



# Article Many-Objective Optimization and Decision-Making Method for Selective Assembly of Complex Mechanical Products Based on Improved NSGA-III and VIKOR

Rongshun Pan <sup>1</sup>, Jiahao Yu <sup>1</sup>, and Yongman Zhao <sup>1,2,\*</sup>

- <sup>1</sup> Department of Industrial Engineering, College of Mechanical and Electrical, Shihezi University, Shihezi 832003, China; panrongshun@stu.shzu.edu.cn (R.P.); 20202009046@stu.shzu.edu.cn (J.Y.)
- <sup>2</sup> Department of Data Science and Big Data Technology, College of Information Science and Technology, Shihezi University, Shihezi 832003, China
- \* Correspondence: zhrym@shzu.edu.cn

Abstract: In Industry 4.0, data are sensed and merged to drive intelligent systems. This research focuses on the optimization of selective assembly of complex mechanical products (CMPs) under intelligent system environment conditions. For the batch assembly of CMPs, it is difficult to obtain the best combinations of components from combinations for simultaneous optimization of success rate and multiple assembly quality. Hence, the Taguchi quality loss function was used to quantitatively evaluate each assembly quality and the assembly success rate is combined to establish a many-objective optimization model. The crossover and mutation operators were improved to enhance the ability of NSGA-III to obtain high-quality solution set and jump out of a local optimal solution, and the Pareto optimal solution set was obtained accordingly. Finally, considering the production mode of Human–Machine Intelligent System interaction, the optimal compromise solution is obtained by using fuzzy theory, entropy theory and the VIKOR method. The results show that this work has obvious advantages in improving the quality of batch selective assembly of CMPs and assembly success rate and gives a sorting selection strategy for non-dominated selective assembly schemes while taking into account the group benefit and individual regret.

Keywords: selective assembly; Taguchi quality loss; many-objective optimization; NSGA-III; VIKOR

## 1. Introduction

In the manufacturing environment of Industry 4.0, it becomes easy to accurately collect various data during production. Relying on a large amount of data, intelligent systems can optimize and make decisions in the production process. Most mechanical products are manufactured by the processing and assembly of components [1]. Under the condition of high requirements for product matching accuracy, due to the limitation of component processing capacity and manufacturing cost, it is unrealistic and uneconomical to completely rely on improving the accuracy of the processing process to meet and improve the product matching accuracy [2]. In practice, the demand of producers is to reduce manufacturing costs as much as possible while ensuring quality to obtain maximum product competitiveness. Therefore, selective assembly is one of the feasible methods to achieve high-precision assembly and reduce costs by using data and intelligent systems.

Researchers have made a lot of progress in selective assembly by researching the grouping of assembly components [3,4]. However, due to the many limitations of these methods, researchers have conducted further research on selective assembly methods from other perspectives. With the development of research, some intelligent optimization algorithms with better optimization effects have been used in the selective assembly field. Kannan et al. [5] proposed a genetic algorithm for selective assembly. In this research, the selective assembly process was analyzed, and a better combination was selected through the use of genetic algorithms to minimize assembly differences. Kannan et al. [6] used a



Citation: Pan, R.; Yu, J.; Zhao, Y. Many-Objective Optimization and Decision-Making Method for Selective Assembly of Complex Mechanical Products Based on Improved NSGA-III and VIKOR. *Processes* 2022, 10, 34. https:// doi.org/10.3390/pr10010034

Academic Editor: Arkadiusz Gola

Received: 7 December 2021 Accepted: 21 December 2021 Published: 24 December 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). genetic algorithm to obtain the best combination of the selection group with the smallest gap within the gap specification and at the same time minimized the deviation from the objective value by minimizing the Taguchi loss function. Fei et al. [7] proposed a new grouping method and chromosome structure using genetic algorithms to minimize the remaining components. Tan and Wu [8] summarized the research results of selective assembly and expanded the concept of selective assembly, distinguishing between direct selective assembly, using information directly obtained from measurements on component characteristics, and fixed bin selective assembly using components sorted into bins. The branch-demarcation algorithm is used to solve the selective assembly problem. Lu and Fei [9] proposed a new grouping method using a genetic algorithm to minimize the remaining components of a multidimensional chain and improved the genetic algorithm chromosome structure into a two-dimensional structure, making it more competitive in solving multidimensional chain problems. Manickam and De [10] proposed an optimal size selective assembly method, which is suitable for small batch quantities with wide tolerances, and successfully applied genetic algorithms in some components of the launch vehicle. Liu and Liu [11] proposed a method to determine the number of assembly groups for remanufactured engines. By improving the previous assembly method of specifically determining the number of groups, the number and interval of the components to be assembled are variable to reduce the waste of components. Wang [12] proposed the maximum assembly acceptance method under the condition of a non-normal distribution in matching components and optimized it by a genetic algorithm based on numerical experiments under different conditions. Kannan et al. [13] proposed a new selective assembly model to obtain specified tolerances in high-precision assembly and achieved a high success rate using genetic algorithms. Chu et al. [14] proposed an improved genetic algorithm to conduct application research on the selective assembly process of rotor vector (RV) reducers and verified the feasibility and effectiveness of the method. Jeevanantham et al. [15] studied the accumulation and extension of geometric and dimensional tolerances (GDTs) using tolerance analysis. By integrating GDT, the size deviation of components is grouped, and the optimal combination of component selective assembly is obtained by a genetic algorithm to obtain the minimum assembly deviation.

Kannan et al. [16] used the particle swarm optimization algorithm (PSO) to obtain the selective grouping and combination of component assembly, which significantly reduced the assembly deviation. Raj et al. [17] used the PSO algorithm to develop a code suitable for solving selective assembly problems and assembled components in batches to improve assembly efficiency. Rajesh et al. [18] studied the grouping selective assembly process, established the assembly loss model by using the Taguchi quality loss function, and obtained the combination of assembly losses that were as small as possible by using an improved immune algorithm. Rajesh et al. [19] proposed the Taguchi quality loss function based on symmetric intervals, studied the grouping selective assembly process, and used the improved sheep flock heredity algorithm to minimize the assembly clearance variation. Asha et al. [20] used a non-dominated sorting genetic algorithm (NSGA) to obtain the best combination of the selection group with the smallest change in the assembly clearance between the piston and the cylinder under the consideration of multiple characteristics. Although the multiple-characteristics-related situation is not considered in the assembly optimization process, it is of guiding significance for the selection of the processing and manufacturing process of the components. Raj et al. [21] applied the NSGA-II to batch selective assembly and applied it to complex components composed of pistons, piston rings, and cylinders. They proved that the algorithm is superior to existing methods in the literature in generating solutions with no remaining components and minimal gap changes. Researchers have applied NSGA in selective assembly and have begun to pay attention to solving the many-objective optimization problem in selective assembly.

Some of the latest research work has led to further development of selective assembly. Sun [22] applied the Taguchi theory to group and match the three-dimensional uniform transformation deviation, and the overall assembly quality was improved. Xing et al. [23], considering that the geometric error of the mating surface is an important factor affecting assembly quality, proposed an optimization method of shaft hole selective assembly based on relative entropy and dynamic programming, providing a new idea for the selective assembly of multiple batches of precision components. Rezaei Aderiani et al. [24] used digital twin technology to apply selective assembly methods to the assembly of automotive sheet metal components, which significantly improved the geometric variation and average deviation of the components. This research clearly discusses the difference between sheet metal assembly and linear assembly and notes that the selective assembly problem should be a many-objective optimization problem. Rezaei Aderiani et al. [25] improved the genetic coding method of intelligent optimization algorithms for solving selective assembly problems and proved in three cases that the effect of intelligent algorithms was better than mixed integer linear programming and mixed integer nonlinear programming in selective assembly, and this method could effectively reduce the calculation time. A methodology has been proposed in [26,27], which defines system requirements formally and automatically verifies and evaluates the system through simulation. Based on Formal Requirements Modeling Language and the Modelica Language, it is verified in two practical engineering project cases. This provides a theoretical basis and feasibility for formal confirmation of system requirements and simulation verification.

In engineering practice, the assembly of CMPs involves a multidimensional assembly dimension chain formed by multiple component sizes. From the literature survey, most of the above works focus on the improvement of selective assembly scheme of single dimension chain, less work has been done to model and research the quality loss of CMPs with a multidimensional assembly dimension chain, and there are certain limitations:

- In most of works, the optimization objective in the selective assembly process is relatively single. The assembly quality loss simply sums up the quality loss of each separate assembly dimension chain without considering the independent quality of each dimension chain;
- (2) There have been very few previous studies considering the production mode of Human–Machine Intelligent System interaction and could not order and make decisions on the non-dominated solution set.

Therefore, this study aims to solve the problem of selective assembly in the manufacturing process of complex mechanical products, including quality optimization, cost reduction, and scheme decision making.

The proposed problem can be solved in three stages. Firstly, the selective assembly process of CMPs is described, and then this mathematical model is established using the success rate and the Taguchi quality loss function. Secondly, taking the overall assembly success rate and the quality loss of the closed loop of each single assembly dimension chain as multiple optimization objectives, improved NSGA-III (NSGA-III-I) is used to solve the simulation data to obtain the Pareto optimal solutions. Finally, considering the production mode of the human–machine intelligent system interaction, fuzzy theory, entropy theory, and the VIKOR method are used to obtain the optimal compromise scheme for selective assembly.

The rest of the paper is organized as follows. In Section 2, the mathematical model and optimization objectives of selective assembly problem for CMPs are introduced. Section 3 presents a case of selective assembly and the assembly of pistons, piston rings, and cylinders. The proposed NSGA-III-I is described in Section 4. In particular, we introduced specific coding, crossover, and mutation strategies to solve the problem of selective assembly. In Section 5, the decision-making method combining fuzzy theory, entropy theory, and VIKOR is described. In Section 6, the NSGA-III-I algorithm is used to optimize the case selective assembly problem and achieved better results compared with other algorithms, and the optimal scheme was obtained successfully from the set of non-dominated solutions by using the proposed decision-making method. Finally, in the last section, conclusions are drawn.

# 2. Mathematical Model for Selective Assembly of CMPs

General CMP assembly has the following characteristics: Different quality characteristics (QCs) of the same component constitute different dimension chain closed loops (DCCLs), the same quality characteristics of the same component constitutes a plurality of different DCCLs, and a CMP is composed of many different types and quantities of components [28]. Based on the relationship between DCCLs and QCs, correlation analysis and constraint conditions were established. The optimization mathematical model of the selective assembly of CMPs was established through the assembly success rate and the Taguchi quality loss function.

#### 2.1. Correlation Analysis of Components to Be Assembled

In the process of mechanical product assembly, the dimension chain in the assembly link refers to DCCLs composed of QCs from different components in a specific order. DCCLs were used to indicate assembly accuracy during assembly. Therefore, the value of DCCLs represents the actual value of quality requirements (QRs) of the assembly. The correlation between QCs and QRs is the mathematical representation of the assembly structure, and the assembly structure is the basis of selective assembly. The assembly structure relationship of CMPs is shown in Figure 1.



Figure 1. Assembly structure relationship of CMPs.

The assembly structure of CMPs is analyzed according to the following steps:

Step 1: CMPs are split into individual component, and QC of each component is recorded.

Step 2: Various assembly relations of CMP are found and correspond to QCs of components.

Step 3: Formally express the assembly structure by polychromatic set theory and establish the assembly structure drawing. In this work, multiple dashed block diagrams are used to replace different colors.

# 2.2. Constraints in Assembly Process

Assembly quality and function of CMPs are determined by the value of each QR. When any QR-value exceeds the upper limit of the objective interval or is lower than the lower limit of the objective interval, the product assembly fails. If all QR-values are within the objective interval, the assembly is said to be qualified. Accordingly, the constraint matrix of QRs is constructed:

$$Y = \begin{bmatrix} y_{\min,1} & y_{\min,2} & y_{\min,3} & y_{\min,4} & \cdots & y_{\min,n} \\ y_{max,1} & y_{max,2} & y_{max,3} & y_{max,4} & \cdots & y_{max,n} \end{bmatrix}$$
(1)

where  $y_{\min,n}$  and  $y_{max,n}$  denote the lower and upper limits of the design interval value of the *n*th QR, respectively.

Each element in the vector y is the actual value of each QR of the assembly after assembly. When each element in the vector y is within the scope of the upper and lower limits, the assembly is successful. The vector y is as follows:

 $y = \begin{bmatrix} y_1 & y_2 & y_3 & y_4 & \dots & y_n \end{bmatrix}$ (2)

### 2.3. Taguchi Quality Loss Function

In the process of processing, due to the existence of various random factors, the processed components cannot be completely consistent, and the actual value of the component size has a certain deviation compared with the ideal value. When the deviation is within the acceptable interval, it is called the dimensional tolerance of the component, and the dimensional tolerance of the component determines the QR value of the assembly.

The traditional basis for judging whether a product is qualified is to judge whether the actual QR-value meets the design interval value. As long as it meets the design interval value, the product is considered qualified. Only products that exceed the design interval value are reworked, repaired, or scrapped, and quality loss occurs; according to Taguchi's loss function approach [29], even if it is a qualified product (the actual QR-value is within the design value interval), the fluctuation of the QR-value can still cause losses to users and society. The farther the QR-value is from the ideal value, the greater the loss will be. Therefore, the QR-value should be as close as possible to their ideal value. Two views of quality loss are shown in Figure 2.



**Figure 2.** Two views of quality loss: (a) Traditional view of quality loss; (b) Taguchi's quality loss approach.

Taguchi proposed three types of quality loss functions, loss functions for the largerthe-better, the smaller-the-better, the nominal-the-best characteristics, as follows:

Larger-the-Better : 
$$L(y) = k \frac{1}{y^2}$$
 (3)

Nominal-the-Best : 
$$L(y) = k(y - M)^2$$
 (4)

Smaller-the-Better : 
$$L(y) = ky^2$$
 (5)

where *L* denotes the Taguchi quality loss, *k* denotes a constant of proportionality independent of *y*, *y* is the actual value of QR, and *M* is the ideal value of QR.

In engineering practice, the nominal-the-best Taguchi quality loss model is often used for the quality loss of each QR of assembly, and it can be guaranteed that the QR-value falls within the design interval [18,30]. For the assembly meeting the design interval, when the QR-value is close to the ideal value, the quality loss caused by the deviation is 0. The quality loss will increase accordingly when the QR-value deviates from the ideal value. If the QR-value exceeds the objective interval value, the quality loss is equal to the cost of product treatment or rework. Therefore, the quality loss of QR is calculated by Formulas (6)–(9):

$$\lambda = k\Delta^2 \tag{6}$$

$$k = \frac{\lambda}{\Delta^2} \tag{7}$$

$$\Delta^2 = T^2[max(\alpha,\beta)] \tag{8}$$

$$L(y) = \begin{cases} \frac{\lambda}{T^2[max(\alpha,\beta)]} (y-M)^2, \ y \in [M-\alpha T] \cup [M+\beta T] \\ \lambda, \ y \in (-\infty, M-\alpha T) \cup (M+\beta T, +\infty) \end{cases}$$
(9)

where  $\Delta$  denotes the maximum difference between the tolerance boundary and the ideal value, *T* denotes the width of the design interval,  $\lambda$  denotes the quality loss of assembly failure, and  $\alpha$  and  $\beta$  denote the partitioning ratios of the values of the design interval,  $\alpha + \beta = 1$ . The functional image of the nominal-the-best Taguchi quality loss model in assembly is shown in Figure 3.



Figure 3. Functional image of the nominal-the-best Taguchi quality loss model.

#### 2.4. Many-Objective Optimization Model of Selective Assembly for CMPs

In the selective assembly process, a comprehensive and reasonable evaluation of the selective assembly scheme of CMPs requires comprehensive consideration of all quantifiable influencing factors. In selective assembly, the overall success rate and the degree of each QR close to the ideal value are the most representative evaluation indices of the scheme.

#### 2.4.1. Assessment Based on Assembly Success Rate

In the selective assembly of existing components to be assembled, more successful assemblies should be obtained as far as possible under the condition of meeting assembly requirements, and the surplus of mismatched components should be reduced. Assembly success rate is defined in the range [0, 1]. The assembly success rate in optional scheme *X* can be calculated by Formula (10):

$$\eta_s(X) = \frac{n_s(X)}{n_a(X)} \tag{10}$$

where  $n_a$  denotes the number of assemblies to be assembled;  $n_s$  denotes number of qualified assemblies obtained through selective assembly; *X* denotes the scheme for selective assembly; and  $\eta_s$  denotes the success rate of selective assembly.

#### 2.4.2. Assessment Based on Quality Loss of Individual QR

In previous studies, the multidimensional chain was often converted into a singledimensional chain when the quality loss of the assembly was modeled, and the simple linear addition of the quality loss was used to set an optimization objective. This will lead to product quality requirement confusion, and it is not easy to judge the effect of the assembly scheme. The optimization effect of product assembly is not ideal, and the performance of schemes with the same quality loss varies greatly. Based on the characteristics of general CMPs, the proposed work takes the average quality loss function of a single QR as the optimization objective function as calculated by Formulas (11) and (12):

$$L(y_i) = \frac{1}{q} \sum_{j=1}^{q} L(y_{ij})$$
(11)

$$L(y_{ij}) = \begin{cases} \frac{\lambda}{T_i^2[max(\alpha,\beta)]} (y_{ij} - M_i)^2, y_{ij} \in [M - \alpha T_i] \cup [M + \beta T_i] \\ \lambda, y_{ij} \in (-\infty, M_i - \alpha T_i) \cup (M_i + \beta T_i, +\infty) \end{cases}$$
(12)

where  $T_i$  denotes the width of the design interval for the *i*th QR,  $y_{ij}$  denotes the actual value of the *i*th QR of the *j*th assembly, and  $M_i$  denotes the objective value of the *i*th QR.  $L(y_{ij})$  denotes the quality loss of the *i*th QR of the *j*th assembly, and  $L(y_i)$  denotes the mean value of the sum of the quality loss of the *i*th QR of all assemblies in the selective assembly scheme; *q* denotes the number of components of the same type.

#### 2.4.3. Many-Objective Optimization Model of Selective Assembly for CMPs

The pros and cons of CMP batch selective assembly schemes are jointly evaluated by the overall assembly success rate and product quality loss. To unify the optimization direction of a set of objective functions to the minimum direction, the first objective function is deformed. From the formula in this chapter, the mathematical model (Formula (13)) of the many-objective optimization for selective assembly can be obtained as follows:

$$Min f(X) = \begin{bmatrix} f_1(X) & f_2(X) & f_3(X) & \cdots & f_m(X) \end{bmatrix}$$
s.t.
$$\begin{cases} f_1(X) = 1 - \eta_s(X) \\ f_2(X) = L(y_1) \\ f_3(X) = L(y_2) \\ & \cdots \\ f_m(X) = L(y_{m-1}) \\ X = \{x_1, x_2, \dots, x_p\} \\ x_1, x_2, \dots, x_p \in O_q \end{cases}$$
(13)

where *m* denotes the number of optimization objective functions, m = n + 1, *n* denotes the number of QR; f(X) denotes the objective vector,  $f_1(X)$ ,  $f_2(X)$ , ...,  $f_m(X)$  denote each element value in the vector, and *X* denotes the selective assembly scheme;  $x_p$  denotes the full permutation of the *p*th type of components;  $O_q$  denotes the domain of  $x_1, x_2, ..., x_p$ , which is the full permutation of 1 to *q*; *q* denote the number of components of the same type.

#### 3. Case Analysis

An example of the selective assembly of pistons and cylinders proposed by Asha [20] is used to verify the selective assembly of CMPs in this work. The assembly structure is shown in Figure 4.



Figure 4. Piston and cylinder assembly.

3.1. Components QCs and QRs of the Assembly

The assembly is composed of three components: a piston, a piston ring, and a cylinder. The considered QCs of the components are listed in Table 1.

$T_111_1 \cap C_1 = 1$		1		1.1
Table L. U. Sand	processing parameters	of each com	inonent in th	e assembiv
Tuble In QCD untu	processing parameters	or cuerr com	iponent in ut	e abbennory.

		Quality Characteristic				Manufacturing Details				
Mating - Component Code	Name Description Co	Description	Code Specification	Specification	Basic Dimension	Tolerance Achieved (µm)			Standard Deviation	
			•	(mm)	Min	Max	Mean	(σ)		
		Piston groove diameter	А	$42^{+0.000}_{-0.018}$	42.0	0	18	9	6	
C1	C1 Piston	Piston diameter	В	$50^{+0.012}_{-0.000}$	50.0	0	12	6	4	
	Piston groove thickness	С	$3.2^{+0.000}_{-0.012}$	03.2	0	12	6	4		
		Piston ring width	D	$4^{+0.000}_{-0.018}$	04.0	0	18	9	6	
C2	Piston ring	Piston ring thickness	Е	$3^{+0.000}_{-0.006}$	03.0	0	6	3	2	
		Piston ring outer diameter	F	$50\substack{+0.028 \\ -0.004}$	50.0	0	24	12	8	
C3	Cylinder	Cylinder inner diameter	G	$50\substack{+0.028\\-0.004}$	50.0	0	24	12	8	

The QCs of all the components of the piston and cylinder assembly are analyzed and QRs that need to be analyzed and calculated during the selective assembly process of the assembly are obtained, as shown in Figure 5.



Figure 5. Piston cylinder assembly structure and QRs.

Figure 5 shows four QRs:  $y_1$ ,  $y_2$ ,  $y_3$ , and  $y_4$ . The corresponding QCs are A-D-G, C-E, B-G, and F-G. This assembly selection process involves three components, seven QCs, and four QRs.



If the components are assembled through the principle of interchangeability, the tolerances of four QRs will be the sum of the tolerances of the corresponding component QCs, which is the maximum tolerance. Consider the center value of each design tolerance interval as the ideal value of each QR, and the design tolerance intervals of each QR of the assembly are listed in Table 2.

Table 2. Related data of each QR of the assembly.

Quality Requirements	Maximum Tolerance	Design Tolerance Interval	Ideal Value
<i>y</i> <sub>1</sub>	$0^{+0.082}_{+0.004}$	$0^{+0.062}_{+0.024}$	0.043
$y_2$	$0.2^{+0.006}_{-0.012}$	$0.2^{+0.002}_{-0.008}$	0.197
<i>y</i> <sub>3</sub>	$0^{+0.028}_{-0.008}$	$0^{+0.019}_{+0.001}$	0.010
$y_4$	$0^{+0.024}_{-0.024}$	$0^{+0.012}_{-0.012}$	0.000

According to the design tolerance interval of each QR of the assembly, the constraint matrix is obtained as follows:

 $Y = \begin{bmatrix} 0.062 & 2.002 & 0.019 & 0.012 \\ 0.024 & 1.992 & 0.001 & -0.012 \end{bmatrix}$ 

The ideal value vector of QRs is as follows:

 $M = \begin{bmatrix} 0.043 & 0.197 & 0.010 & 0 \end{bmatrix}$ 

The objective tolerance interval of QRs is as follows:

$$[M_i - 0.5T_i] \cup [M_i + 0.5T_i]$$

#### 3.2. Optimization Objectives of the Case Selective Assembly

According to the many-objective selective assembly optimization model established in this work and the assembly QRs of the piston and cylinder assembly in the case study, the optimization objectives in the selective assembly process of the piston and cylinder group are established as follows:

$$Min f(X) = \begin{bmatrix} f_1(X) & f_2(X) & f_3(X) & f_4(X) & f_5(X) \end{bmatrix}$$
  
s.t. 
$$\begin{cases} f_1(X) = 1 - \eta_s(X) \\ f_2(X) = L(y_1) \\ f_3(X) = L(y_2) \\ f_4(X) = L(y_3) \\ f_5(X) = L(y_4) \\ X = \{x_1, x_2, x_3\} \\ x_1, x_2, x_3 \in O_{50} \end{cases}$$

#### 3.3. Data Simulation of Each Component

In this work, the purpose of the proposed method was to maximize the success rate of assembly while reducing the quality loss of each quality requirement as much as possible. To verify the feasibility of the scheme, the Monte Carlo simulation method was used to obtain the QC data of each component. A total of 300 sets of components were simulated and randomly divided into 6 groups, each with 50 components. Simulating the component QC data in the actual production process through the discrete normal distribution probability mass function (Formula (14)) is defined as:

$$pmf(h) = \Pr(x - 0.5 < H \le x + 0.5) \text{ for } h \in \mathbb{Z}$$
 (14)

where *H* denotes the normal distribution of QCs of the component, *Z* denotes the set of integers, and the size table of the components is listed in Table A1.

#### 4. NSGA-III-I

The selective assembly model of the CMPs constructed by the above content was a many-objective optimization model with multiple objectives that needed to be optimized at the same time, and the optimization objectives were often greater than three. An NSGA-III-I selective assembly many-objective optimization algorithm suitable for this model was designed.

The NSGA-III is a new many-objective optimization algorithm proposed on the basis of the NSGA-II [31]. The NSGA-II selects individuals of the same non-dominant level by calculating the distance of individual crowding degrees. It was believed that the larger the crowding degree distance is, the better. This method was particularly suitable for solving the optimization problem of two objectives. However, when it solves optimization problems with more than three objectives, the individuals selected by crowding distance sorting are not uniformly distributed on the non-dominated layer, which easily falls into the local optimum, which affects the performance of the algorithm. The NSGA-III selects individuals based on the method of reference points and optimizes the selection by calculating the shortest distance between individuals and reference points. When facing many-objective optimization problems with three or more objectives, the algorithm can be ensured to converge to a uniformly distributed Pareto surface [32].

The NSGA-III algorithm is limited by the coding method and completely random initial solution generation rules, resulting in the algorithm performance being greatly dependent on the initial solution set, and the feasible solution space of the constructed selective assembly model increases exponentially with the increase in components. It greatly increases the risk of obtaining a local optimal solution. To improve the global searching ability of the NSGA-III, a new mapping method between genotype and phenotype was selected, and the idea of a simulated annealing algorithm was introduced into crossover and mutation operators to enhance the ability of the algorithm to obtain the global Pareto

# 4.1. Pareto Optimal Solution

front (PF).

The solution set finally obtained by the NSGA-III algorithm is the Pareto optimal solution set. When the optimization direction is the minimum value direction, the solution set is defined as follows:

For any two decision variables  $X_a$ ,  $X_b \in X_f$ , where  $X_f$  is the feasible solution set:

(1)  $\forall j = \{1, 2, \dots, k\}: f_j(X_a) \leq f_j(X_b);$ (2)  $\exists j = \{1, 2, \dots, k\}: f_j(X_a) < f_j(X_b).$ 

If and only if the above conditions are met,  $X_b$  is dominated by  $X_a$ . In the solution space, there is a solution set X, which is not dominated by any solution; it is the Pareto optimal solution [33].

# 4.2. Input Module

For the experimental problems considered in this paper, the required input data are as follows:

- Problem data:
  - (1) Batch size;
  - (2) Number of component types in the assembly;
  - (3) Dimensional data of components.
- Algorithm Parameters:
  - (1) Population size;
  - (2) Crossover probability;
  - (3) Mutation probability;
  - (4) Termination Index.

#### 4.3. Initializing Module

#### 4.3.1. Encoding

Selective assembly is a many-objective combinatorial optimization problem. The way to solve the problem is to find the best possible selective assembly scheme. Traditional binary coding will produce a large number of infeasible solutions when contending with selective assembly problems, which increases the difficulty of the algorithm in solving the problem. Kannan et al. [5] proposed a classic coding method to solve the selective assembly problem using genetic algorithms (Figure 6a). However, this kind of coding scheme still encounters failure in achieving a one-to-one correspondence between the selective assembly scheme and the chromosome. In the case of different genotypes, it can match the same selective assembly scheme, which will cause the performance of the algorithm to decrease. Therefore, Rezaei Aderiani et al. [25] proposed a new coding scheme that can ensure that the dyeing and selection scheme is one-to-one, and the dotted lines indicate that the genes are fixed and immutable, as shown in Figure 6b. Taking an assembly consisting of two components A and B as an example, five for each component, (A1B4), (A2B3), (A3B5), (A4B1), and (A5B2), this combination can be expressed as:



Figure 6. Encoding: (a) Kannan Encoding; (b) Rezaei Aderiani Encoding.

#### 4.3.2. Initialization of Algorithm Parameters

Initialize the population and randomly generate N chromosomes; Initialize the number of iterations, Gen = 0; Initialize to reference points.

#### 4.4. New Population Generation Module

The new population generation module was divided into a crossover module and a mutation module. In NSGA-III, the gene mutation method with fixed crossover and mutation probability was adopted. This strategy keeps the crossover and mutation probability unchanged during the operation of the algorithm and cannot consider the global search performance and convergence performance required by the algorithm. Aiming at the many-objective optimization problem in the CMP selective assembly, the adaptive change idea of a simulated annealing algorithm was used to improve crossover and mutation operators to enhance the searching ability and convergence of the algorithm.

#### 4.4.1. New Crossover Operator

Notably, when the crossover probability of the algorithm was small, it was beneficial to maintaining population diversity and increasing global search ability. In contrast, when the crossover probability was large, the efficiency of local search was strong, and the convergence ability of the algorithm was improved. Therefore, the crossover operator was improved according to the crossover probability characteristics of the algorithm. In particular, the crossover probability was small in the early stage of the algorithm operation to maintain population diversity, and the crossover probability was large in the late stage to improve the convergence ability of the algorithm. The crossover strategy (Formula (15)) is as follows:

$$ProC = 1 - p_{C1} \times \frac{p_{C2}^2}{p_{C2}^2 + Gen^2}$$
(15)

where *ProC* denotes the crossover probability,  $p_{C1}$  and  $p_{C2}$  denote the crossover parameters, and *Gen* denotes the number of iterations. In this work,  $p_{C1} = 0.6$  and  $p_{C2} = 50$  are selected.

Step 1: Select crossed chromosomes and generate a random number r for each pair of chromosomes, if r < ProC, corresponding chromosomes are selected for the crossover operation.

Step 2: The chromosomes selected for crossover are paired, and crossover operations are performed between the same substrings of different chromosomes. To avoid invalid chromosomes, the crossover point is placed between the second and thirds substrings, and a single point crossover is performed. An example of a two-chromosome crossing operation to produce two offspring chromosomes is shown in Figure 7.



#### Figure 7. Crossover operation.

## 4.4.2. New Mutation Operator

The mutation operator locally changes the genes of the chromosomes, hoping to create better chromosomes. When the algorithm mutation probability is too low, it will greatly

increase the risk of falling into the local optimum; when the algorithm mutation probability is too high, it will cause the algorithm solution set to fail to converge. Therefore, by improving the mutation strategy of the algorithm, this study makes the mutation probability relatively high at the beginning of the algorithm, increases the diversity and richness of the population, and gradually decreases with the increase in the number of iterations, making it easier to retain more excellent individuals and improving the convergence performance of the algorithm. The mutation strategy (Formula (16)) is as follows:

$$ProM = 1 - p_{m1} \times \frac{p_{m2}^2}{p_{m2}^2 + Gen^2}$$
(16)

where *ProM* denotes the mutation probability,  $p_{m1}$  and  $p_{m2}$  denote mutation parameters, and *Gen* denotes the number of iterations. In this work,  $p_{m1} = 0.6$  and  $p_{m2} = 50$  are selected.

Mutation operations are used to target genes on each chromosome. To avoid the occurrence of invalid chromosomes, the neighborhood swapping method is used to generate a random number r for each gene; if r < ProM during the mutation process, the selected gene is swapped with the previous gene. An example of the mutation operation is shown in Figure 8.



Figure 8. Mutation operation.

### 4.5. Merging of Parent and Offspring Populations

The combined population  $R_t$  with population size 2N was obtained by merging the parent population  $P_t$  with population size N and the offspring population  $Q_t$  with population size N.  $R_t$  is the transition population between population  $P_t$  and the next population  $P_{t+1}$ .

#### 4.6. Evaluation Module and Environment Selection Module

The fitness vector elements of each chromosome correspond to the five objective function values of  $f_1(X)$ ,  $f_2(X)$ ,  $f_3(X)$ ,  $f_4(X)$ , and  $f_5(X)$ . The five objective function values of N chromosomes are calculated.

#### 4.6.1. Selected Individuals

The transition group  $R_t$  with a population size of 2N is merged by the current group  $P_t$  and its progeny group  $Q_t$ . To make the size of  $P_{t+1}$  meet the requirements of population size and ensure that the overall performance is better than  $P_t$ , a non-dominated sorting operation is performed on  $R_t$ , and individuals are selected with higher non-dominated levels and added to  $P_{t+1}$  until the size of  $P_{t+1}$  cannot accommodate individuals at the next level of dominance, and this dominance level is recorded as  $F_t$ .

The associated reference point operations on individuals in the current population (excluding individuals in  $F_l$ ) are performed, the reference point *j* with the least associated individuals is selected and its associated number is recorded as  $\rho_j$ . Then, the situation of this reference point can be elaborated as follows:

- (1) If there is no individual associated with the reference point in  $F_l$ , a new reference point vector is selected.
- (2) If the number of population individuals associated with this reference point is 0  $(\rho_j = 0)$ , but there are individuals in  $F_l$  that are related to this reference point vector, then the individual with the smallest distance from it is found and extracted from  $F_l$ , which is then added it to the selected next-generation population, set  $\rho_i = \rho_i + 1$ .
- (3) If  $\rho_j > 0$ , there are multiple individuals in  $F_l$  that are associated with this reference point vector, and then individuals in  $F_l$  are randomly selected that are associated with the reference point until the population size is *N*.

#### 4.6.2. Associated Reference Point

According to the fitness vector of all individuals in the population, the minimum value of each objective function is calculated, the ideal point is updated, the hyperplane is updated, each individual is associated to the reference point and reference vector according to the strategy, and the number of individuals associated with the reference point is calculated. The specific association strategy is as follows:

Step 1 Generate hyperplane:

First, select the minimum value of each objective function in the current population, calculate the ideal point, take the ideal point as the origin and the objective function as the coordinate axis, and normalize the individual using Formula (17):

$$f'_{i}(X) = f_{i}(X) - z_{i}^{\min}$$
, for  $j = 1, 2, ... m$  (17)

where  $z_j^{\min}$  denotes the ideal point;  $f_j(X)$  denotes the *j*th fitness value of each individual in the population; and  $f'_j(X)$  denotes *j*th fitness value of each individual after normalization in the population.

Next, the achievement scalarizing function  $(ASF(X, w) = \max_{j=1}^{m} f'_j(X) / w_j)$  is used

to calculate the extreme points. An extreme point refers to the individual whose objective function value of one dimension is very large and the objective function value of the other dimensions is very small; then, the population is traversed, and the extreme points  $z_i^{max}$  of each dimension are found. Then, a hyperplane is constructed based on these extreme points, and the intersection of the hyperplane and the coordinate axis is the intercept  $a_i$ . Taking the three-object problem as an example, the hyperplane is obtained as shown in Figure 9.



Figure 9. Hyperplane: The three-objective optimization problem taken as an example.

In NSGA-III, it is necessary to obtain a hyperplane equidistant from the origin and normalize the individuals of the population according to the intercept of the coordinate axes so that a standard hyperplane can be obtained in each generation. After this transformation operation, the individuals of the population are normalized to the plane where the reference point is. The conversion Formula (18) is as follows:

$$f''_{j}(X) = \frac{f'_{j}(X)}{a_{j}}, \text{ for } j = 1, 2, \dots, m$$
 (18)

Step 2 Generate reference points:

This study adopted the reference point establishment method proposed by Das and Dennis [34]: The reference point is placed on a normalized m - 1 dimensional standard hyperplane, which is evenly inclined to all objective axes and has an intercept on each axis. *H* points are uniformly generated on the hyperplane, and the boundary crossing method is used to construct weights. If the objective dimension is divided into *p* equal parts, the total

number of reference points *H* in the *m*-objective problem is  $\binom{m+p-1}{n}$ .

Taking the three-objective problem as an example, the reference point is created in the triangle plane with (0, 0, 1), (0, 1, 0), and (1, 0, 0) as the vertices. If the coordinate axis is divided into four equal parts, p = 4,  $H = \begin{pmatrix} 3+4-1 \\ 4 \end{pmatrix} = 15$ . The distribution of reference points is shown in Figure 10.



**Figure 10.** Reference point distribution of the standard hyperplane for the three-objective optimization problem.

In this experiment, the problem to be solved is a five-objective problem, and the reference point is created in the hyperplane with (1, 0, 0, 0, 0), (0, 1, 0, 0, 0), (0, 0, 1, 0, 0), (0, 0, 1, 0, 0), (0, 0, 1, 0), (0, 0, 1, 0), (0, 0, 1, 0), (0, 0, 1, 0), (0, 0, 1, 0), (0, 0, 1, 0), (0, 0, 1, 0), (0, 0, 0, 1) as vertices.

Step 3 Associated reference point:

Each reference vector is constructed by connecting each reference point and the origin. For each individual in the population, the reference vector was traversed to find the reference point nearest to the individual, and the association information and distance were recorded. The association between the individual and the reference point was completed. The distance between the individual and the reference vector will be described by the Euclidean distance.

#### 4.7. Termination Criterion

When the number of iterations reaches the maximum number or other relevant setting conditions are met, the algorithm terminates. In the experiment in this paper, the maximum number of iterations of the algorithm is set to 500.

#### 4.8. Output Module

The output of the algorithm is the 500th generation population, and the population size is *N*. Decoding can obtain *N* non-dominated selective assembly schemes. The basic search process of the proposed algorithm is shown in Figure 11.



Figure 11. Flowchart of the proposed NSGA-III-I for batch selective assembly.

#### 5. VIKOR Method to Sort the Solution Set

The Pareto optimal solution set obtained by the NSGA-III-I algorithm solving the selective assembly high-dimensional many-objective optimization model is actually a non-dominated solution set, which includes a large number of mutually non-dominated

solutions. It is impossible to determine the priority of each scheme from the numerical relationship, so the schemes of selective assembly need to be sorted and selected by means of human-machine intelligent system interaction. The VIKOR method is a multi-criteria decision-making method proposed by Opricovic [35], which is an optimal compromise decision-making method based on ideal points, as well as an optimal compromise solution. It is developed from the  $L_{p-metric}$  distance measure function (Formula (19)):

$$L_{p-metric}(X_i) = \left\{ \sum_{j=1}^{m} \left[ \omega_j \left( f_j^* - f_{ij} \right) / \left( f_j^* - f_j^- \right) \right]^p \right\}^{1/p}$$
(19)

where  $X_i$  denotes the *i*th scheme; *p* denotes the distance parameter of the  $L_{p-metric}$  distance measure function;  $f_{ij}$  denotes the evaluation value of the *i*th scheme in the *j*th index;  $f_j^*$  and  $f_j^-$  denote the best and worst evaluation values of the *j*th index;  $\omega_j$  denotes the weight of the *j*th index.

#### 5.1. Calculation of the Weight of Each Indicator

In the decision-making process of human-machine intelligent system interaction, it is necessary to assign weight to each optimization objective, and the index weight is a measure reflecting the relative importance of each optimization objective. In this paper, the entropy weight method is used to calculate the objective weight of the index. The triangular fuzzy number is used to fuzzily quantify the engineering decision maker language variables, and then the subjective weight of the indicators is solved. Finally, the subjective weight and the objective weight are combined into a combination weight with high robustness.

### 5.1.1. Entropy Method to Determine the Objective Weight

The entropy method is an approach to obtain the weight [36]. According to the basic principles of information theory, information is a measure of the order degree of a system, and entropy is a measure of the disorder degree of a system. If the information entropy of the indicator is smaller, then the amount of information provided by the indicator is larger, and it should play a larger role in the comprehensive evaluation such that the weight should be higher.

According to the definition of information entropy in information theory, the information entropy of a set of data, and the objective weights of each indicator are calculated using Formulas (20)–(22):

$$E_{j} = -\frac{1}{\ln n} \sum_{i=1}^{n} p_{ij} \ln p_{ij}$$
(20)

$$p_{ij} = f_{ij} / \sum_{i=1}^{n} f_{ij}$$
(21)

$$\omega'_{j} = \frac{1 - E_{j}}{m - \sum E_{j}}, \ j = 1, 2, \dots, m$$
 (22)

where  $E_j$  denotes the information entropy,  $p_{ij}$  denotes the proportion of  $f_{ij}$  in the *j*th indicator data, *n* is the number of decision-making schemes, and  $f_{ij}$  denotes the value of the *j*th objective function in the *i*th scheme after normalization.  $\omega'_j$  denotes the objective weight of the *j*th indicator.

#### 5.1.2. Triangular Fuzzy Numbers to Determine Subjective Weights

1

Taking into account the uncertainty and ambiguity in the decision making process, a triangular fuzzy number was introduced to establish a subjective weight matrix [37]. The correspondence between linguistic variables and triangular fuzzy numbers is determined, as listed in Table 3:

The Importance of Each Criterion	Triangular Fuzzy Set
Very low	(0, 0, 0.2)
Low	(0, 0.1, 0.25)
Medium low	(0, 0.3, 0.45)
Medium	(0.25, 0.5, 0.65)
Medium high	(0.45, 0.7, 0.8)
High	(0.55, 0.9, 0.95)
Very high	(0.85, 1, 1)

Table 3. Mapping table of language variables and trigonometric fuzzy numbers.

Different engineering decision makers may have different views on each index. The fuzzy weight vector of decision makers  $B_k$  on the optimization objective is  $\omega''_{jk} = \left\{ (\omega''_{jk1}, \omega''_{jk2}, \omega''_{jk3}) | j = 1, 2, ..., m \right\}$ . Therefore, after gathering different decision makers to evaluate the indicators, each element of the subjective weight matrix is expressed using Equations (23) and (24):

$$\omega''_{j} = \left\{ (\omega''_{j1}, \omega''_{j2}, \omega''_{j3}) | j = 1, 2, ..., m \right\},$$

$$\left\{ \begin{array}{l} \omega''_{j1} = \min_{k} \left\{ \omega''_{jk1} \right\} \\ \omega''_{j2} = \frac{1}{K} \sum_{k=1}^{K} \left\{ \omega''_{jk2} \right\} \\ \omega''_{j3} = \max_{k} \left\{ \omega''_{jk3} \right\} \end{array} \right.$$
(23)

Then, the subjective weight of the indicator is obtained. Because each element of the subjective weight matrix is a triangular fuzzy number, the mean area method is used for defuzzification to obtain the subjective weight of each indicator (Equation (25)):

$$\omega''_{j} = \frac{(\omega''_{j1} + 2\omega''_{j2} + \omega''_{j3})/4}{\sum_{j=1}^{m} (\omega''_{j1} + 2\omega''_{j2} + \omega''_{j3})/4}$$
(25)

#### 5.1.3. Determination of the Combination Weight

In order to balance the impact of the subjective and objective factors on the result of the selective assembly scheme, multiplication and division were selected to determine the combined weight of the subjective weight and the objective weight. The calculation method of the combined weight uses Equation (26):

$$\omega_j = \frac{(\omega'_j)^{\gamma'} \cdot (\omega''_j)^{\gamma''}}{\sum_{j=1}^m \left[ (\omega'_j)^{\gamma'} \cdot (\omega''_j)^{\gamma''} \right]}$$
(26)

where  $\gamma'$  and  $\gamma''$  denote the proportion coefficients of the objective weight and subjective weight in the total weight, respectively. Generally,  $\gamma' + \gamma'' = 1$ . In this work,  $\gamma' = \gamma'' = 0.5$ .

# 5.2. VIKOR Method for Multicriteria Decision-Making

VIKOR's basic point of view is to fully consider the subjective preferences of decision makers and, at the same time, to trade-off the limited decision-making schemes by maximizing group benefits and minimizing individual regrets, making the decision results more reliable. The compromise solution  $F^c$  is the closest to the ideal solution  $F^*$ , which is the result of mutual concessions between the two criteria, as shown in Figure 12.



Figure 12. Ideal solution and compromise solutions in the VIKOR method.

The step-by-step procedure is given below:

- (1) Summarize the *m* fitness values of the *n* Pareto optimal solutions under the corresponding optimization objectives and normalize them. *n* is the number of Pareto optimal solutions in the final population, and *m* is the number of optimization objectives. In this experiment, n = 126, m = 5.
- (2) Establish a standard decision matrix  $\omega \Delta f_{ij}$  based on the weight matrix  $\omega$ ; positive and negative ideal solutions and critical evaluation values are determined. Positive ideal solution  $F^* = \{f_1^*, f_2^*, \dots, f_m^*\}$  and negative ideal solution  $F^- = \{f_1^-, f_2^-, \dots, f_m^-\}$ , and  $f_i^*$  and  $f_i^-$  are the best and worst evaluation values of the *j*-th index, respectively.
- (3) Calculate the group utility value  $S_i$  and individual regret value  $R_i$  of each Pareto optimal solution by using Equations (27) and (28):

$$S_i = \sum_{j=1}^n \omega_j (f_j^* - f_{ij}) / (f_j^* - f_j^-)$$
(27)

$$R_{i} = \max_{i} \omega_{j} \left( f_{j}^{*} - f_{ij} \right) / \left( f_{j}^{*} - f_{j}^{-} \right)$$
(28)

(4) Calculate the compromise value  $Q_i$  of each Pareto optimal solution scheme by using Equation (29):

$$Q_{i} = \frac{v \left(S_{i} - S_{\min}\right)}{\left(S_{max} - S_{\min}\right)} + \frac{\left(1 - v\right) \left(S_{i} - R_{\min}\right)}{\left(R_{max} - R_{\min}\right)}$$
(29)

where  $S_{max}$ ,  $S_{min}$ ,  $R_{max}$ , and  $R_{min}$  are the maximum and minimum values of  $S_i$  and  $R_i$ , respectively.

(5) Sort the Pareto optimal solution set *X* according to the increasing *Q<sub>i</sub>*-value: *X<sub>1</sub>, X<sub>2</sub>, ..., X<sub>i</sub>, X<sub>n</sub>*. If *X<sub>1</sub>* is the solution with the smallest *Q*-value and both Condition 1 and Condition 2 are satisfied, *X<sub>1</sub>* is the stable optimal solution.

Condition 1:  $Q(X_2) - Q(X_1) \ge 1/(n-1)$ ; Condition 2: After the solutions are sorted based on the *Q*-value, the *S*-value, or *R*-value of the first-ranked solution  $X_1$  is better than the second-ranked solution  $X_2$ .

If the above two conditions cannot be established at the same time, a compromise solution will be obtained, which can be divided into two cases:

- Case 1: If only Condition 2 is not satisfied, the compromise solution is  $X_1, X_2$ ;
- Case 2: As long as Condition 1 is not satisfied, the compromise solution is  $X_1, X_2, ..., X_i$ , where *i* is the maximized value determined by  $Q(X_i) Q(X_1) \ge 1/(n-1)$ .

# 6. Results and Discussions

### 6.1. Analysis of Simulation Results of Different Algorithms

In order to verify the effectiveness of the NSGA-III-I algorithm in solving manyobjective selective assembly optimization problems, a comparative simulation experiment was designed for the NSGA-III-I algorithm, interchangeable assembly (IA), NSGA-II, and NSGA-III. The basic parameters and operating environment of each algorithm were consistent. The original code for each algorithm comes from PlatEMO [38]. All of the algorithms were programmed with MATLAB and conducted on an AMD Ryzen 5 2400 G Processor computer. The parameters of each algorithm are as shown in Table 4.

Table 4. Parameters of each algorithm.

Algorithms	Population	Generation	Crossover Probability	Mutation Probability
NSGA-II	126	500	0.6	0.05
NSGA-III	126	500	0.6	0.05
NSGA-III-I	126	500	-	-

#### 6.1.1. Analysis of Final Solution Set of Each Algorithm

The hypervolume (HV) index was used to compare the performance of different algorithms for selective assembly problems. Suppose the solution set obtained by the algorithm is  $X_p$ ,  $Ref = (r_1, r_2, \dots, r_m)$  is the reference point and is dominated by any of the solutions in  $X_p$ . The HV value is the spatial hypervolume of the solution set  $X_p$  bounded by the reference point *Ref*. It is calculated by using Formula (30):

$$Hv(X_p) = Leb(\bigcup_{X \in X_p} [f_1(X), r_1] \times [f_2(X), r_2] \times \dots \times [f_m(X), r_m])$$
(30)

where  $Leb(X_p)$  denotes the Lebesgue measure of the solution set  $X_p$  and  $[f_1, r_1] \times [f_2, r_2] \times \cdots \times [f_m, r_m]$  denote a hypercube surrounded by all points that are dominated by the solution in  $X_p$  but not dominated by *Ref*.

The HV index is sensitive to the selection of reference point *Ref* when measuring the performance of different algorithms. To minimize the influence of the selection of reference points on the HV value, the final solution set of multiple algorithms was combined, and nondominated sorting was conducted to obtain the approximate real PF for the same set of experimental data, taking the reference point  $Ref = (1, 1, \dots, 1)$  [38]. The HV index is a positive index, and the larger the HV value is, the better the comprehensive performance of the algorithm is. When the HV value is 0, it indicates that the comprehensive performance of the algorithm is poor and that the solution set does not converge to the PF.

The 300 sets of component data were equally divided into 6 batches, each with 50 sets of components. Each algorithm was run 6 times per batch to reduce the influence of randomness, and the performance of each algorithm in the selection of assembly problems was compared according to the running results. The optimal value of the HV value of each algorithm solution set, the median value, the worst value and the mean value are listed in Table 5. Rows 1–4 in each group show the optimal, median, worst, and average HV values of the IA, NSGA-II, NSGA-III, and NSGA-III-I, respectively. The bolded data are the maximum value of each row. NSGA-III-I obtained the maximum optimal value, maximum worst value, maximum median value and maximum mean value of the HV value in six groups of simulation experiments. NSGA-III-I has better overall performance compared with other algorithms.

Batch.no	IA	NSGA-II	NSGA-III	NSGA-III-I
	0	$1.388 \times 10^{-2}$	$2.418 \times 10^{-2}$	$4.804 imes10^{-2}$
1	0	$5.641  imes 10^{-3}$	$1.471  imes 10^{-2}$	$3.254 imes10^{-2}$
1	0	$6.238 imes10^{-5}$	$4.188 imes10^{-3}$	$1.775 imes10^{-2}$
	0	$5.485  imes 10^{-3}$	$1.416  imes 10^{-2}$	$3.323 imes10^{-2}$
	0	$1.691\times 10^{-2}$	$1.978\times 10^{-2}$	$3.275 imes10^{-2}$
2	0	$7.422  imes 10^{-3}$	$1.136  imes 10^{-2}$	$2.273  imes 10^{-2}$
2	0	$1.271 \times 10^{-3}$	$2.986  imes 10^{-3}$	$1.084 imes10^{-2}$
	0	$7.792 \times 10^{-3}$	$1.201 \times 10^{-2}$	$2.197 imes10^{-2}$
	0	$3.606  imes 10^{-2}$	$2.161\times 10^{-2}$	$7.844 imes10^{-2}$
2	0	$2.854 imes10^{-3}$	$1.045 imes10^{-2}$	$4.722  imes 10^{-2}$
3	0	0	$2.897  imes 10^{-3}$	$1.046 imes10^{-2}$
	0	$9.258  imes 10^{-3}$	$1.162  imes 10^{-2}$	$4.829 imes10^{-2}$
	0	$1.808  imes 10^{-2}$	$2.539\times 10^{-2}$	$4.295 imes10^{-2}$
4	0	$6.214  imes 10^{-3}$	$9.664  imes 10^{-3}$	$3.645  imes 10^{-2}$
4	0	$5.581  imes 10^{-5}$	$2.044  imes 10^{-3}$	$1.821  imes 10^{-2}$
	0	$6.814 \times 10^{-3}$	$1.232 \times 10^{-2}$	$3.432  imes 10^{-2}$
	0	$3.070  imes 10^{-2}$	$5.397 imes10^{-2}$	$1.298 imes10^{-1}$
5	0	$2.422 \times 10^{-2}$	$2.288  imes 10^{-2}$	$7.950  imes 10^{-2}$
	0	$9.908  imes 10^{-3}$	$4.829  imes 10^{-3}$	$3.517  imes 10^{-2}$
	0	$2.270 \times 10^{-2}$	$2.976 \times 10^{-2}$	$8.157  imes 10^{-2}$
	0	$8.226 \times 10^{-3}$	$1.042 \times 10^{-2}$	$2.895  imes 10^{-2}$
6	0	$1.478  imes 10^{-3}$	$4.212 \times 10^{-3}$	$1.433 imes10^{-2}$
0	0	$5.620 imes10^{-4}$	$5.032  imes 10^{-5}$	$9.303 imes10^{-3}$
	0	$2.771 \times 10^{-3}$	$4.408  imes 10^{-3}$	$1.744 imes10^{-2}$

**Table 5.** Performance comparison between the proposed algorithm and other algorithms on the experimental problem with a batch size of 50.

The content in bold is the maximum value of each row.

To reflect the advantages and disadvantages of the Pareto optimal solution set of each algorithm more clearly, the Pareto optimal solution set obtained by the group of experiments of the first batch (batch size = 50) was used to describe the algorithms. Six random simulation experiments of each algorithm used the HV value to sort, and then the solution sets of different algorithms with the same serial number were classified and compared. The parallel coordinate diagrams of the solution sets of the three algorithms in the first group of experiments are shown in Figure 13.

In this group of experiments, a partial dominance relationship existed in the corresponding solution set after sorting the HV values of the solution set of each algorithm. The average value of each objective of the IA method was selected, and the partial nondominated solutions obtained by each algorithm were compared, as shown in Table A2.

In the six repeated experiments of the first batch, under the condition that the solution sets obtained by each algorithm are the current non-dominated solutions, the solution sets obtained by NSGA-III-I in the six experiments all have a certain number of solutions dominating the solutions of other methods. It shows that the NSGA-III-I has better performance in solving the problem of many-objective optimization in the selective assembly than other methods in the study, and the success rate increased from 57% (IA method) to 96–98% (NSGA-III-I) while the quality loss of each quality requirement was significantly reduced.



**Figure 13.** Parallel coordinate graph of the solution set of three types of algorithms in the first set of experiments.

#### 6.1.2. Analysis of Convergence of Each Algorithm

In the experiments, the convergence process of each algorithm is described by the variation of the minimum value of each objective function in each generation. The mean value of six repeated experiments is used to reduce the error (Figure 14).



Figure 14. Convergence process of each algorithm.

The results show that each objective function value of the NSGA-II and NSGA-III decreases rapidly in the iterative process but gradually falls into the local optimum around 200 generations, and it is difficult to jump out. In the NSGA-III-I, each objective function value keeps a relatively stable decline rate and can finally explore a solution set closer to the real Pareto frontier.

#### 6.1.3. Analysis of the Influence of Components Batch Size on Algorithm

To further study the influence of batch size and quantity on the performance of each algorithm, the same basic parameters and operating environment were used to repeat the test on the same 300 sets of components. The experiment is repeated for the same 300 mating components with 12 batches of 25 components each, 4 batches of 75 components each, and 3 batches of 100 components each. The results are shown in Tables A3–A5 respectively.

The results show that when the number of components in each batch is small (batch size = 25), the proposed algorithm obtains the maximum value in the optimal value, the maximum value in the median value and the maximum value in the worst value of HV in most experimental groups. With the increase in the number of components in each batch, the algorithm obtained the maximum value of the optimal value, the maximum value of the median value, and the maximum value of the worst value of all experimental groups.

As the number of components in each batch increases, some algorithms have gradually failed to converge (Figure 15). The results show that the IA method completely fails to converge to the approximate PF. With the increase in the number of components in each batch, both the NSGA-II and NSGA-III fail to converge to varying degrees. Only the NSGA-III-I algorithm can still converge to the approximate PF, with better performance than other algorithms and a better selective assembly scheme obtained.

#### Average number of times each algorithm does not converge



**Figure 15.** The average number comparison of non-convergence times of different methods in the same batch of experiments.

# 6.2. The VIKOR Method to Sort the Optimal Compromise Scheme

Taking the first set of experiments with 50 components per batch as an example, the Pareto solution set obtained by the NSGA-III-I algorithm is used as a decision matrix. The ideal solution in this experiment  $F^- = Min f(x) = (0.0200, 0.1658, 0.2320, 0.1731, 0.1537)$ . The step-by-step procedures and results are given below.

Step 1: The entropy method to obtain the objective weight (listed in Table 6).

Step 2: Triangular fuzzy numbers to determine the subjective weights of the assembly quality requirements. Six engineering decision makers were invited to evaluate the importance of each optimization objective in the cylinder–piston assembly selection optimization model (listed in Table 7). The comparison table converts semantic variables into triangular fuzzy numbers (listed in Table 8). The defuzzification method (Equation (24)) is performed to obtain the subjective weights (listed in Table 9).

Indicators	Information Entropy	Objective Weight $\omega^{'}{}_{j}$
$f_1$	0.9779	0.2613
$f_2$	0.9814	0.2197
$f_3$	0.9877	0.1457
$f_4$	0.9810	0.2246
$f_5$	0.9874	0.1486

Table 6. Objective weight of each index.

Table 7. Evaluation of the importance of each indicator by engineering decision makers.

Decision Makers	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$
$B_1$	Very high	Medium	Medium high	High	Low
$B_2$	Very high	Medium high	Medium	High	Medium low
$B_3$	High	Medium low	Medium low	Very high	Medium
$B_4$	Very high	Medium high	Medium	High	Medium
$B_5$	High	Medium high	Medium low	Very high	Medium high
<i>B</i> <sub>6</sub>	Very high	High	Very Low	High	Medium

Table 8. Importance evaluations transformed from semantic variables to triangular fuzzy numbers.

Decision Makers	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$
$B_1 \\ B_2 \\ B_3 \\ B_4 \\ B_5 \\ B_6$	$\begin{array}{c} (0.85, 1, 1) \\ (0.85, 1, 1) \\ (0.55, 0.9, 0.95) \\ (0.85, 1, 1) \\ (0.55, 0.9, 0.95) \\ (0.85, 1, 1) \end{array}$	$\begin{array}{l}(0.25, 0.5, 0.65)\\(0.45, 0.7, 0.8)\\(0, 0.3, 0.45)\\(0.45, 0.7, 0.8)\\(0.45, 0.7, 0.8)\\(0.45, 0.7, 0.8)\\(0.55, 0.9, 0.95)\end{array}$	$\begin{array}{c} (0.45, 0.7, 0.8) \\ (0.25, 0.5, 0.65) \\ (0, 0.3, 0.45) \\ (0.25, 0.5, 0.65) \\ (0, 0.3, 0.45) \\ (0, 0, 0, 0.2) \end{array}$	$\begin{array}{c} (0.55, 0.9, 0.95) \\ (0.55, 0.9, 0.95) \\ (0.85, 1, 1) \\ (0.55, 0.9, 0.95) \\ (0.85, 1, 1) \\ (0.55, 0.9, 0.95) \\ (0.55, 0.9, 0.95) \end{array}$	$\begin{array}{c}(0, 0.1, 0.25)\\(0, 0.3, 0.45)\\(0.25, 0.5, 0.65)\\(0.25, 0.5, 0.65)\\(0.45, 0.7, 0.8)\\(0.25, 0.5, 0.65)\end{array}$

Table 9. Triangular fuzzy matrix and subjective weight of each index.

Indicators	Triangular Fuzzy Matrix	Subjective Weight $\omega^{''}{}_j$
$f_1$	(0.55, 0.97, 1)	0.2831
$f_2$	(0, 0.63, 0.95)	0.1792
$f_3$	(0, 0.38, 0.8)	0.1265
$f_4$	(0.55, 0.93, 1)	0.2766
f <sub>5</sub>	(0, 0.43, 0.8)	0.1346

Step 3: Combination weightings are calculated by using Equation (25) (listed in Table 10).

Table 10. The combined weight of each index.

Indicators	$\omega_{j}^{'}$	$\omega^{''}{}_{j}$	$\omega_j$
$f_1$	0.2613	0.2831	0.2729
$f_2$	0.2197	0.1792	0.1991
f3	0.1457	0.1265	0.1362
$f_4$	0.2246	0.2766	0.2500
$f_5$	0.1486	0.1346	0.1419

Step 4: Calculation of the *S*-value, *R*-value, and *Q*-value of each Pareto solution in turn. The six optimal schemes sorted according to the *Q*-values are listed in Table 11.

Experiment No	Chromosome No.	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	Q	S	R
1	51	0.0200	0.1808	0.2352	0.2119	0.2032	0.0020	0.1739	0.1017
2	84	0.0200	0.2027	0.2544	0.2119	0.1690	0.0278	0.1998	0.1017
3	60	0.0200	0.1765	0.2496	0.2158	0.1882	0.0333	0.1750	0.1120
4	12	0.0200	0.1753	0.2352	0.2168	0.2054	0.0452	0.1793	0.1146
5	68	0.0200	0.1730	0.2448	0.2178	0.1890	0.0454	0.1719	0.1172
6	56	0.0200	0.1807	0.2320	0.2168	0.1988	0.0467	0.1808	0.1146

Table 11. Information on the 6 schemes with the best *Q*-value sorting.

Step 5: Discussions and decision making based on the above results. The analysis of the calculation results in Table 11 shows that (1)  $Q(X_2) - Q(X_1) \ge 1/(126 - 1)$ , and (2) when sorting according to the *Q*-value, scheme  $X_1$  is the optimal choice; when sorting according to the *R* value, scheme  $X_1$  is still the optimal choice; and when sorting according to the *S* value, scheme  $X_1$  is superior to scheme  $X_2$ . According to VIKOR theory, when the decision maker has a compromising attitude towards all schemes (v = 0.5), scheme  $X_1$  can be selected as the optimal scheme among all non-dominated schemes (listed in Table 12).

Table 12. Selective assembly scheme corresponding components number combination.

Component	Best Combination of Mating Components
Piston	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, <b>44</b> , 45, 46, 47, 48, 49, 50
Piston ring	18, 32, 14, 29, 38, 15, 8, 6, 23, 30, 34, 22, 25, 16, 12, 7, 42, 1, 9, 36, 27, 21, 33, 41, 17, 3, 13, 43, 37, 49, 26, 20, 10, 11, 46, 45, 50, 48, 2, 40, 5, 39, 28, <b>19</b> , 47, 44, 31, 24, 4, 35
Cylinder	35, 33, 36, 22, 27, 3, 2, 28, 21, 7, 6, 43, 41, 37, 8, 9, 50, 29, 4, 1, 45, 30, 40, 32, 17, 13, 15, 18, 44, 49, 42, 48, 47, 39, 25, 26, 5, 20, 11, 34, 16, 38, 24, <b>23</b> , 12, 19, 14, 31, 46, 10

The content in bold is the only set that cannot be successfully assembled in the 50 sets of components.

According to the above five steps, we conclude that scheme  $X_1$  is the optimal compromise solution. Table 12 shows the detailed content of scheme  $X_1$ , where each row represents the sequence number arrangement of the same type of components. The 50 pistons, 50 piston rings, and 50 cylinders are assembled in one-to-one correspondence. For example, piston #1 lines up with piston ring #18 and cylinder #35, and they are all located at the first position in their respective arrangement sequence. In particular, piston #44, piston ring #19, and cylinder #23 is the only set that cannot be successfully assembled in the 50 sets of components.

#### 7. Conclusions

This research focuses on the optimization of selective assembly of complex mechanical products (CMPs) under intelligent system environment conditions. A general method based on the NSGA-III-I and VIKOR was proposed to solve the many-objective optimization and decision-making problem in the selective assembly of CMPs, which could optimize the assembly effect more effectively and select the assembly scheme rationally. The main contributions are as follows:

(1) A mathematical model of selective assembly based on the success rate and the Taguchi quality loss was constructed, dimensional constraints and quality requirements were established through assembly structure analysis, and multiple optimization objectives s were established on this basis.

- (2) For several optimization objectives, the NSGA-III-I was proposed, and the simulation case was solved. Experiments were designed to compare the NSGA-III-I with the IA, NSGA-II, and NSGA-III methods, and the performance changes of each algorithm under different batch conditions were studied. The results show that the proposed algorithm has obvious advantages in solving many-objective selective assembly problems. The proposed method increases the assembly success rate from 57% (IA) to 96–98% (NSGA-III-I), while reducing quality loss significantly. It effectively avoids falling into the local optimum compared with the NSGA-II and NSGA-III.
- (3) Considering the production mode of human-machine intelligent system interaction, the optimal compromise solution is obtained by using fuzzy theory, entropy theory, and the VIKOR method. The planning scheme can simultaneously maximize group benefits and minimize individual regret.

Future research work will add a quantitative method and constraint conditions of form and position tolerance on the basis of the proposed selective assembly model so that the method can be more widely used in the selective assembly process of almost all components. Therefore, it can be integrated into a digital manufacturing execution system (MES) and systematically applied to CMP manufacturing enterprises. After designing for specific problems, the proposed method can be applied to many different types of many-objective combinatorial optimization, such as a production scheduling problem, a vehicle routing problem, and a distributed network resource allocation optimization problem, etc.

**Author Contributions:** Conceptualization, R.P. and Y.Z.; methodology, R.P.; software, R.P.; validation, Y.Z., R.P. and J.Y.; formal analysis, Y.Z.; investigation, J.Y.; resources, Y.Z. and J.Y.; data curation, R.P.; writing—original draft preparation, R.P.; writing—review and editing, R.P.; visualization, R.P.; supervision, Y.Z.; project administration, Y.Z.; funding acquisition, Y.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Ministry of Industry and Information Technology of China, grant number "TC200H01X-5" and the Science and Technology Department of Xinjiang Production and Construction Corps, China, grant number "S2020AA784".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

**Conflicts of Interest:** The authors declare that they have no conflict of interest.

#### Appendix A

Table A1. Data of the example problem.

Batch no.	Component	Characteristic	Tolerances (μm) of the Quality Characteristics of Individual Components (for Component no. 1–50)
1	C1	А	13, 6, 11, 7, 13, 8, 9, 3, 2, 4, 6, 12, 5, 5, 10, 17, 1, 8, 16, 13, 15, 7, 13, 1, 3, 0, 7, 10, 8, 7, 2, 9, 11, 14, 10, 11, 12, 15, 13, 1, 14, 13, 3, 10, 11, 11, 15, 4, 7, 11
		В	4, 4, 0, 6, 2, 4, 11, 2, 10, 4, 6, 4, 8, 5, 7, 7, 6, 4, 7, 6, 4, 9, 1, 4, 8, 3, 2, 7, 3, 5, 10, 3, 4, 3, 12, 12, 1, 1, 6, 3, 4, 5, 0, 5, 3, 6, 3, 8, 10, 6
		С	3, 12, 4, 9, 9, 2, 1, 9, 8, 11, 9, 6, 7, 0, 9, 7, 2, 3, 5, 6, 7, 10, 9, 11, 4, 7, 11, 7, 5, 7, 9, 0, 12, 4, 5, 8, 0, 6, 1, 10, 5, 11, 3, 12, 7, 8, 6, 6, 9, 8,
	C2	D	1, 10, 6, 11, 1, 9, 7, 10, 7, 10, 7, 15, 8, 2, 9, 5, 3, 12, 0, 9, 6, 11, 10, 8, 16, 11, 5, 7, 9, 6, 0, 9, 4, 12, 3, 10, 5, 8, 4, 5, 11, 9, 9, 7, 5, 13, 0, 7, 13, 7
		Е	0, 1, 3, 5, 2, 6, 3, 1, 2, 4, 2, 2, 4, 2, 3, 0, 1, 2, 4, 0, 3, 2, 3, 3, 2, 5, 3, 1, 5, 4, 2, 4, 2, 5, 3, 2, 2, 5, 4, 6, 5, 1, 4, 2, 5, 3, 4, 2, 4, 1
		F	6, 8, 12, 9, 11, 12, 18, 19, 6, 23, 20, 16, 6, 16, 16, 1, 3, 16, 20, 8, 15, 14, 11, 23, 11, 22, 12, 6, 20, 20, 12, 11, 4, 16, 12, 10, 7, 8, 15, 12, 17, 4, 18, 6, 19, 16, 13, 4, 18, 7
	C3	G	15, 23, 18, 11, 5, 15, 9, 15, 18, 9, 15, 3, 13, 6, 11, 13, 12, 16, 7, 13, 16, 11, 23, 0, 23, 19, 4, 8, 2, 16, 23, 10, 8, 10, 14, 5, 9, 13, 16, 6, 16, 19, 8, 11, 11, 15, 16, 14, 16, 6

Batch no.	Component	Characteristic	Tolerances (μm) of the Quality Characteristics of Individual Components (for Component no. 1–50)
2	C1	А	11, 10, 4, 16, 7, 7, 8, 7, 12, 9, 17, 6, 3, 4, 17, 11, 10, 12, 13, 11, 4, 11, 12, 5, 12, 7, 14, 12, 5, 10, 6, 6, 13, 11, 8, 5, 11, 9, 2, 12, 13, 16, 5, 10, 8, 9, 15, 17, 2, 8
		В	2, 8, 8, 5, 6, 9, 6, 9, 5, 5, 5, 1, 3, 6, 5, 6, 5, 10, 7, 1, 3, 6, 8, 3, 3, 6, 5, 5, 2, 2, 7, 10, 2, 10, 7, 4, 4, 11, 2, 12, 2, 4, 6, 3, 12, 6, 5, 4, 9, 3
		С	12, 7, 2, 9, 10, 5, 7, 8, 6, 9, 3, 5, 2, 4, 9, 8, 4, 5, 11, 7, 4, 6, 12, 11, 10, 9, 7, 11, 7, 8, 2, 7, 0, 5, 9, 4, 1, 12, 6, 4, 11, 6, 10, 11, 9, 2, 7, 5, 9, 2
	C2	D	2, 10, 12, 7, 4, 2, 15, 4, 7, 13, 17, 17, 6, 9, 1, 9, 8, 11, 11, 7, 11, 5, 18, 7, 13, 12, 14, 12, 5, 13, 8, 13, 14, 7, 9, 12, 10, 11, 14, 5, 14, 1, 4, 6, 15, 2, 10, 12, 10, 15
		Е	5, 2, 0, 5, 2, 2, 1, 5, 4, 2, 3, 4, 5, 4, 3, 2, 2, 2, 1, 4, 4, 4, 1, 3, 3, 5, 3, 1, 3, 6, 2, 2, 0, 4, 0, 4, 2, 2, 4, 1, 2, 5, 4, 5, 4, 1, 5, 5, 2, 2
		F	7, 17, 19, 16, 13, 11, 10, 15, 14, 9, 6, 14, 11, 12, 21, 21, 11, 21, 15, 7, 6, 9, 16, 18, 3, 17, 18, 4, 16, 9, 9, 15, 4, 13, 13, 10, 14, 6, 13, 12, 9, 17, 23, 13, 24, 16, 12, 19, 15, 17
	C3	G	14, 13, 9, 20, 22, 15, 8, 10, 11, 13, 24, 15, 14, 17, 12, 11, 18, 23, 10, 7, 17, 11, 1, 21, 17, 23, 3, 1, 10, 15, 18, 5, 11, 8, 16, 23, 10, 0, 10, 24, 2, 12, 1, 18, 3, 4, 20, 9, 12, 7
3	C1	А	16, 9, 5, 8, 4, 9, 5, 16, 11, 14, 12, 16, 7, 6, 5, 6, 15, 17, 0, 13, 6, 16, 7, 13, 7, 12, 7, 13, 13, 10, 12, 9, 15, 3, 7, 14, 6, 5, 13, 13, 3, 1, 6, 16, 3, 15, 11, 1, 5, 6
		В	4, 6, 4, 5, 5, 9, 2, 8, 6, 5, 5, 4, 7, 11, 7, 12, 12, 9, 0, 9, 4, 5, 5, 5, 9, 9, 10, 2, 8, 8, 6, 11, 6, 6, 1, 7, 4, 2, 2, 7, 12, 1, 3, 2, 3, 9, 10, 1, 4, 6
		С	7, 11, 8, 5, 5, 3, 2, 10, 5, 5, 2, 5, 8, 9, 2, 8, 3, 7, 1, 5, 7, 6, 9, 11, 11, 7, 7, 8, 8, 9, 5, 12, 12, 5, 8, 3, 4, 6, 9, 9, 4, 1, 5, 11, 7, 4, 5, 5, 0, 7
	C2	D	10, 5, 9, 13, 11, 2, 18, 11, 16, 10, 13, 15, 9, 13, 16, 8, 17, 11, 5, 15, 9, 15, 7, 13, 13, 13, 11, 3, 8, 14, 1, 9, 16, 0, 17, 12, 3, 5, 6, 13, 11, 8, 10, 10, 1, 13, 8, 15, 9, 4
		Е	3, 1, 2, 2, 5, 3, 4, 2, 4, 3, 2, 4, 3, 3, 4, 4, 4, 5, 5, 6, 1, 3, 6, 2, 3, 3, 4, 1, 3, 5, 1, 0, 1, 6, 1, 4, 4, 6, 2, 1, 2, 4, 3, 4, 1, 3, 5, 1, 0, 1, 6, 1, 4, 4, 6, 2, 1, 2, 4, 3, 4, 1, 3, 5, 1, 0, 1, 6, 1, 4, 4, 6, 2, 1, 2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4,
		F	13, 8, 9, 18, 9, 20, 5, 13, 5, 9, 12, 12, 16, 15, 20, 12, 21, 4, 12, 12, 3, 23, 18, 6, 18, 16, 15, 11, 15, 9, 15, 9, 17, 11, 2, 9, 13, 14, 8, 13, 8, 7, 18, 3, 11, 1, 7, 8, 5, 6
	C3	G	20, 4, 4, 17, 23, 8, 6, 7, 22, 6, 12, 11, 7, 12, 14, 6, 5, 5, 18, 16, 5, 21, 15, 18, 20, 6, 9, 9, 15, 20, 7, 7, 8, 15, 6, 11, 13, 10, 11, 17, 5, 15, 11, 14, 12, 5, 22, 22, 24, 19
4	C1	А	11, 3, 11, 2, 7, 5, 3, 3, 9, 8, 13, 16, 9, 2, 9, 11, 7, 4, 7, 10, 6, 11, 15, 15, 9, 7, 4, 9, 5, 15, 9, 8, 1, 3, 1, 1, 16, 8, 7, 15, 10, 17, 9, 4, 12, 12, 8, 11, 4, 5
		В	5, 2, 10, 5, 8, 7, 10, 6, 7, 1, 2, 8, 4, 6, 1, 6, 8, 1, 3, 1, 5, 4, 3, 10, 9, 5, 1, 5, 2, 10, 10, 4, 6, 7, 7, 7, 8, 10, 10, 5, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10
		С	9, 7, 7, 5, 12, 10, 7, 9, 7, 3, 2, 8, 7, 10, 8, 6, 8, 7, 9, 11, 9, 5, 5, 5, 7, 7, 4, 10, 6, 6, 1, 3, 8, 7, 3, 6, 9, 3, 6, 9, 3, 6, 9, 5, 6, 1, 6, 6, 10, 5, 4, 11
	C2	D	1, 9, 6, 11, 3, 12, 10, 12, 2, 5, 7, 10, 17, 6, 11, 7, 16, 15, 10, 14, 10, 9, 2, 7, 7, 8, 4, 14, 14, 11, 10, 12, 16, 12, 18, 16, 8, 9, 14, 12, 9, 9, 15, 10, 12, 10, 12, 8, 6
		Е	5, 5, 2, 3, 2, 1, 3, 4, 1, 4, 6, 3, 4, 2, 1, 4, 4, 6, 5, 4, 2, 2, 1, 6, 5, 3, 2, 4, 2, 2, 2, 2, 4, 4, 1, 4,
		F	5, 19, 11, 22, 16, 7, 13, 8, 16, 6, 8, 18, 6, 13, 22, 15, 9, 7, 17, 16, 13, 23, 11, 20, 14, 14, 15, 6, 17, 17, 17, 17, 16, 13, 23, 17, 17, 16, 13, 23, 17, 17, 16, 13, 24, 14, 15, 15, 16, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17
	C3	G	10, 16, 12, 7, 3, 1, 13, 4, 6, 20, 12, 8, 23, 14, 10, 3, 0, 20, 2, 6, 14, 15, 2, 8, 16, 5, 19, 10, 9, 11, 15, 6, 10, 23, 12, 13, 21, 20, 8, 13, 18, 15, 16, 16, 16, 7, 17, 6, 21, 4
5	C1	А	7, 3, 5, 8, 18, 5, 11, 15, 10, 1, 10, 16, 12, 7, 7, 12, 7, 8, 4, 14, 4, 12, 8, 17, 10, 17, 17, 16, 8,
		В	12, 4, 6, 9, 10, 10, 12, 11, 3, 3, 16, 2, 15, 9, 13, 11, 14, 4, 13, 8, 13 5, 7, 3, 5, 8, 4, 3, 7, 6, 9, 9, 4, 4, 7, 11, 10, 4, 7, 10, 6, 2, 5, 9, 3, 6, 9, 12, 5, 7, 1, 8, 11, 0, 4, 8,
		С	8, 11, 6, 6, 4, 3, 10, 6, 2, 6, 0, 7, 6, 7, 4 8, 10, 5, 4, 1, 7, 5, 7, 0, 6, 4, 10, 4, 2, 7, 9, 4, 6, 9, 10, 10, 7, 3, 8, 4, 10, 10, 7, 5, 2, 5, 4, 0, 9,
	C2	D	5, 3, 7, 2, 10, 11, 5, 11, 7, 2, 1, 6, 5, 8, 4, 10 15, 4, 16, 9, 7, 4, 17, 15, 12, 11, 12, 12, 12, 1, 5, 14, 5, 5, 8, 6, 2, 15, 8, 1, 4, 15, 7, 8, 13, 11,
		Ē	17, 8, 14, 18, 11, 1, 2, 4, 5, 6, 0, 17, 17, 9, 5, 7, 11, 16, 11, 8 3, 2, 3, 5, 3, 4, 2, 5, 2, 2, 4, 5, 5, 3, 4, 1, 5, 6, 4, 5, 6, 2, 4, 5, 2, 5, 2, 3, 5, 1, 2, 5, 0, 3, 4, 4, 2,
		F	6, 3, 4, 4, 4, 3, 4, 3, 5, 2, 3, 2, 1 9, 8, 13, 21, 9, 6, 11, 22, 7, 10, 5, 6, 5, 21, 11, 8, 14, 15, 13, 9, 14, 16, 20, 16, 10, 24, 17, 1,
	C3	G	12, 8, 6, 7, 1, 13, 20, 11, 10, 7, 20, 10, 6, 16, 15, 16, 14, 9, 16, 4, 1, 4 2, 20, 18, 9, 13, 12, 12, 15, 17, 6, 23, 11, 7, 22, 16, 11, 14, 11, 16, 9, 11, 6, 7, 6, 3, 13, 14, 17,
		6	10, 9, 14, 10, 11, 7, 11, 9, 16, 22, 12, 20, 10, 21, 15, 16, 22, 14, 10, 4, 11, 18
6	C1	А	16, 10, 13, 16, 15, 15, 15, 15, 15, 16, 14, 15, 11, 7, 15, 16, 17, 7, 16, 12, 9, 5, 9, 12, 0, 10, 14, 7, 5, 16, 12, 10, 11, 9, 15, 16, 12, 10, 11, 9, 15, 16, 12, 10, 11, 9, 15, 16, 12, 10, 11, 9, 15, 16, 16, 16, 16, 16, 16, 16, 16, 16, 16
		В	9, 8, 6, 12, 11, 8, 10, 4, 7, 7, 9, 10, 6, 11 9, 8, 6, 12, 11, 8, 10, 4, 7, 7, 9, 10, 6, 11
		С	7, 4, 2, 2, 5, 4, 5, 12, 6, 4, 0, 5, 0, 9, 12, 5, 6, 5, 0, 11, 6, 7, 9, 4, 1, 10, 1, 1, 4, 7, 6, 6, 6, 5, 9, 5, 8, 5, 4, 6, 5, 7, 7, 8, 8, 10, 9, 12, 5, 6
	C2	D	2, 14, 14, 2, 1, 2, 8, 11, 7, 12, 9, 3, 9, 15, 10, 10, 10, 10, 3, 10, 12, 7, 2, 8, 17, 8, 9, 13, 5, 6, 7, 14, 7, 7, 8, 2, 6, 9, 14, 10, 2, 10, 5, 4, 5, 13, 5, 14, 16, 13
		Е	3, 4, 2, 2, 1, 6, 5, 3, 4, 2, 2, 3, 2, 2, 1, 4, 2, 3, 4, 4, 2, 3, 3, 3, 5, 2, 2, 3, 6, 4, 1, 3, 2, 5, 3, 1, 2, 3, 1, 2, 3, 3, 2, 2, 6, 2, 2, 1, 4, 1
		F	7, 7, 5, 9, 7, 19, 5, 7, 13, 23, 6, 10, 19, 18, 15, 11, 6, 24, 18, 6, 19, 7, 12, 16, 6, 16, 13, 18, 19, 12, 19, 20, 16, 14, 2, 11, 16, 18, 11, 11, 7, 5, 5, 10, 9, 7, 22, 16, 19, 20
	C3	G	7, 19, 17, 4, 18, 20, 22, 13, 12, 4, 4, 6, 14, 6, 6, 22, 24, 16, 4, 0, 16, 2, 24, 11, 21, 15, 24, 20, 17, 16, 12, 19, 16, 17, 21, 17, 19, 12, 4, 8, 2, 18, 22, 4, 18, 9, 19, 9, 23, 22

# Table A1. Cont.

Experiment no	Selective Assembly Method	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$
	IA	0.4343	0.5523	0.6112	0.5709	0.5642
	NSGA-II	0.0800	0.1783	0.3456	0.3143	0.3408
1	NSGA-III	0.0400	0.1690	0.3136	0.2491	0.2274
	NSGA-III-I	0.0200	0.1474	0.2896	0.2153	0.1636
	IA	0.4330	0.5481	0.6107	0.5689	0.5614
2	NSGA-II	0.0800	0.2023	0.2960	0.2746	0.3018
2	NSGA-III	0.0600	0.1973	0.2832	0.2627	0.2347
	NSGA-III-I	0.0200	0.1736	0.2448	0.2178	0.1890
	IA	0.4263	0.5435	0.6048	0.5616	0.5606
2	NSGA-II	0.0600	0.2053	0.3656	0.3835	0.2415
3	NSGA-III	0.0400	0.1878	0.3368	0.2667	0.2281
	NSGA-III-I	0.0200	0.1459	0.3160	0.2565	0.1888
	IA	0.4267	0.5465	0.6108	0.5636	0.5577
4	NSGA-II	0.0800	0.2301	0.3680	0.2533	0.3369
4	NSGA-III	0.0600	0.1987	0.3352	0.2479	0.2731
	NSGA-III-I	0.0400	0.1409	0.2992	0.2262	0.2114
	IA	0.4238	0.5405	0.6041	0.5632	0.5560
-	NSGA-II	0.1000	0.2482	0.3064	0.3859	0.3333
5	NSGA-III	0.0800	0.1966	0.3056	0.2743	0.2590
	NSGA-III-I	0.0200	0.1874	0.2912	0.2395	0.1500
	IA	0.4284	0.5469	0.6085	0.5628	0.5602
(	NSGA-II	0.1000	0.2352	0.4840	0.2689	0.2649
6	NSGA-III	0.0600	0.2135	0.3864	0.2546	0.2278
	NSGA-III-I	0.0400	0.1981	0.2872	0.2195	0.2099
3456	INSGA-III NSGA-III-I IA NSGA-III NSGA-III NSGA-III-I NSGA-III NSGA-III-I IA NSGA-III NSGA-III NSGA-III NSGA-III NSGA-III	0.0600 0.0400 0.0200 0.4267 0.0800 0.0600 0.4238 0.1000 0.0800 0.0200 0.4284 0.1000 0.4284 0.1000 0.0600 0.0400	0.2053 0.1878 0.1459 0.5465 0.2301 0.1987 0.1409 0.5405 0.2482 0.1966 0.1874 0.5469 0.2352 0.2135 0.1981	0.3656 0.3368 0.3160 0.6108 0.3680 0.3352 0.2992 0.6041 0.3064 0.3056 0.2912 0.6085 0.4840 0.3864 0.2872	0.3835 0.2667 0.2565 0.5636 0.2533 0.2479 0.2262 0.5632 0.3859 0.2743 0.2395 0.5628 0.2689 0.2546 0.2195	0.2415 0.2281 0.1888 0.5577 0.3369 0.2731 0.2114 0.5560 0.3333 0.2590 0.1500 0.5602 0.2649 0.2278 0.2099

 Table A2. The dominant relation of the partial solution of each algorithm.

The content in bold is the highest priority non-dominated solution.

**Table A3.** Performance comparison between the proposed algorithm and other algorithms on the experimental problem with a batch size of 25.

Batch.no	IA	NSGA-II	NSGA-III	NSGA-III-I
	IA	NSGA-II	NSGA-III	NSGA-III-I
1	0	$3.247 imes10^{-2}$	$5.987 imes10^{-2}$	$8.406 imes10^{-2}$
1	0	$2.924 imes10^{-2}$	$4.715 imes10^{-2}$	$5.229 imes10^{-2}$
	0	$1.232 \times 10^{-2}$	$3.989  imes 10^{-2}$	$3.894  imes 10^{-2}$
	0	$2.663 \times 10^{-2}$	$4.848  imes 10^{-2}$	$5.599 imes10^{-2}$
r	0	$5.847 imes10^{-2}$	$1.086 imes10^{-1}$	$1.283 imes10^{-1}$
Z	0	$3.972 \times 10^{-2}$	$9.298 imes10^{-2}$	$1.179 imes10^{-1}$
	0	$1.297 \times 10^{-2}$	$4.729 \times 10^{-2}$	$7.856  imes 10^{-2}$
	0	$3.888  imes 10^{-2}$	$8.671\times 10^{-2}$	$1.105 imes10^{-1}$
3	0	$8.940  imes 10^{-2}$	$1.303 imes10^{-1}$	$1.356 imes10^{-1}$
5	0	$7.876 imes10^{-2}$	$7.324  imes 10^{-2}$	$6.705  imes 10^{-2}$
	0	$4.991 \times 10^{-2}$	$2.916 \times 10^{-2}$	$5.357  imes 10^{-2}$
	0	$7.551  imes 10^{-2}$	$7.376 imes10^{-2}$	$8.221  imes 10^{-2}$
4	0	$4.312 \times 10^{-2}$	$4.730  imes 10^{-2}$	$4.246  imes 10^{-2}$
4	0	$1.503 \times 10^{-2}$	$2.280  imes 10^{-2}$	$1.839 \times 10^{-2}$
	0	$4.162 \times 10^{-3}$	$1.804 \times 10^{-3}$	$1.282  imes 10^{-2}$
5	0	$1.902 \times 10^{-2}$	$2.424 \times 10^{-2}$	$2.172 \times 10^{-2}$
	0	$8.281 \times 10^{-2}$	$7.696 \times 10^{-2}$	$1.246 imes10^{-1}$
	0	$3.813  imes 10^{-2}$	$5.625  imes 10^{-2}$	$1.032 imes10^{-1}$
	0	$2.131 \times 10^{-2}$	$1.101 \times 10^{-2}$	$7.887  imes 10^{-2}$

Batch.no	IA	NSGA-II	NSGA-III	NSGA-III-I
	0	$4.739  imes 10^{-2}$	$5.330 \times 10^{-2}$	$1.026  imes 10^{-1}$
<i>,</i>	0	$9.702 \times 10^{-2}$	$1.650  imes 10^{-1}$	$1.809 imes10^{-1}$
6	0	$2.543 \times 10^{-2}$	$1.164 imes10^{-2}$	$1.582 imes10^{-1}$
	0	$4.678 imes10^{-3}$	$4.501  imes 10^{-3}$	$1.510 imes10^{-1}$
	0	$4.020  imes 10^{-2}$	$5.816  imes 10^{-2}$	$1.639 imes10^{-1}$
7	0	$6.026 \times 10^{-2}$	$7.701 \times 10^{-2}$	$1.060 imes10^{-1}$
7	0	$2.017 \times 10^{-2}$	$4.252 \times 10^{-2}$	$5.693  imes 10^{-2}$
	0	$1.437 \times 10^{-2}$	$2.038 imes10^{-2}$	$3.206 \times 10^{-3}$
	0	$2.590  imes 10^{-2}$	$4.805  imes 10^{-2}$	$5.784 imes10^{-2}$
0	0	$1.998 imes 10^{-1}$	$2.089 imes10^{-1}$	$2.664 imes10^{-1}$
0	0	$1.501 imes10^{-1}$	$1.750 imes10^{-1}$	$2.388 imes10^{-1}$
	0	$1.213 imes10^{-1}$	$1.689  imes 10^{-1}$	$1.949 imes10^{-1}$
	0	$1.537  imes 10^{-1}$	$1.818  imes 10^{-1}$	$2.315 imes10^{-1}$
0	0	$1.423  imes 10^{-1}$	$1.870  imes 10^{-1}$	$2.259 imes10^{-1}$
	0	$1.289  imes 10^{-1}$	$1.455 imes10^{-1}$	$1.813 imes10^{-1}$
	0	$3.925 \times 10^{-3}$	$1.870  imes 10^{-1}$	$2.259 imes10^{-1}$
	0	$1.011 imes10^{-1}$	$1.536 imes10^{-1}$	$1.787 imes10^{-1}$
10	0	$1.233 imes10^{-1}$	$1.265 imes10^{-1}$	$1.437 imes10^{-1}$
10	0	$7.368  imes 10^{-2}$	$7.051  imes 10^{-2}$	$1.195 imes10^{-1}$
	0	$4.386 \times 10^{-3}$	$6.628 \times 10^{-3}$	$8.995  imes 10^{-2}$
	0	$7.120  imes 10^{-2}$	$6.839 imes10^{-2}$	$1.195 imes10^{-1}$
11	0	$1.486  imes 10^{-2}$	$1.762 \times 10^{-2}$	$5.549 imes10^{-2}$
11	0	$3.852 \times 10^{-3}$	$9.424  imes 10^{-3}$	$3.040  imes 10^{-2}$
	0	$8.100  imes 10^{-4}$	$2.206 \times 10^{-3}$	$1.817  imes 10^{-2}$
	0	$5.997 imes10^{-3}$	$9.203  imes 10^{-3}$	$3.161 imes10^{-2}$
12	0	$3.858 \times 10^{-2}$	$9.113  imes 10^{-2}$	$1.107 imes10^{-1}$
12	0	$2.890  imes 10^{-2}$	$4.795 imes10^{-2}$	$8.209  imes 10^{-2}$
	0	$7.884  imes 10^{-3}$	$2.141 \times 10^{-2}$	$7.060  imes 10^{-2}$

Table A3. Cont.

The content in bold is the maximum value of each row.

**Table A4.** Performance comparison between the proposed algorithm and other algorithms on the experimental problem with a batch size of 75.

Batch.no	IA	NSGA-II	NSGA-III	NSGA-III-I
	0	$5.532  imes 10^{-4}$	$4.049 imes10^{-4}$	$1.364  imes 10^{-2}$
1	0	0	$7.551  imes 10^{-5}$	$7.339 imes10^{-3}$
1	0	0	0	$6.041 imes10^{-4}$
	0	$1.804 imes10^{-4}$	$1.210 imes10^{-4}$	$6.862  imes 10^{-3}$
	0	$6.621 imes10^{-4}$	$3.098  imes 10^{-3}$	$1.851 imes10^{-2}$
C	0	$4.643 imes10^{-4}$	$8.197 imes10^{-4}$	$1.252 imes10^{-2}$
Z	0	$1.854 imes10^{-4}$	$3.277  imes 10^{-5}$	$6.627 imes10^{-3}$
	0	$4.497 imes10^{-4}$	$1.276 \times 10^{-3}$	$1.225  imes 10^{-2}$
	0	$7.173 imes10^{-4}$	$5.674  imes 10^{-3}$	$2.628 imes10^{-2}$
2	0	$2.273  imes 10^{-4}$	$1.301  imes 10^{-4}$	$2.521  imes 10^{-2}$
3	0	0	0	$1.375 imes10^{-3}$
	0	$2.521 \times 10^{-4}$	$1.469 \times 10^{-3}$	$1.934  imes 10^{-2}$
4	0	$1.569 imes10^{-3}$	$4.199 imes10^{-3}$	$2.218 imes10^{-2}$
	0	$2.479  imes 10^{-4}$	$4.340 imes10^{-5}$	$1.170 imes10^{-2}$
	0	0	0	$9.357 imes10^{-4}$
	0	$4.618  imes 10^{-4}$	$7.978 imes10^{-4}$	$1.192  imes 10^{-2}$

The content in bold is the maximum value of each row.

Batch.no	IA	NSGA-II	NSGA-III	NSGA-III-I
	0	$3.850 \times 10^{-7}$	$3.488  imes 10^{-4}$	$7.787  imes 10^{-3}$
1	0	0	0	$4.805 imes10^{-3}$
1	0	0	0	$2.558 imes10^{-3}$
	0	$6.417 imes10^{-8}$	$6.714 imes10^{-5}$	$5.110  imes 10^{-3}$
	0	$2.206\times10^{-5}$	$6.050  imes 10^{-5}$	$1.788 imes10^{-2}$
2	0	0	0	$8.503 imes10^{-3}$
2	0	0	0	$6.618 imes10^{-4}$
	0	$5.479 \times 10^{-6}$	$1.465  imes 10^{-5}$	$8.875 imes10^{-3}$
3	0	$1.965 imes10^{-4}$	$1.872  imes 10^{-4}$	$6.665  imes 10^{-3}$
	0	0	$1.842 imes10^{-5}$	$3.867  imes 10^{-3}$
	0	0	$1.152  imes 10^{-21}$	$5.807 imes10^{-4}$
	0	$4.083 imes10^{-5}$	$4.665 imes10^{-5}$	$3.702 imes10^{-3}$

**Table A5.** Performance comparison between the proposed algorithm and other algorithms on the experimental problem with a batch size of 100.

The content in bold is the maximum value of each row.

#### References

- Mahalingam, S.K.; Nagarajan, L.; Salunkhe, S.; Nasr, E.A.; Davim, J.P.; Hussein, H.M.A. Harmony Search Algorithm for Minimizing Assembly Variation in Non-linear Assembly. *Appl. Sci.* 2021, 11, 9213. [CrossRef]
- Liu, L.; Zhu, F.; Chen, J.; Ma, Y.; Tu, Y. A quality control method for complex product selective assembly processes. *Int. J. Prod. Res.* 2013, *51*, 5437–5449. [CrossRef]
- Kannan, S.M. Manufacturing Mean Design for Selective Assembly to Minimize Surplus Parts. Proceedings of International Conference on Quality and Reliability; RMIT University: Melbourne, Australia, 2002; pp. 259–264.
- 4. Matsuura, S.A.; Shinozaki, N.B. Shifting the process mean to minimize surplus components and unacceptable products in selective assembly. *J. Qual.* **2011**, *18*, 87–100.
- 5. Kannan, S.; Jayabalan, V.; Jeevanantham, K. Genetic algorithm for minimizing assembly variation in selective assembly. *Int. J. Prod. Res.* **2003**, *41*, 3301–3313. [CrossRef]
- 6. Kannan, S.M.; Jeevanantham, A.K.; Jayabalan, V. Modelling and analysis of selective assembly using Taguchi's loss function. *Int. J. Prod. Res.* **2008**, *46*, 4309–4330. [CrossRef]
- Fei, J.F.; Lu, C.; Wang, S.L. Minimizing Surplus Parts in Selective Assembly Using GA. Appl. Mech. Mater. 2012, 215–216, 178–181. [CrossRef]
- 8. Tan, M.H.Y.; Wu, C.F.J. Generalized selective assembly. IIE Trans. 2012, 44, 27–42. [CrossRef]
- 9. Lu, C.; Fei, J. An approach to minimizing surplus parts in selective assembly with genetic algorithm. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* 2014, 229, 508–520. [CrossRef]
- Manickam, N.; De, T.N. Optimum Group Size Selection for Launch Vehicle Sections Linear Assembly by Selective Assembly Method. Int. J. Eng. Technol. Res. 2015, V4, 4. [CrossRef]
- 11. Liu, S.; Liu, L. Determining the Number of Groups in Selective Assembly for Remanufacturing Engine. *Procedia Eng.* **2017**, 174, 815–819. [CrossRef]
- 12. Wang, W.M.; Li, D.B.; He, F.; Tong, Y.F. Modelling and Optimization for a Selective Assembly Process of Parts with Non-Normal Distribution. *Int. J. Simul. Model.* **2018**, *17*, 133–146. [CrossRef]
- 13. Kannan, S.M.; Pandian, G.R. A New Selective Assembly Model for Achieving Specified Tolerance in High Precision Assemblies. *Int. J. Precis. Eng. Manuf.* **2020**, *21*, 1217–1230. [CrossRef]
- 14. Chu, X.; Xu, H.; Wu, X.; Tao, J.; Shao, G. The method of selective assembly for the RV reducer based on genetic algorithm. *ARCHIVE Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* 2018, 232, 921–929. [CrossRef]
- Jeevanantham, A.K.; Chaitanya, S.V.; Rajeshkannan, A. Tolerance Analysis in Selective Assembly of Multiple Component Features to Control Assembly Variation Using Matrix Model and Genetic Algorithm. *Int. J. Precis. Eng. Manuf.* 2019, 20, 1801–1815. [CrossRef]
- 16. Kannan, S.M.; Sivasubramanian, R.; Jayabalan, V. Particle swarm optimization for minimizing assembly variation in selective assembly. *Int. J. Adv. Manuf. Technol.* 2009, 42, 793–803. [CrossRef]
- 17. Raj, M.V.; Sankar, S.S.; Ponnambalam, S.G. Particle swarm optimization algorithm to maximize assembly efficiency. *Int. J. Adv. Manuf. Technol.* **2011**, *59*, 719–736. [CrossRef]
- 18. Babu, J.R.; Asha, A. Minimising assembly loss for a complex assembly using Taguchi's concept in selective assembly. *Int. J. Product. Qual. Manag.* **2015**, 15, 335. [CrossRef]
- 19. Babu, J.R.; Asha, A. Modelling in selective assembly with symmetrical interval-based Taguchi loss function for minimising assembly loss and clearance variation. *Int. J. Manuf. Technol. Manag.* **2015**, *29*, 288–308. [CrossRef]

- Asha, A.; Kannan, S.; Jayabalan, V. Optimization of clearance variation in selective assembly for components with multiple characteristics. *Int. J. Adv. Manuf. Technol.* 2008, 38, 1026–1044. [CrossRef]
- Raj, M.V.; Sankar, S.S.; Ponnambalam, S.G. Minimizing clearance variations and surplus parts in multiple characteristic radial assembly through batch selective assembly. *Int. J. Adv. Manuf. Technol.* 2011, 57, 1199–1222.
- Sun, W.; Mu, X.; Sun, Q.; Jiang, Y. The grouping and matching of three dimensional deviation based on homogeneous transformation and Taguchi theory. J. Adv. Mech. Des. Syst. Manuf. 2017, 11, JAMDSM0010. [CrossRef]
- Xing, M.; Zhang, Q.; Jin, X.; Zhang, Z. Optimization of Selective Assembly for Shafts and Holes Based on Relative Entropy and Dynamic Programming. *Entropy* 2020, 22, 1211. [CrossRef]
- Aderiani, A.R.; Wärmefjord, K.; Söderberg, R.; Lindkvist, L. Developing a selective assembly technique for sheet metal assemblies. *Int. J. Prod. Res.* 2019, 57, 7174–7188. [CrossRef]
- 25. Aderiani, A.R.; Wärmefjord, K.; Söderberg, R. An Improved Phenotype-Genotype Mapping for Solving Selective Assembly Problem Using Evolutionary Optimization Algorithms. *J. Comput. Inf. Sci. Eng.* **2020**, *20*, 20.
- 26. Borgne, A.L.; Belloir, N.; Bruel, J.M.; Nguyen, T. Formal Requirements Engineering for Smart Industries: Toward a Model-Based Graphical Language. In Proceedings of the 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), Toulouse, France, 18–21 July 2016; pp. 1028–1032.
- Bouskela, D.; Falcone, A.; Garro, A.; Jardin, A.; Otter, M.; Thuy, N.; Tundis, A. Formal requirements modeling for cyber-physical systems engineering: An integrated solution based on FORM-L and Modelica. *Requir. Eng.* 2021, 1–30. [CrossRef]
- Wang, K.; Zhang, S.; He, W.; Wang, M.; Chen, J.; Wang, L. Selective Assembly of Complicated Mechanical Product Based on SPEA2. J. Shanghai Jiaotong Univ. 2016, 50, 1047–1053.
- 29. Antony, J. Simultaneous Optimisation of Multiple Quality Characteristics in Manufacturing Processes Using Taguchi's Quality Loss Function. *Int. J. Adv. Manuf. Technol.* 2001, 17, 134–138. [CrossRef]
- 30. Bai, Y.; Wang, H.; Wang, L.; Tian, K. Study and analysis on fuzzy quality control for the high-end manufacturing process based on Taguchi quality loss function. *J. Comput. Methods Sci. Eng.* **2018**, *19*, 121–136. [CrossRef]
- Deb, K.; Jain, H. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Trans. Evol. Comput.* 2014, 18, 577–601. [CrossRef]
- Jain, H.; Deb, K. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach. *IEEE Trans. Evol. Comput.* 2014, 18, 602–622. [CrossRef]
- 33. Srinivas, N.; Deb, K. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evol. Comput.* **1994**, 2, 221–248. [CrossRef]
- 34. Das, I.; Dennis, J.E. Normal-Boundary Intersection: A New Method for Generating the Pareto Surface in Nonlinear Multicriteria Optimization Problems. *SIAM J. Optim.* **1996**, *8*, 631–657. [CrossRef]
- 35. Opricovic, S.; Tzeng, G.H. Extended VIKOR method in comparison with outranking methods. *Eur. J. Oper. Res.* 2007, 178, 514–529. [CrossRef]
- Bai, H.; Feng, F.; Wang, J.; Wu, T. A Combination Prediction Model of Long-Term Ionospheric foF2 Based on Entropy Weight Method. *Entropy* 2020, 22, 442. [CrossRef]
- Wang, X.; Wang, K. A multi-criteria decision-making method based on triangular interval-valued fuzzy numbers and the VIKOR method. J. Intell. Fuzzy Syst. 2021, 40, 221–233. [CrossRef]
- Ye, T.; Ran, C.; Zhang, X.; Jin, Y. PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization. *IEEE Comput. Intell. Mag.* 2017, 12, 73–87.