





Article

# Harmony Search Algorithm for Minimizing Assembly Variation in Non-linear Assembly

Siva Kumar Mahalingam <sup>1</sup>, Lenin Nagarajan <sup>1,\*</sup>, Sachin Salunkhe <sup>1</sup>, Emad Abouel Nasr <sup>2</sup>,  
Jôao Paulo Davim <sup>3</sup> and Hussein M. A. Hussein <sup>4,5</sup>

- <sup>1</sup> Department of Mechanical Engineering, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai 600062, India; lawan.sisa@gmail.com (S.K.M.); drsalunkhesachin@veltech.edu.in (S.S.)
- <sup>2</sup> Industrial Engineering Department, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia; eabdelghany@ksu.edu.sa
- <sup>3</sup> Department of Mechanical Engineering, University of Aveiro, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal; pdavim@ua.pt
- <sup>4</sup> Mechanical Engineering Department, Faculty of Engineering, Helwan University, Cairo 11732, Egypt; hussein@h-eng.helwan.edu.eg
- <sup>5</sup> Mechanical Engineering Department, Faculty of Engineering, Ahram Canadian University, Giza 12566, Egypt
- \* Correspondence: n.lenin@gmail.com

**Abstract:** The proposed work aims to acquire the maximum number of non-linear assemblies with closer assembly tolerance specifications by mating the different bins' components. Before that, the components are classified based on the range of tolerance values and grouped into different bins. Further, the manufacturing process of the components is selected from the given and known alternative processes. It is incredibly tedious to obtain the best combinations of bins and the best process together. Hence, a novel approach using the combination of the univariate search method and the harmony search algorithm is proposed in this work. Overrunning clutch assembly is taken as an example. The components of overrunning clutch assembly are manufactured with a wide tolerance value using the best process selected from the given alternatives by the univariate search method. Further, the manufactured components are grouped into three to nine bins. A combination of the best bins is obtained for the various assembly specifications by implementing the harmony search algorithm. The efficacy of the proposed method is demonstrated by showing 24.9% of cost-savings while making overrunning clutch assembly compared with the existing method. The efficacy of the proposed method is demonstrated by showing 24.9% of cost-savings while making overrunning clutch assembly compared with the existing method. The results show that the contribution of the proposed novel methodology is legitimate in solving selective assembly problems.

**Keywords:** selective assembly; overrunning clutch assembly; univariate search method; harmony search algorithm

check for  
updates

**Citation:** 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. <https://doi.org/10.3390/app11199213>

Academic Editor: Vladimir Modrak

Received: 6 September 2021

Accepted: 28 September 2021

Published: 3 October 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/>).

## 1. Introduction

Product quality is the focus of any manufacturing process. In general, two or more components are assembled to create an assembly. The quality of the assembly depends on the quality of the components, which affects the product's functionality. Tolerance plays an essential role in the component's quality, deciding the fit between the mating parts. The components manufactured with closer tolerance make the precise assembly more suitable for functional requirements. Selective assembly is one of the feasible methods for making precise assemblies with lower manufacturing costs. The complete elimination or reduction of secondary operations by forming wide-tolerance components is the reason for lower manufacturing costs. In selective assembly, the components are grouped as bins based on the uniform grouping method, the equal probability method, or the uniform tolerance method, for example. According to the best bin combinations, the precise assemblies

are made by mating the components randomly selected from the corresponding bins. Most of the research on linear/radial assembly have mainly focused on minimizing the objectives, namely, clearance variation or surplus parts. Research works focusing on non-linear assembly are seldom found. Moreover, in the existing literature, the tolerance of the components is usually considered for obtaining the best bin combinations rather than the dimension of components. Since different combinations of bins are possible, this problem environment can be treated as an NP-hard problem. Selection of process for making components from the given alternative processes also plays a vital role in further reducing the product's manufacturing cost.

## 2. Related Research

The literature relevant to the proposed work has been categorized as selective assembly and the harmony search algorithm. The literature related to both topics is detailed below.

### 2.1. Selective Assembly

Kern (2003) [1] proposed an approach for selecting assembly by considering variations in the dimensional distributions. Further, the author developed closed-form equations for the different techniques of selective assembly. Mease et al. (2004) [2] introduced a method to classify the components into various classes based on the dimensional variations. The assembly was made by pairing the selected components from the different classes. Further, the absolute and squared error loss functions were studied to select the optimum method. Matsuura et al. (2008) [3] explored a method to minimize clearance variation when pairing the components with less variance for making an assembly. Kwon et al. (2009) [4] studied the effectiveness of optimal binning strategies based on the squared error loss function by considering similar normal distributions of the dimensions of two different components. Matsuura et al. (2011) [5] developed the optimal mean shift to reduce surplus components while making selective assembly. The equal width concept was used for grouping the components into different bins. Three shifted means method was used for the fabrication of components.

Yue et al. (2014) [6] proposed a genetic algorithm to minimize the variation in clearance while mating the components available in the selective groups to manufacture hole and shaft assemblies. Babu and Asha (2014) [7] introduced a customized artificial immune system algorithm for mating the components available in selective groups to minimize assembly clearance variation. Further, they applied Taguchi's loss function to identify the deviation from the mean. The count of selective groups was also one of the important concerns in the proposed work. Ju and Li (2014) [8] solved selective assembly problems using a two-stage decomposition procedure. A two-component assembly system with unreliable Bernoulli machines and limited inventories was considered. The effectiveness of the proposed method was analyzed analytically.

Xu et al. (2014) [9] applied a new method with the combination of discarding and binning theorems to minimize the components' variations while making the selective assembly of a hard disk drive. The utilization rate of the components was high with regard to the number of assemblies made using the newly proposed method. Lu and Fei (2015) [10] proposed a grouping method in selective assembly to increase the success rate of manufacturing the assemblies and reducing the surplus parts. A genetic algorithm with a 2D chromosome structure was used for this purpose. The effectiveness of the proposed method was studied based on solving three different cases. Manickam and De (2015) [11] developed a genetic algorithm to identify the better combination of groups containing the components to create the maximum number of assemblies with less total cost. The small number of components with wider tolerance was made available in the individual groups. This was the uniqueness of the presented work. The model was developed using MATLAB software.

Babu and Asha (2015) [12] evaluated the losses in making assemblies by applying the symmetrical interval-based Taguchi loss function. Further, the dual objectives, namely, minimum clearance variation and minimum losses in making assemblies, were considered

to identify the better combination of the parts' groups using the sheep flock heredity algorithm. Ju et al. (2016) [13] studied the performance of the selective assembly system using Bernoulli machine reliability models. For this study, the assemblies made through two components with different qualities were considered. The two-stage decomposition procedure was applied to evaluate the effectiveness of the proposed models analytically. Malaichamy et al. (2016A) [14] developed software capable of applying all kinds of selective assembly methods to identify suitable solutions with the least possible time. For instance, equal area and width methods were used to minimize variation and scrap in selective assembly. The random assembly method was inbuilt in the software to select the best method for minimizing the manufacturing cost of the assembly. Since the software was developed using an optional programming language, changes to the input data were extremely easy, and, in no time, the result could be viewed graphically.

Malaichamy et al. (2016B) [15] introduced a software approach to help engineers to visualize the tolerance cost curve for a given process or a good combination of processes in no time. The tolerance cost was calculated using the reciprocal power cost model. The best process could be selected in this work from the given alternative processes for the known tolerance value. A simulated annealing algorithm was implemented to compute the best bin number and its combinations with almost nil surplus parts for the given dimensional distribution of components of a shaft housing assembly. Compared with random assembly, a good number of surplus parts were reduced by implementing an equal width selective assembly method. Liu and Liu (2017) [16] discussed the remanufacturing of the engine through the selective assembly concept. Remanufacturing is a sustainable strategy in which the group numbers and range of components in each group are not kept constant. The assembly accuracy was greatly increased using the proposed strategy. Chu et al. (2018) [17] developed a method for manufacturing gear reducers using a novel strategy called GA-based selective assembly. The backlash of the gear reducer was the major concern in this work to verify the meeting of assembly requirements.

## 2.2. Harmony Search Algorithm

Geem et al. (2001) [18] described the harmony search optimization algorithm's features, robustness, simplicity, and search efficiency in solving engineering problems. Geem et al. (2002) [19] proposed a harmony search optimization algorithm to solve the problems associated with pipe network design. Lee and Geem (2004) [20] described a novel structural optimization algorithm using the harmony search method. The effectiveness of the proposed algorithm was verified by identifying the solutions to different truss problems. Lee and Geem (2005) [21] solved different engineering optimization problems that included minimizing mathematical functions and optimizing structural problems using a harmonic-search-based method. Flow demand and low head were considered objectives in this work. Wang et al. (2009) [22] analyzed the harmony search algorithm's effectiveness with the colonial selection algorithm. The colonial search method was used to increase the harmony memory members in the harmony search method.

From the literature survey, it is clear that research works have been very limited in the manufacturing of non-linear assemblies with a closer assembly tolerance specification. Further, the usage of the harmony search algorithm for solving selective assembly problems is seldom found. The problem environment is discussed in the next section.

## 3. Problem Environment

In selective assembly, parts are manufactured with wider tolerance, measured, and partitioned into groups, and assemblies are made by assembling the components within the random combination of groups. Manufacturing tolerance, assembly specification, and the number of groups are the three main factors that play important roles in controlling surplus parts that affect the product's manufacturing cost. After parts are manufactured, it is tedious to obtain the best combination of groups for different assembly specifications. It is laborious work to compute the numbers of closer assembly for each bin number from

the manufactured components. Alternative process selection for making components also makes the problem highly complicated. The problems described above are challenging to process/design engineers and reduce implementation in real situations. Moreover, the components involved in making a non-linear assembly require the individual mating of dimensions for each assembly. Compared to making linear assembly, this required additional computation effort is tedious.

#### 4. Solution Methodology

The proposed problem environment can be solved in three stages. In the first stage, the best process for manufacturing each component of an assembly is obtained from the alternative processes using the univariate search method. In the second stage, 1000 simulated dimensions of each component are generated using MATLAB for the different combinations of alternative processes obtained from the first stage. Further, the components are partitioned into different bin numbers according to the equal area method, which is one of the techniques used in the selective assembly method. In the last stage, the harmony search algorithm is implemented to obtain the best bin combinations. Then, the non-linear assemblies are made by mating the components according to the best bin combinations with almost nil surplus parts. To show the effectiveness of the proposed method, the manufacturing cost of assemblies made both from the random assembly [23] and selective assembly are compared. This three-stage approach is explained in a detailed way in the numerical illustration section.

#### 5. Numerical Illustration

Overrunning clutch assembly (OCA), dealt with by Ganesan et al. (2001) [23], shown in Figure 1, is considered an example product to show the efficiency of the proposed method. It consists of a hub, four rollers, and a cage. The nominal dimensions and their allocated tolerance; the minimum, maximum process tolerances; and the tolerance cost function constants of alternative processes of each component are listed in Table 1. The critical dimension  $Y$  is determined using Equation (1), and the accepted value of  $Y$  is assumed to be  $7.0124 \pm 2^\circ$ . Equation (2) represents the reciprocal tolerance cost function to calculate the component's tolerance cost. The cost function constants are taken from Ganesan et al. (2001) [23]. The cost of assembly based on allocated and maximum process tolerances is computed using Equations (3) and (4), respectively, and listed in Table 2.

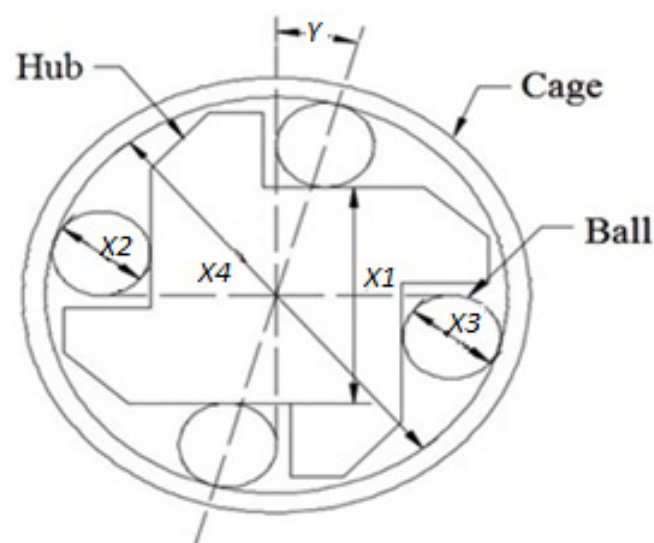


Figure 1. Overrunning clutch assembly.

$$Y = \cos^{-1} \left( \frac{X1 + \left( \frac{X2+X3}{2} \right)}{X4 - \left( \frac{X2+X3}{2} \right)} \right) \tag{1}$$

$$C_i = A_i + \frac{B_i}{t_i} \tag{2}$$

$$C_a = \sum_{i=1}^{nc} A_i + \frac{B_i}{t_{ai}} \tag{3}$$

$$C_m = \sum_{i=1}^{nc} A_i + \frac{B_i}{t_{mi}} \tag{4}$$

where

- X1 —Dimension of the hub in mm
- X2 and X3 —Dimension of the rollers in mm
- X4 —Dimension of the cage in mm
- Y —Critical dimension in degrees
- A<sub>i</sub> —Fixed cost of the *i*th component in \$
- B<sub>i</sub> —Cost function constants of the *i*th component
- t<sub>i</sub> —*i*th component's tolerance in mm
- C<sub>i</sub> —Cost of the *i*th component in \$
- t<sub>ai</sub>, t<sub>mi</sub> —*i*th component's allocated and maximum tolerance in mm
- C<sub>a</sub>, C<sub>m</sub> —Cost of assembly in \$ based on t<sub>ai</sub> and t<sub>mi</sub>
- nc —Number of components
- i* —Component number index

**Table 1.** Manufacturing details of overrunning clutch assembly.

C.No.	Nominal Dimension (D <sub>i</sub> ) (mm)	t <sub>ai</sub> (mm)	P <sub>s</sub>	P.No.	A <sub>i</sub>	B <sub>i</sub>	t <sub>min</sub> (mm)	t <sub>max</sub> (mm)
X1	55.29	0.179806	3	P1	10.0	0.015	0.015	0.08
				P2	5.0	0.500	0.060	0.15
				P3	3.5	0.750	0.120	0.25
X2	22.86	0.165358	2	P1	8.0	0.250	0.020	0.15
				P2	3.0	0.650	0.080	0.30
X3	22.86	0.120132	1	P1	2.5	0.300	0.040	0.20
				P2	5.0	0.045	0.120	0.25
X4	101.69	0.200581	3	P1	4.0	0.560	0.080	0.12
				P2	6.0	0.160	0.150	0.25
				P3	0.5	0.880	0.200	0.40

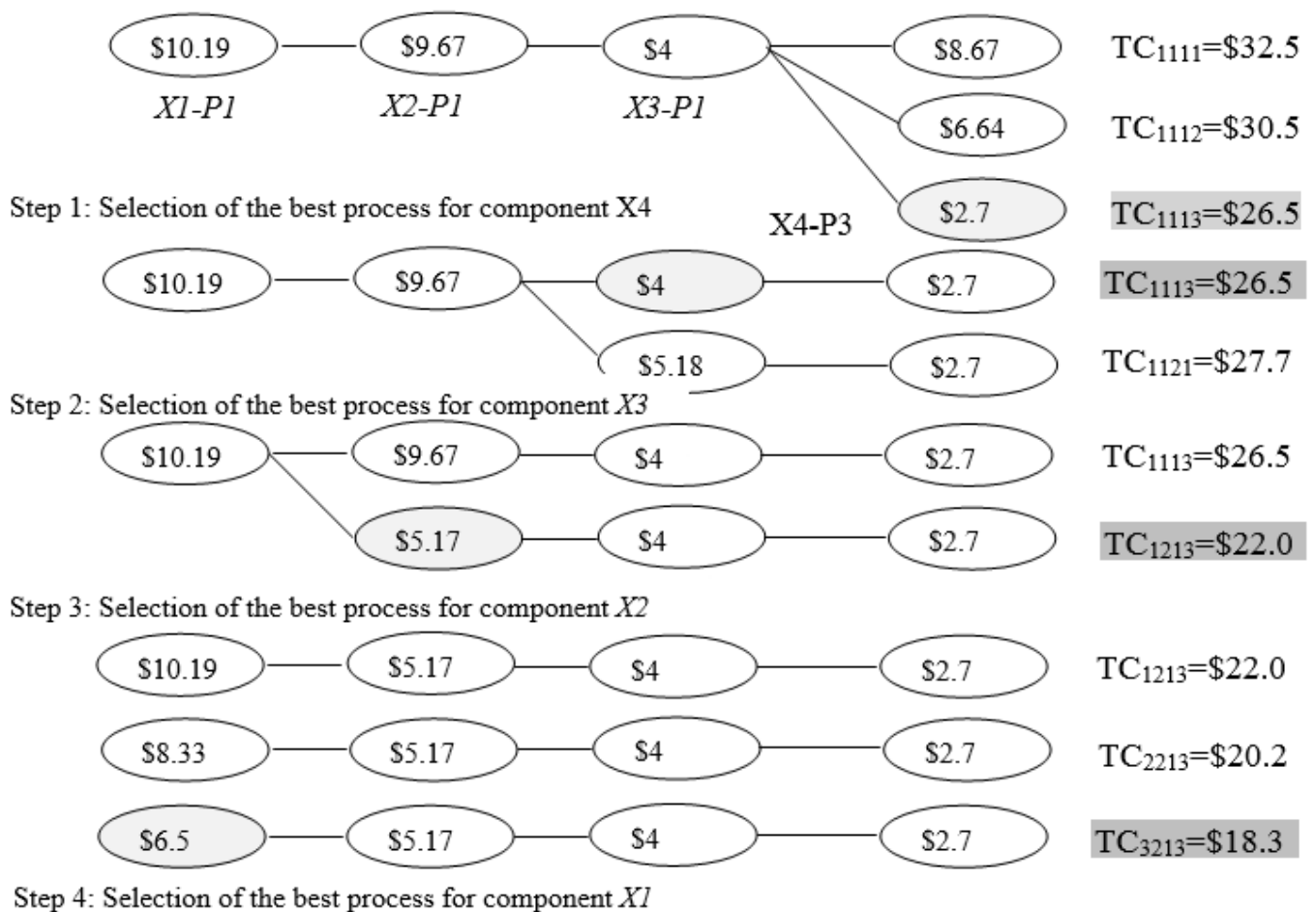
C.No.—component name; P.No.—process number.

**Table 2.** Manufacturing cost of overrunning clutch assembly for t<sub>ai</sub>, t<sub>min</sub>, and t<sub>max</sub>.

C.No.	C <sub>ai</sub> (\$)	P.No.	C <sub>min</sub> (\$)	C <sub>max</sub> (\$)
X1	7.67	P1	11.00	10.19
		P2	13.33	8.33
		P3	9.75	6.50
X2	6.93	P1	20.50	9.67
		P2	11.13	5.17
X3	5.00	P1	10.00	4.00
		P2	5.38	5.18
X4	4.89	P1	11.00	8.67
		P2	7.07	6.64
		P3	4.90	2.70

5.1. Stage I

The selection of the best process for each component from the given alternative processes using the univariate search method is illustrated in Figure 2. It is understood from Figure 2 that the minimum manufacturing cost of \$18.37 can be achieved through making the components X1, X2, X3, and X4 using process combinations (3213) P3, P2, P1, and P3, respectively, which is nearly 24.98% less compared with the \$24.49 reported in the existing method dealt by Ganesan et al. (2001). It is also understood that the process combinations of 2213 and 1213 for components X1, X2, X3, and X4, respectively, can yield 17.52% and 9.92% savings in manufacturing cost. However, in a real situation, the savings may vary slightly because of surplus parts present in the selective assembly method.



**Figure 2.** Univariate search method to select the best process for each component. X1-P1 indicates that X1 component is produced by the P1 process; TC1113 = \$26.56 indicates that components X1, X2, X3, and X4 are manufactured using processes P1, P1, P1, and P3, respectively, and the total cost to manufacture the same will be \$26.56.

5.2. Stage II

As discussed in Section 4, 1000 random values have been generated for each component according to the mean ( $\mu_i$ ) and standard deviation ( $\sigma_i$ ) presented in Table 3. The dimensional distribution of 1000 components of X1, X2, X3, and X4 was generated using the normrnd (C.No.) function in MATLAB.

**Table 3.** Mean and standard deviation of components for different process combinations.

C.No.	$\mu_i$ (mm)	PC1–1213			PC2–2213			PC3–3213		
		P.No.	$t_{mi}$ (mm)	$\sigma_i = t_{mi}/3$	P.No.	$t_{mi}$ (mm)	$\sigma_i = t_{mi}/3$	P.No.	$t_{mi}$ (mm)	$\sigma_i = t_{mi}/3$
X1	55.29	P1	0.08	0.02667	P2	0.15	0.05	P3	0.25	0.08333
X2	22.86	P2	0.3	0.1	P2	0.3	0.1	P2	0.3	0.1
X3	22.86	P1	0.25	0.08333	P1	0.25	0.08333	P1	0.25	0.08333
X4	101.69	P3	0.4	0.13333	P3	0.4	0.13333	P3	0.4	0.13333

*P.No.*—process number; *PC1*—1st process combinations for components X1, X2, X3, and X4;  $\mu_i$ —mean dimension of the components;  $t_{mi}$ —tolerance of the *i*th component;  $\sigma_i$ —standard deviation of the *i*th component.

### 5.3. Stage III—Implementation of HSA

The harmony search algorithm (HSA) is a meta-heuristic algorithm, and it works based on the identification of good harmony by musicians through a continuous improvisation process. The HSA has the following advantages: (i) quick convergence, (ii) easy to adapt, and (iii) the least computational time. Further, from the literature survey, it is observed that the HSA has outperformed in solving complex optimization problems. Hence, the HSA has been used in this work. Table 4 presents the different terms used in the HSA, its equivalent term in both optimization problems and the present work formulation, and its range of values and examples. The schematic diagram shown in Figure 3 illustrates the implementation of the HSA to obtain the best bin combinations. The technical terms and their meanings used in Figure 3 are presented in Table 5. For demonstration purposes, the components are partitioned into five bins. The step-by-step procedure is given below.

**Table 4.** Representation of variables in HSA.

HSA Parameters	Equivalent Term in Optimization Problem	Equivalent Term in Present Work	Example
Musical Instrument Pitch Range	Decision Variable Value Range	No. of components Number of bins	X1, X2, X3, and X4 5 (X1) 13245
Harmony	A Solution Vector	A combination of bins for each component	(X2) 43251 (X3) 42135 (X4) 52143
Aesthetics	Objective Function	Number of accepted assemblies	985
Practice	Iteration	No. of iterations	150
Experience	Memory Matrix	Storing of the best solution	
Harmony Memory Size	Size of Solution Vector	No. of initial solutions	20
HMCR (0.7–0.99)	Harmony Memory Considering Rate	0.7	0.7
PAR	Pitch Adjusting Rate	0.3	0.3

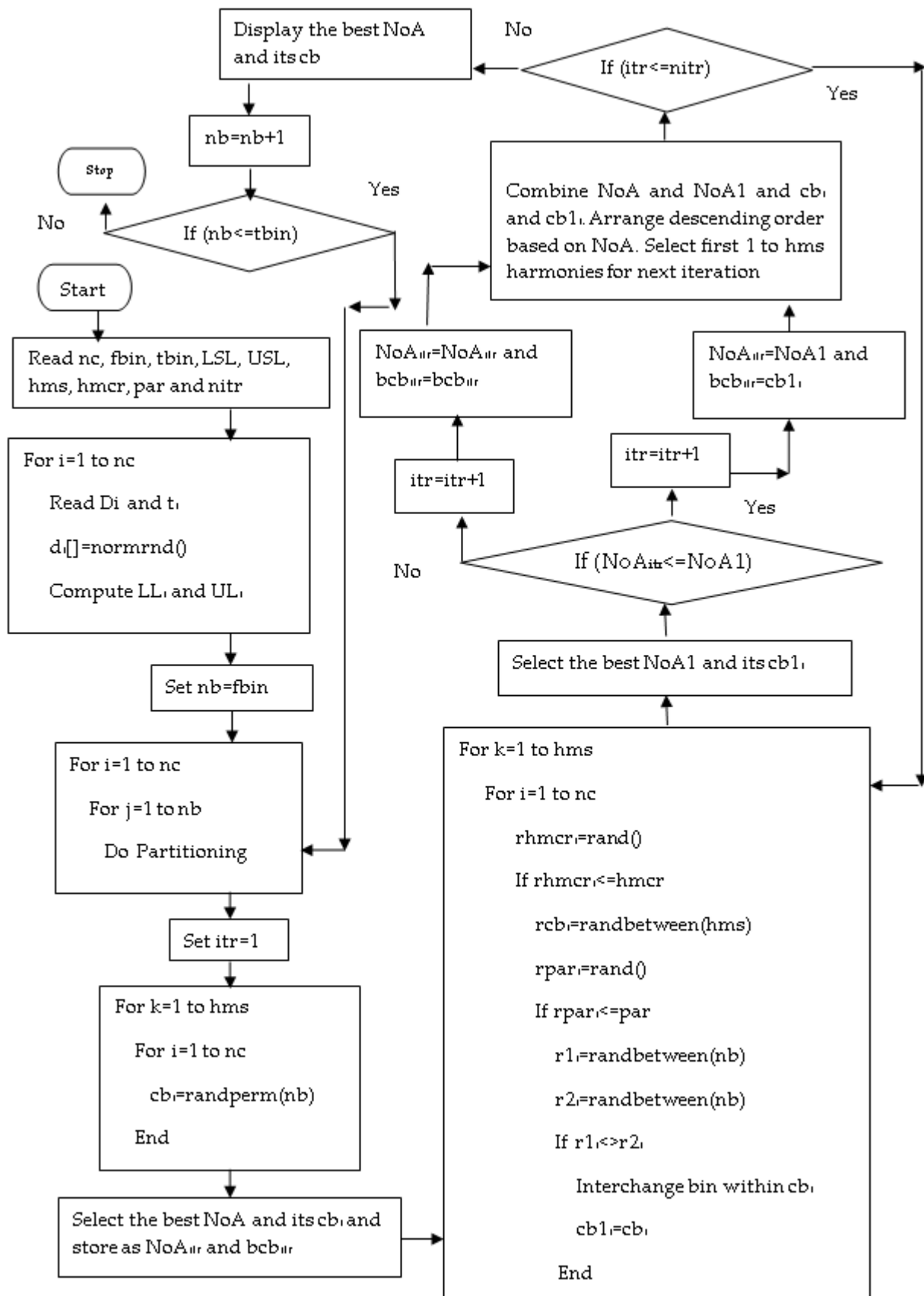


Figure 3. Implementation of HSA.



**Table 5.** Technical terms and their meanings.

Terms	Meaning
Trimming	Remove the components which do not satisfy $LL_i \leq d_i \leq UL_i$
Partitioning	Arranging the components in ascending order based on its $d_i$ and placing an equal number of components in each bin starting from bin number 1 to $nb$
Mating	Selecting one component ( $X1, X2, X3,$ and $X4$ ) randomly from the bin based on the bin combination of $cb_i$ and making an assembly. Check the assembly’s specification (within $LSL$ and $USL$ ); if it meets the requirement, then treat it as good assembly ( $NoA$ ); otherwise, treat it as a surplus part
<i>normrnd</i>	Matlab function will generate the required number of normally distributed random numbers based on the given mean and standard deviation value
<i>randperm (nb)</i>	Matlab function will generate a permutation combination for the given bin number $nb$

- Step 1: A random combination of bins for each component for the size of harmony memory 10 is generated (listed in Table 6).
- Step 2: The corresponding bin’s components are randomly matched with other components for each harmony number, and the assembly is produced. If the assembly meets the given specification limit, it is accepted as an assembly; otherwise, it is treated as a surplus part. This will be carried out until the component exists in each bin. The accepted assemblies are counted and listed in Table 6 as NoA.
- Step 3: Table 7 represents the arrangements of harmony, from maximum NoA to minimum NoA. Table 8 illustrates the best bin combination that will produce maximum NoA in the first iteration.

**Table 6.** Initial harmony memory.

H.No.(k)	$cb_{X1}$	$cb_{X2}$	$cb_{X3}$	$cb_{X4}$	$NoA_k$
1	2 5 1 4 3	2 5 3 1 4	5 3 4 1 2	3 1 2 4 5	308
2	4 2 1 5 3	4 2 5 1 3	4 2 5 3 1	5 3 4 2 1	649
3	4 2 1 5 3	1 5 2 3 4	3 4 2 5 1	3 1 5 4 2	407
4	4 3 1 5 2	3 2 4 5 1	5 2 3 4 1	3 5 2 4 1	508
5	1 5 4 3 2	1 3 4 5 2	3 2 4 5 1	2 4 1 3 5	326
6	2 1 3 4 5	3 2 4 5 1	3 1 5 2 4	3 5 2 4 1	557
7	5 2 3 1 4	2 5 3 1 4	1 2 3 4 5	1 5 4 2 3	354
8	1 2 4 3 5	1 3 5 2 4	5 2 4 3 1	5 3 4 2 1	397
9	2 3 1 4 5	3 1 2 5 4	2 5 1 4 3	5 4 2 1 3	212
10	4 3 1 2 5	4 5 3 2 1	3 2 1 4 5	3 2 4 5 1	458

H.No.—harmony numbers;  $cb_{X1}, cb_{X2}, cb_{X3},$  and  $cb_{X4}$  are the combinations of bins for  $X1, X2, X3,$  and  $X4$  components.

**Table 7.** Harmonies after sorting based on NoA.

H.No.(k)	$cb_{X1}$	$cb_{X2}$	$cb_{X3}$	$cb_{X4}$	$NoA_k$
1	4 2 1 5 3	4 2 5 1 3	4 2 5 3 1	5 3 4 2 1	649
2	2 1 3 4 5	3 2 4 5 1	3 1 5 2 4	3 5 2 4 1	557
3	4 3 1 5 2	3 2 4 5 1	5 2 3 4 1	3 5 2 4 1	508
4	4 3 1 2 5	4 5 3 2 1	3 2 1 4 5	3 2 4 5 1	458
5	4 2 1 5 3	1 5 2 3 4	3 4 2 5 1	3 1 5 4 2	407
6	1 2 4 3 5	1 3 5 2 4	5 2 4 3 1	5 3 4 2 1	397
7	5 2 3 1 4	2 5 3 1 4	1 2 3 4 5	1 5 4 2 3	354
8	1 5 4 3 2	1 3 4 5 2	3 2 4 5 1	2 4 1 3 5	326
9	2 5 1 4 3	2 5 3 1 4	5 3 4 1 2	3 1 2 4 5	308
10	2 3 1 4 5	3 1 2 5 4	2 5 1 4 3	5 4 2 1 3	212

**Table 8.** Best combination of bins for the first iteration.

<i>bc<sub>itr</sub></i>				<i>NoA<sub>itr</sub></i>
<i>cb<sub>X1</sub></i>	<i>cb<sub>X2</sub></i>	<i>cb<sub>X3</sub></i>	<i>cb<sub>X4</sub></i>	
4 2 1 5 3	4 2 5 1 3	4 2 5 3 1	5 3 4 2 1	649

Step 4: A random number less than 1 is generated for each component of each harmony number (presented in Table 9 as *RHMCR*).

Step 5: If this number is less than or equal to *HMCR*, then a random number between 1 and *HMS* (*rcb*) is generated, and others are assumed to be zero. This is presented in Table 9.

**Table 9.** *RHMCR* and *rcb* values.

<i>H.No.</i>	<i>RHMCR</i>				<i>rcb</i>			
	<i>X1</i>	<i>X2</i>	<i>X3</i>	<i>X4</i>	<i>rcb<sub>X1</sub></i>	<i>rcb<sub>X2</sub></i>	<i>rcb<sub>X3</sub></i>	<i>rcb<sub>X4</sub></i>
1	0.0697	0.6581	0.1267	0.0176	1	6	7	7
2	0.1398	0.9349	0.9424	0.3936	8	0	0	1
3	0.4975	0.3145	0.218	0.8001	1	9	3	0
4	0.0639	0.7984	0.1875	0.442	7	0	3	4
5	0.9388	0.0765	0.605	0.1321	0	8	1	1
6	0.2711	0.4453	0.7794	0.9148	9	3	0	0
7	0.6946	0.413	0.216	0.0049	1	6	6	1
8	0.0204	0.9048	0.0267	0.7212	1	0	8	0
9	0.0013	0.4189	0.1216	0.8414	1	6	1	0
10	0.6254	0.8232	0.0301	0.6189	2	0	8	7

*RHMCR*—a random number decides either to accept or not accept the selection of the existing tune/bin combination of components for improvisation; *rcb*—a random number generated between 1 and *hms* for each variable.

Step 6: The *cb<sub>X1</sub>*, *cb<sub>X2</sub>*, *cb<sub>X3</sub>*, and *cb<sub>X4</sub>* values corresponding to *rcb<sub>X1</sub>*, *rcb<sub>X2</sub>*, *rcb<sub>X3</sub>*, and *rcb<sub>X4</sub>* are taken from Table 6 and listed in Table 10. If the value of *rcb* is zero, then the corresponding harmony's *cbx* value is considered.

**Table 10.** Harmony after *HMCR*.

<i>H.No.(k)</i>	<i>cb<sub>X1</sub></i>	<i>cb<sub>X2</sub></i>	<i>cb<sub>X3</sub></i>	<i>cb<sub>X4</sub></i>
1'	4 2 1 5 3	1 3 5 2 4	1 2 3 4 5	1 5 4 2 3
2'	1 5 4 3 2	3 2 4 5 1	3 1 5 2 4	5 3 4 2 1
3'	4 2 1 5 3	2 5 3 1 4	5 2 3 4 1	3 5 2 4 1
4'	5 2 3 1 4	4 5 3 2 1	5 2 3 4 1	3 2 4 5 1
5'	4 2 1 5 3	1 3 4 5 2	4 2 5 3 1	5 3 4 2 1
6'	2 5 1 4 3	3 2 4 5 1	5 2 4 3 1	5 3 4 2 1
7'	4 2 1 5 3	1 3 5 2 4	5 2 4 3 1	5 3 4 2 1
8'	4 2 1 5 3	1 3 4 5 2	3 2 4 5 1	2 4 1 3 5
9'	4 2 1 5 3	1 3 5 2 4	4 2 5 3 1	3 1 2 4 5
10'	2 1 3 4 5	3 2 4 5 1	3 2 4 5 1	1 5 4 2 3

Step 7: A random number (*RPAR*) less than 1 is generated for each value of *RHMCR*, which is not equal to zero and is less than the *HMCR* value (listed in Table 11).

Step 8: Two random numbers, *r1* and *r2*, within bin numbers, are generated for each value of *RPAR*, which is not equal to zero (presented in Table 11).

Step 9: New harmony, i.e., *cb<sub>X1</sub>*, *cb<sub>X2</sub>*, *cb<sub>X3</sub>*, and *cb<sub>X4</sub>*, is obtained wherever the bin is located within their bin combinations, according to *r1* and *r2* (presented in Table 12).

Step 10: Then, *NoAs* are obtained by mating the components randomly corresponding to the bin combinations given in *cb<sub>X1</sub>*, *cb<sub>X2</sub>*, *cb<sub>X3</sub>*, and *cb<sub>X4</sub>* (listed in Table 12).

Step 11: The selection and selected harmonies for the next iteration are presented in Tables 13 and 14.

Step 12: The steps from 4 to 11 are repeated to the specified number of iterations.

Step 13: The above steps, from 1 to 12, can be repeated for various bin numbers, starting from 3 to 9. Figures 4 and 5 represent the iteration number vs. *NoA* and the number of bin vs. *NoA* for various bin numbers.

Step 14: By changing the product specification, starting from  $\pm 0.25^\circ$  to  $\pm 2^\circ$ , the above steps from 1 to 13 can be repeated.

**Table 11.** *RPAR*, *r1*, and *r2* values.

<i>H.No.</i>	<i>RPAR</i>				<i>r1</i>				<i>r2</i>			
	<i>X1</i>	<i>X2</i>	<i>X3</i>	<i>X4</i>	<i>r1<sub>X1</sub></i>	<i>r1<sub>X2</sub></i>	<i>r1<sub>X3</sub></i>	<i>r1<sub>X4</sub></i>	<i>r2<sub>X1</sub></i>	<i>r2<sub>X2</sub></i>	<i>r2<sub>X3</sub></i>	<i>r2<sub>X4</sub></i>
1'	0.0917	0.0738	0.4457	0.8808	3	1	0	0	1	2	0	0
2'	0.2529	0	0	0.6409	1	0	0	0	3	0	0	0
3'	0.5235	0.009	0.344	0	0	1	0	0	0	4	0	0
4'	0.6034	0	0.3208	0.6256	0	0	0	0	0	0	0	0
5'	0	0.1307	0.599	0.4411	0	2	0	0	0	2	0	0
6'	0.5166	0.4696	0	0	0	0	0	0	0	0	0	0
7'	0.5402	0.5435	0.8627	0.7751	0	0	0	0	0	0	0	0
8'	0.3204	0	0.7643	0	0	0	0	0	0	0	0	0
9'	0.1678	0.277	0.0321	0	1	3	1	0	3	3	3	0
10'	0.2417	0	0.8962	0.4639	4	0	0	0	1	0	0	0

*RPAR*—a random number decides the pitch adjustment (interchanging bin numbers within the component); *r1* and *r2*—two random numbers within the bin number (*nb*) to interchange the bin number to generate a new bin combination for a component.

**Table 12.** Harmony after *HMCR* and *PAR*.

<i>H.No.</i>	After <i>HMCR</i> and <i>PAR</i>				<i>NoA</i>
	<i>cb1<sub>X1</sub></i>	<i>cb1<sub>X2</sub></i>	<i>cb1<sub>X3</sub></i>	<i>cb1<sub>X4</sub></i>	
1'	1 2 4 5 3	3 1 5 2 4	1 2 3 4 5	1 5 4 2 3	709
2'	4 5 1 3 2	2 5 4 3 1	4 1 5 3 2	5 3 4 2 1	608
3'	4 2 1 5 3	1 5 3 2 4	5 2 3 4 1	3 4 1 5 2	671
4'	5 2 3 1 4	1 5 2 3 4	5 2 3 4 1	3 2 4 5 1	555
5'	2 1 5 3 4	1 3 4 5 2	4 2 5 3 1	5 3 4 2 1	503
6'	2 5 1 4 3	3 2 4 5 1	2 1 5 4 3	1 3 5 2 4	312
7'	4 2 1 5 3	1 3 5 2 4	5 2 4 3 1	5 3 4 2 1	519
8'	4 2 1 5 3	2 4 5 3 1	3 2 4 5 1	5 3 4 2 1	404
9'	1 2 4 5 3	1 3 5 2 4	5 2 4 3 1	3 2 5 1 4	460
10'	4 1 3 2 5	1 4 3 5 2	3 2 4 5 1	1 5 4 2 3	549

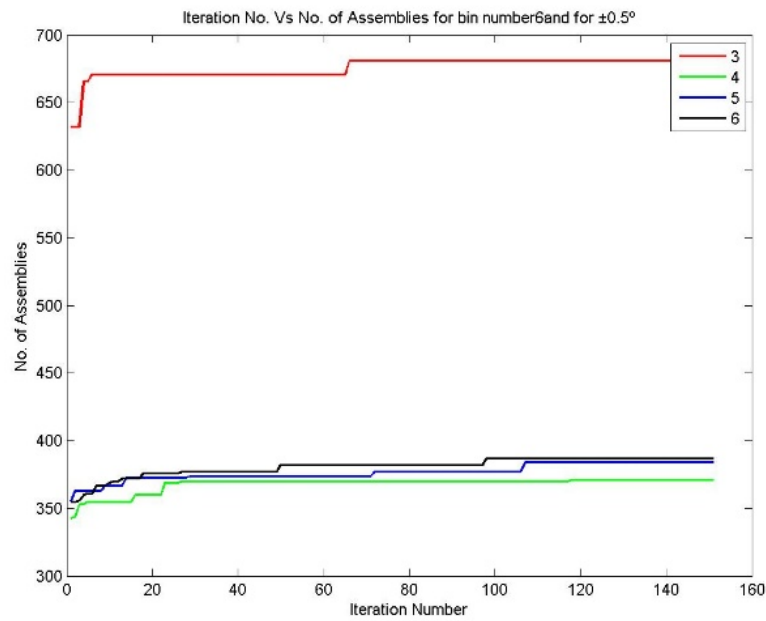
**Table 13.** Selection of harmonies for next iteration.

Before Sorting		After Sorting		<i>SHMNo.</i>
<i>HMNo.</i>	<i>NoA</i>	<i>HMNo.</i>	<i>NoA</i>	
1	649	1'	709	1''
2	557	2'	671	2''
3	508	1	649	3''
4	458	3'	608	4''
5	407	2	557	5''
6	397	4'	555	6''
7	354	5'	549	7''
8	326	6'	519	8''
9	308	3	508	9''
10	212	7'	503	10''
1'	709	8'	460	NS
2'	671	4	458	NS
3'	608	5	407	NS
4'	555	9'	404	NS
5'	549	6	397	NS
6'	519	7	354	NS
7'	503	8	326	NS
8'	460	10'	312	NS
9'	404	9	308	NS
10'	312	10	212	NS

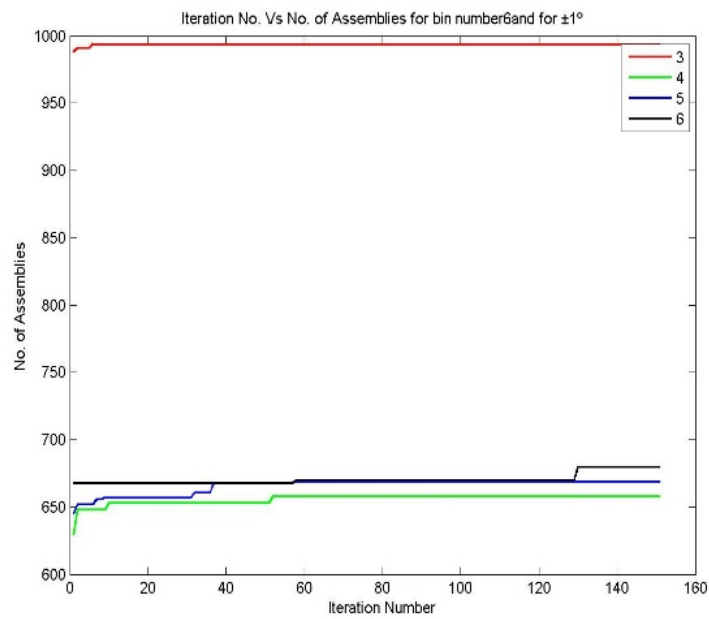
*NS*—not selected; *SHMNo.*—selected harmonies for next iteration.

**Table 14.** Selected harmonies for next iteration.

<i>IHMno</i>	<i>HMNo</i>	<i>X1</i>	<i>X2</i>	<i>X3</i>	<i>X4</i>	<i>NoA</i>
1''	1'	1 2 4 5 3	3 1 5 2 4	1 2 3 4 5	1 5 4 2 3	709
2''	2'	4 2 1 5 3	1 5 3 2 4	5 2 3 4 1	3 4 1 5 2	671
3''	1	4 2 1 5 3	4 2 5 1 3	4 2 5 3 1	5 3 4 2 1	649
4''	3'	4 5 1 3 2	2 5 4 3 1	4 1 5 3 2	5 3 4 2 1	608
5''	2	2 1 3 4 5	3 2 4 5 1	3 1 5 2 4	3 5 2 4 1	557
6''	4'	5 2 3 1 4	1 5 2 3 4	5 2 3 4 1	3 2 4 5 1	555
7''	5'	4 1 3 2 5	1 4 3 5 2	3 2 4 5 1	1 5 4 2 3	549
8''	6'	4 2 1 5 3	1 3 5 2 4	5 2 4 3 1	5 3 4 2 1	519
9''	3	4 3 1 5 2	3 2 4 5 1	5 2 3 4 1	3 5 2 4 1	508
10''	7'	2 1 5 3 4	1 3 4 5 2	4 2 5 3 1	5 3 4 2 1	503

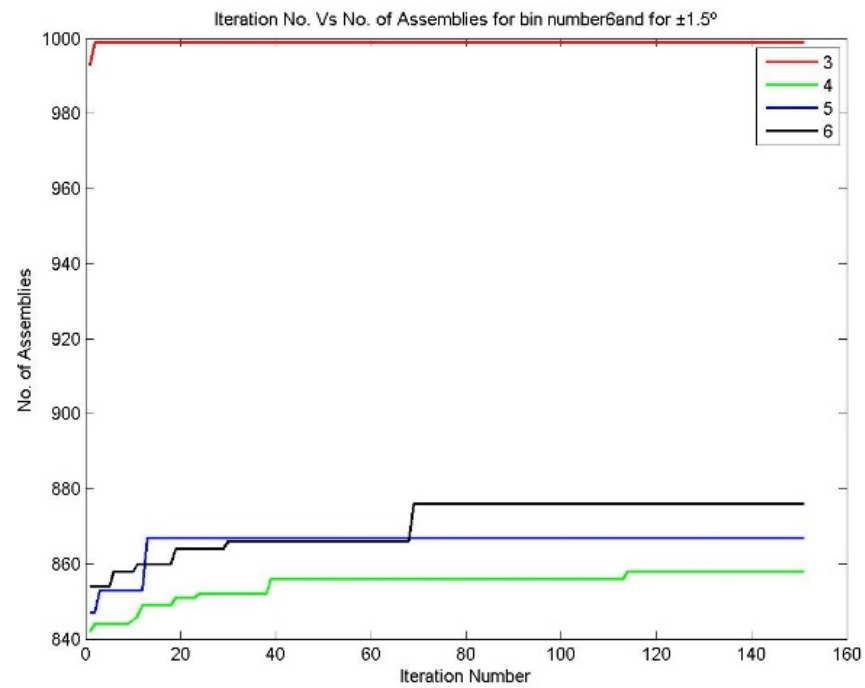


(a). *NoA* for  $\pm 0.5^\circ$

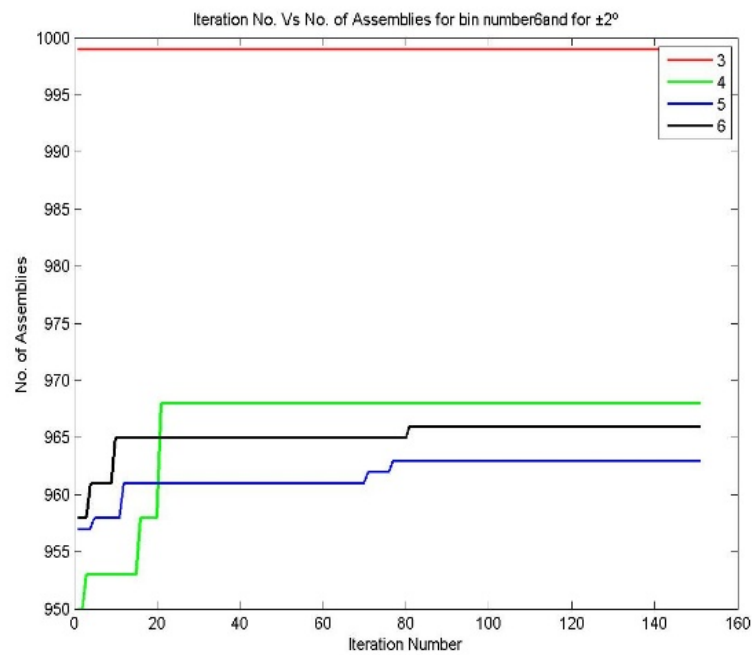


(b). *NoA* for  $\pm 1^\circ$

**Figure 4.** *Cont.*

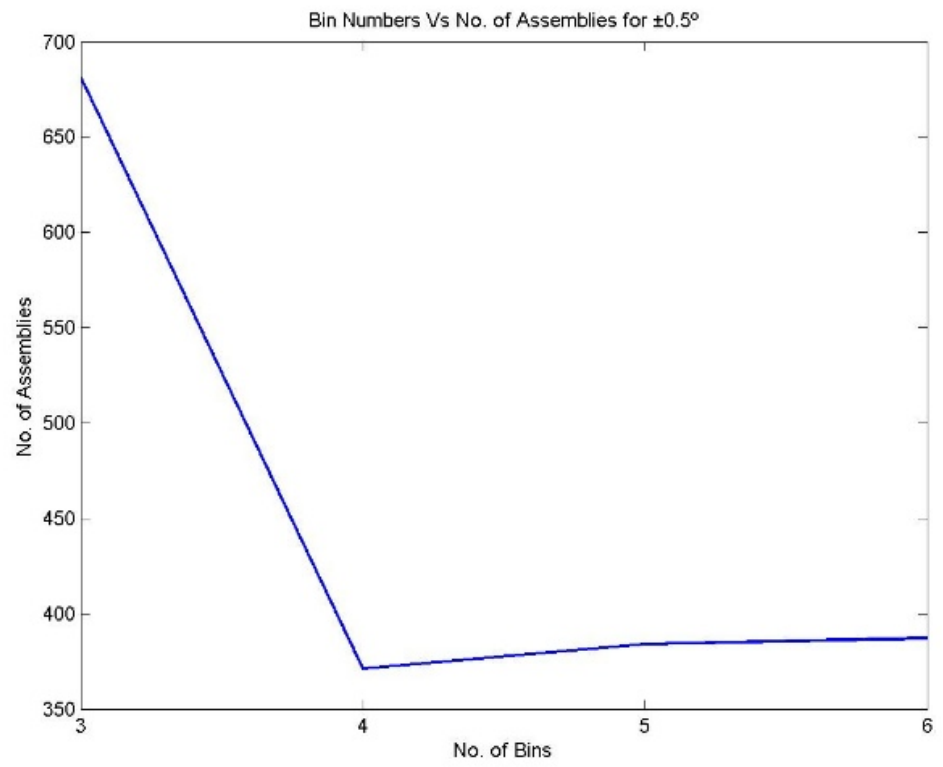


(c). NoA for  $\pm 1.5^\circ$

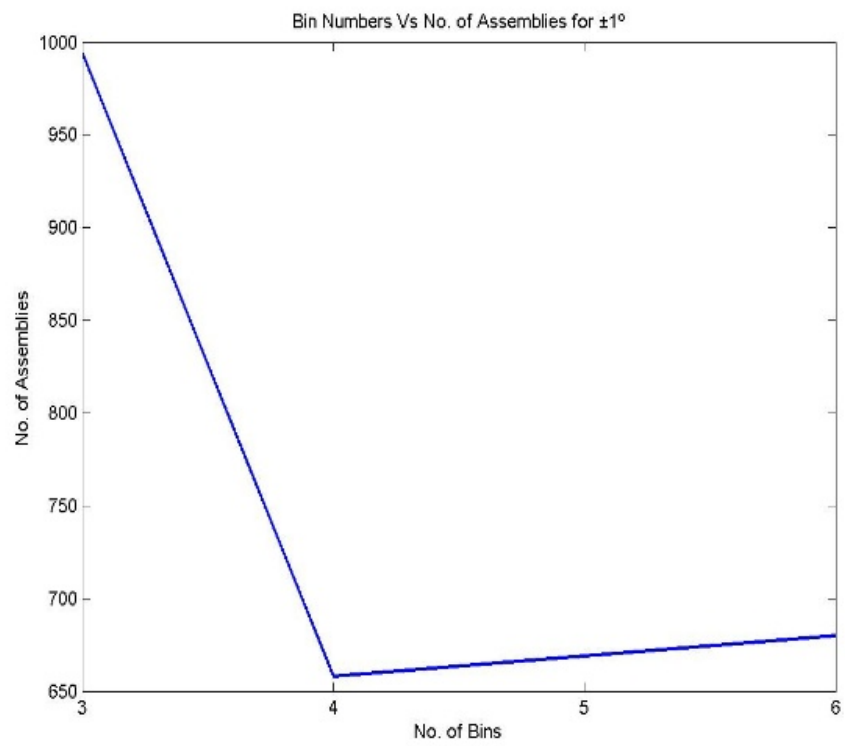


(d). NoA for  $\pm 2^\circ$

**Figure 4.** Iteration number vs. NoA for various bin numbers—the 3213 process combination (a) NoA for  $\pm 0.5^\circ$ , (b) NoA for  $\pm 1^\circ$ , (c). NoA for  $\pm 1.5^\circ$  and (d) NoA for  $\pm 2^\circ$ .

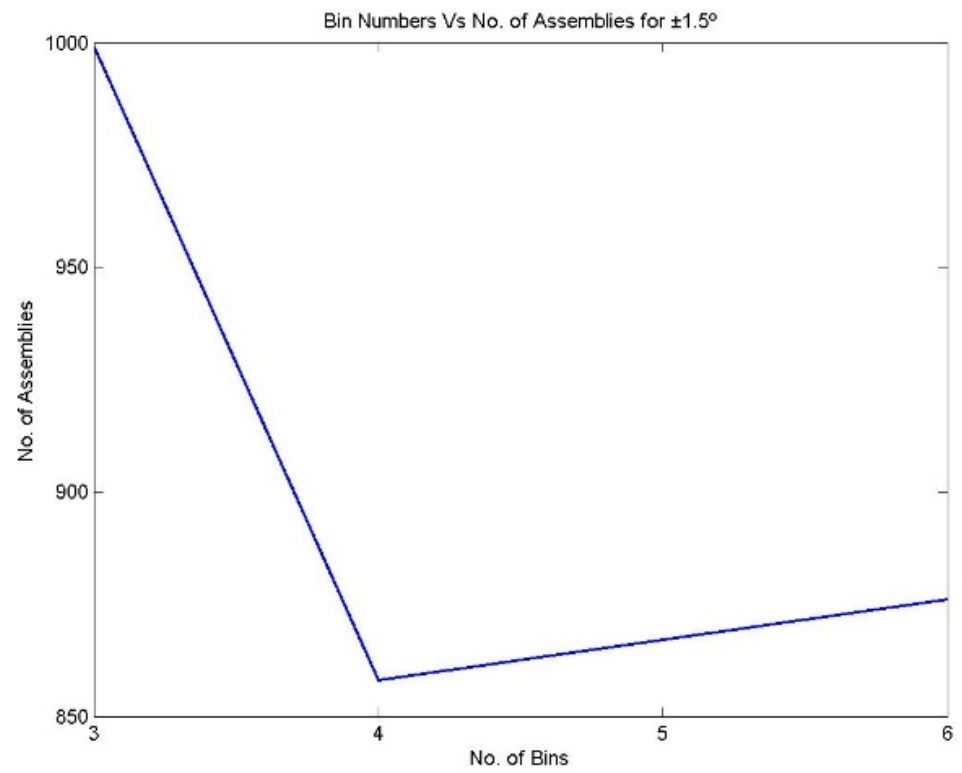


(a). *NoA* for  $\pm 0.5^\circ$

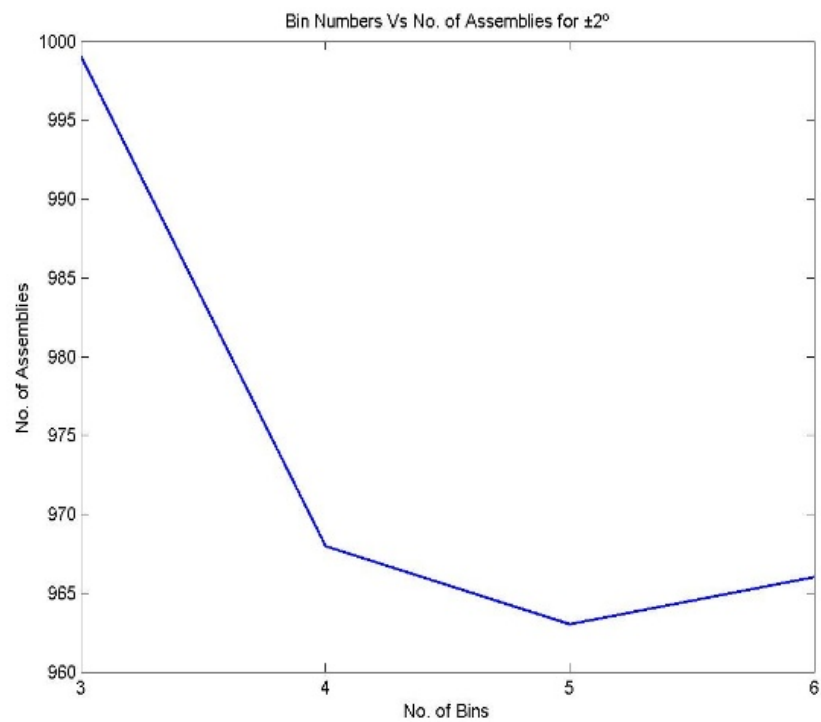


(b). *NoA* for  $\pm 1^\circ$

Figure 5. Cont.



(c). NoA for ±1.5°



(d). NoA for ±2°

**Figure 5.** NoA for various bin numbers and assembly specifications for the 3213 process combination (a) NoA for ±0.5°, (b) NoA for ±1°, (c). NoA for ±1.5° and (d) NoA for ±2°.

### 6. Results and Discussion

An attempt has been made to make assemblies by considering the number of bins/partitions, up to 9. Figure 6 reveals that in the equal area method for making non-linear assemblies, it is possible to make 996 assemblies out of 1000 components by partitioning them into 3 bins for the assembly specification of  $\pm 2^\circ$ . It is also understood that while increasing the partition number, there may be a 5.8% (938 assemblies) drop in producing the number of assemblies for the same assembly specification. In the meantime, the number of assemblies is reduced for the same 1000 components of X1, X2, X3, and X4 while reducing the assembly specification for the same partition number (equal to 3). The assemblies are reduced from 996 to 392 for the assembly specification of  $\pm 2^\circ$  to  $\pm 0.5^\circ$ .

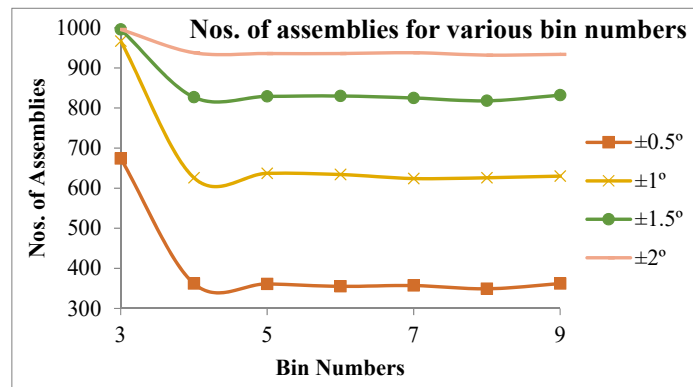


Figure 6. NoA vs. bin number for various assembly specifications for the 3213 process combination.

Figures 7–9 represent the number of assemblies produced for various assembly specifications while changing the partition number for the same set of 1000 components produced based on process combination 3213. A similarity can be observed from the above figures. Except in three bin partitions, all other partitions of components and matching components according to the best bin combinations obtained through the HSA produced almost a very close number of assemblies for various assembly specifications. Figure 10 indicates that maximum assemblies are produced for various assembly specifications and process combinations while partitioning the components into three bins. The assembly specification value after  $\pm 1^\circ$  in all the process combinations could produce an almost equal number of assemblies for bin number three. Table 15 represents the best bin combinations and their maximum number of assemblies for various process combinations.

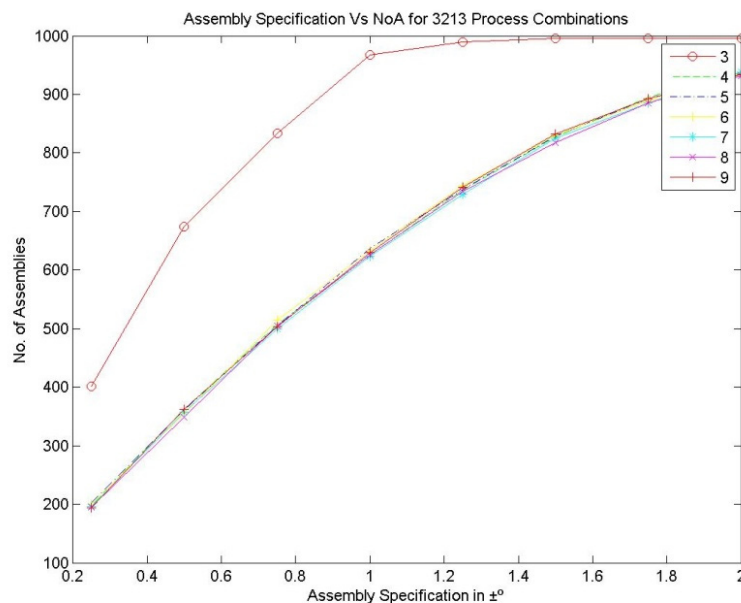


Figure 7. Assembly specification vs. NoA for various bin numbers—the 3213 process combination.



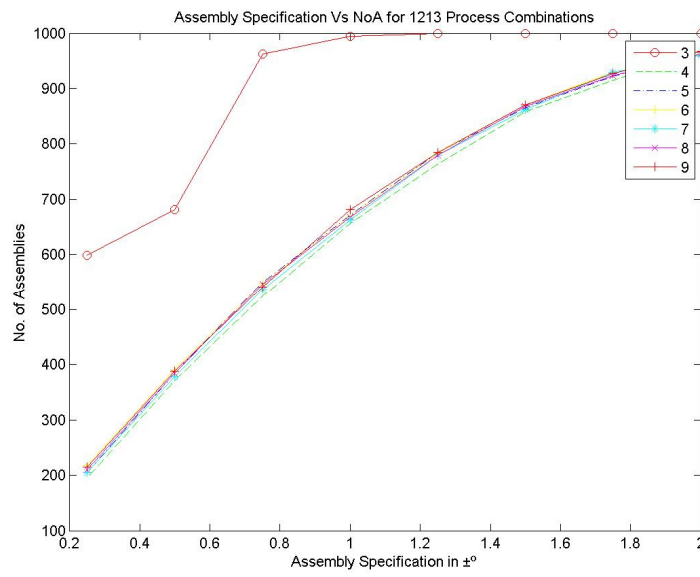


Figure 8. Assembly specification vs. *NoA* for various bin numbers—the 1213 process combination.

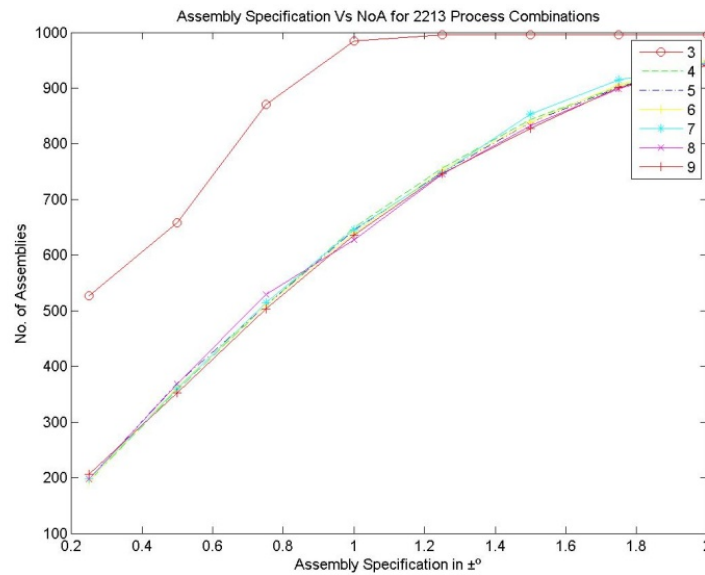


Figure 9. Assembly specification vs. *NoA* for various bin numbers—the 2213 process combination.

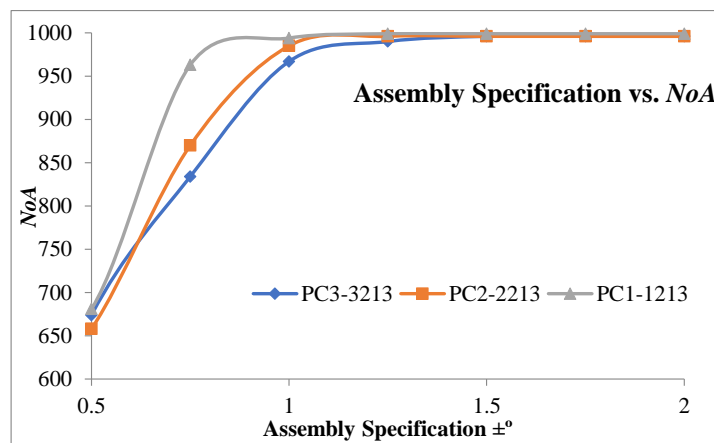


Figure 10. Maximum *NoA* produced for different assembly specifications and various process combinations.

**Table 15.** Best bin combinations for various assembly specifications and process combinations.

AS	PC3–3213					PC2–2213					PC1–1213																												
	X1	X2	X3	X4	NoA	X1	X2	X3	X4	NoA	X1	X2	X3	X4	NoA																								
±0.5°	1	2	3	2	3	1	2	3	1	1	3	2	674	3	2	1	1	2	3	2	3	1	1	2	3	658	2	1	3	3	2	1	3	1	2	3	1	2	681
±1°	1	2	3	3	2	1	2	3	1	2	3	1	967	1	3	2	3	1	2	2	1	3	2	1	3	985	2	1	3	2	1	3	3	2	1	3	1	2	994
±1.5°	1	2	3	3	2	1	2	3	1	2	3	1	996	2	1	3	2	3	1	2	3	1	2	3	1	996	2	1	3	3	2	1	3	2	1	3	2	1	999
±2°	1	3	2	2	1	3	2	3	1	1	3	2	996	2	1	3	1	2	3	3	2	1	2	1	3	996	1	3	2	1	3	2	1	3	2	1	3	2	999

AS—assembly specification.

### 7. Conclusions

This paper addresses a novel methodology by combining the univariate search method and the harmony search algorithm in selective assembly for making non-linear assemblies for various assembly specifications. The best processes for the different components of the assembly are selected from the known alternative processes using the univariate search method, and these components are grouped into 3 to 9 bins. Further, the best bin combinations for making assemblies to reduce manufacturing cost are obtained through the harmony search algorithm. In this work, the component’s dimensions are directly considered for making assemblies from the best bin combinations rather than considering tolerances, as in the existing method. The proposed method is demonstrated on a non-linear overrunning clutch assembly and has proved its efficiency by saving 24.9% of manufacturing cost compared with the existing method for the best process combination of 3213.

**Author Contributions:** Conceptualization and Methodology by L.N. and S.K.M.; Writing—Original Draft Preparation by L.N. and S.K.M.; Writing—Review and Editing by S.S., E.A.N., J.P.D. and H.M.A.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** The authors extend their appreciation to King Saud University for funding this work through Researchers Supporting Project number (RSP-2021/164), King Saud University, Riyadh, Saudi Arabia.

**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.

**Availability of Data and Material:** Not applicable.

### References

- Kern, D.C. Forecasting Manufacturing Variation Using Historical Process Capability Data: Applications for Random Assembly, Selective Assembly, and Serial Processing. Ph.D. Thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, Cambridge, MA, USA, 15 May 2003.
- Mease, D.; Nair, V.N.; Sudjianto, A. Selective assembly in manufacturing: Statistical issues and optimal binning strategies. *Technometrics* **2004**, *46*, 165–175. [CrossRef]
- Matsuura, S.; Shinozaki, N. Selective assembly to minimize clearance variation by shifting the process mean when two components have dissimilar variances. *Quality-Tokyo* **2008**, *38*, 73.
- Kwon, H.M.; Hong, S.H.; Lee, M.K. Effectiveness of the ordered selective assembly for two mating parts. In Proceedings of the 10th Asia Pacific Industrial Engineering & Management Systems, Kitakyushu, Japan, 6 September 2009; pp. 1402–1405. Available online: <https://apiems.org/> (accessed on 6 September 2021).
- Matsuura, S.; Shinozaki, N. Optimal process design in selective assembly when components with smaller variance are manufactured at three shifted means. *Int. J. Prod. Res.* **2011**, *49*, 869–882. [CrossRef]
- Yue, X.; Wu, Z.; Tianze, H.; Julong, Y. A heuristic algorithm to minimize clearance variation in selective assembly. *Rev. Tech. Fac. Ing. Univ. Zulia* **2014**, *37*, 55–65.
- Babu, J.R.; Asha, A. Tolerance modelling in selective assembly for minimizing linear assembly tolerance variation and assembly cost by using Taguchi and AIS algorithm. *Int. J. Adv. Manuf. Technol.* **2014**, *75*, 869–881. [CrossRef]
- Ju, F.; Li, J. A Bernoulli model of selective assembly systems. *IFAC Proc. Vol.* **2014**, *47*, 1692–1697. [CrossRef]
- Xu, H.-Y.; Kuo, S.-H.; Tsai, J.W.H.; Ying, J.F.; Lee, G.K.K. A selective assembly strategy to improve the components’ utilization rate with an application to hard disk drives. *Int. J. Adv. Manuf. Technol.* **2014**, *75*, 247–255. [CrossRef]

10. Lu, C.; Fei, J.-F. 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](#)]
11. Manickam, N.; De, T.N.; Laboratory, A.S. Optimum group size selection for launch vehicle sections linear assembly by selective assembly method. *Int. J. Eng. Res.* **2015**, *4*, 837–843. [[CrossRef](#)]
12. 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. [[CrossRef](#)]
13. Ju, F.; Li, J.; Deng, W. Selective assembly system with unreliable bernoulli machines and finite buffers. *IEEE Trans. Autom. Sci. Eng.* **2016**, *14*, 171–184. [[CrossRef](#)]
14. Malaichamy, T.; Sivasubramanian, R.; Sivakumar, M. Selective assem—A software approach. *Int. J. Adv. Eng. Technol.* **2016**, *7*, 729–735.
15. Malaichamy, T.; Sivasubramanian, R.; Kumar, S.M.; Sundaram, M.C. Simulated annealing algorithm for minimising the surplus parts in selective assembly-A software approach. *Asian J. Res. Soc. Sci. Humanit.* **2016**, *6*, 1567. [[CrossRef](#)]
16. Liu, S.; Liu, L. Determining the number of groups in selective assembly for remanufacturing engine. *Procedia Eng.* **2017**, *174*, 815–819. [[CrossRef](#)]
17. Chu, X.; Xu, H.; Wu, X.; Tao, J.; Shao, G. The method of selective assembly for the RV reducer based on genetic algorithm. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2017**, *232*, 921–929. [[CrossRef](#)]
18. Geem, Z.W.; Kim, J.H.; Loganathan, G. A new heuristic optimization algorithm: Harmony search. *Simulation* **2001**, *76*, 60–68. [[CrossRef](#)]
19. Geem, Z.; Kim, J.; Loganathan, G. Harmony search optimization: Application to pipe network design. *Int. J. Model. Simul.* **2002**, *22*, 125–133. [[CrossRef](#)]
20. Lee, K.S.; Geem, Z.W. A new structural optimization method based on the harmony search algorithm. *Comput. Struct.* **2004**, *82*, 781–798. [[CrossRef](#)]
21. Lee, K.S.; Geem, Z.W. A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice. *Comput. Methods Appl. Mech. Eng.* **2005**, *194*, 3902–3933. [[CrossRef](#)]
22. Wang, X.; Gao, X.Z.; Ovaska, S.J. Fusion of clonal selection algorithm and harmony search method in optimisation of fuzzy classification systems. *Int. J. Bio. Inspired Comput.* **2009**, *1*, 80. [[CrossRef](#)]
23. Ganesan, K.; Suresh, R.K.; Mohanram, P.V.; Vishnukumar, C.H. Optimum tolerance allocation and process selection using simulated annealing. In Proceedings of the 4th Int. Conference on Mechanical Engineering, Theni, Tamil Nadu, India, 26–28 December 2009; Volume VI, pp. 201–205.