strategy" was adopted[5]. Figure 1 (A) and Figure 1 (B) are the algorithm optimization curves before and after the elite retention strategy is adopted. The curves represent the optimal fitness value in the current generation. As shown in the figure, the algorithm optimization curve without elite retention strategy does not completely rise. The curve in the graph has up and down fluctuations. At the downward fluctuations, the optimal individual is lost, and the final convergence result is not satisfactory. However, the optimization curve that adopts the elite retention strategy shows a strict upward trend because the optimal individual is protected without loss and damage, and there is no upward and downward fluctuations, and the final convergence result is better.



(A)No elite retention strategy (B) Adopt an elite retention strategy

Figure1 : Algorithm optimization curve

4. Multi-dimensional chain genetic algorithm

4.1 Description of multi-dimensional chain problem

This research uses automobile engine assembly as an example to verify the improved algorithm. Table 3 is a table of related information about the quality characteristics and dimensions of each part in the assembly of automobile engine pistons and cylinders.

Table 3: Components quality characteristic size

Components	No.	Characteristic Name	Symbol	Size deviation (μm)
piston	1	Piston groove diameter	A	18
	2	Piston diameter	B	12
	3	Piston groove thickness	C	12
Piston ring	4	Piston ring width	D	18
	5	Piston ring thickness	E	6
Cylinder	6	Cylinder inner diameter	F	24

4.2 Grouping of genetic algorithms for multiple dimension chains

According to the size deviation of the parts and different assembly accuracy requirements, they are divided into different numbers of parts groupings. In order to facilitate the realization of coding, when programming the genetic algorithm, select the total number of groups to code, the total number of groups and the number of groups are coded as shown in the formula10.

$$X = \sum_{i=1}^{n-1} \left((x_i - 1) \times \prod_{j=i+1}^{n} a_j\right) + x_n \tag{10}$$

Among them, *n* is the number of dimensional features of parts, that is, the number of dimensions of the array of parts group coding ;X,Total group label after conversion ;a_j ,j-th dimension feature group number ;x_i ,The label of the i-th dimension of the original encoding method.

4.3 Fitness function of genetic algorithm with multiple dimension chains

In the assembly of automobile engines, the assembly accuracy requirements on the three dimensional chains need to be met. As long as the assembly deviation of any one size cannot meet the accuracy requirements, then the assembly of this part is unsuccessful.For each dimensional chain, the assembly success rate can be calculated by referring to formula (11).Then, add up the success probabilities of all assemblies and divide by the number of parts in each batch N to get the total success rate, formula (12) is the fitness function.

$$p(AB) = p(A) \times p(B) \tag{11}$$

$$Fit_i = \frac{\sum_{x=1}^{N} p_{i,x}}{N} \tag{12}$$

Among them, $p_{i,x}$ is the probability of successful assembly of the *x*-th assembly in the *i*-th individual.

4.4 The result of genetic algorithm for multiple dimension chains

The improved genetic algorithm is applied to the selection and assembly of engine cylinders. Table 4 is the optimal assembly combination for the final output of the algorithm optimization. In actual assembly, the engine cylinder parts can be assembled according to the assembly combination given in the table.

Table 4: Optimal assembly combination

Results
9,3,5,9,9,7,10,5,6,10,1,7,11,6,9,5,7,7,11,7,8,5,10,4,11,2,2,11,6,11,3,8,10,5,4,4,4,11,10,8,86,4,4,2,4,3,2,7
1,5,1,1,1,5,0,2,2,5,5,5,2,5,2,5,5,4,5,2,6,3,6,6,5,2,1,5,2,2,3,6,2,6,3,2,2,4,6,6,3,4,3,6,3,5,5,6,6,6,6,3,2,4,6,3
2,5,1,2,1,5,2,4,2,4,4,4,4,5,5,6,5,4,4,4,14,2,2,6,1,3,3,4,4,3,5,6,3,6,4,3,5,5,3,3,3,3,3,3,4,5,5,3,32,2,2,3,3,3,3

References

[1] Nagarajan L, Mahalingam SK, Kandasamy J , et al. A novel approach in selective assembly with an arbitrary distribution to minimize clearance variation using evolutionary algorithms: a comparative study[J]. Journal of Intelligent Manufacturing, 2021(11).
[2] Arram A , Ayob M . A novel Multi-parent order crossover in Genetic Algorithm for combinatorial optimization problems[J]. Computers & Industrial Engineering, 2019, 133(JUL.):267-274.
[3] H Zhao, Zhan Z H , Chen W N , et al. An Improved Selection Operator for Multi-objective Optimization[J]. 2019.
[4] Gao S, Zhang R. Rapid calibration method of MEMS accelerometer based on adaptive GA[J]. Journal of Beijing University of Aeronautics and Astronautics, 2019.
[5] Luo F, Fanglin L, Hou Z. An Improved Genetic Algorithm Based on Elite Retention Strategy and Explosion Operators[J]. Journal of Xihua University(Natural Science Edition), 2018.