Journal of Computers Vol. 33 No. 4, August 2022, pp. 81-94
doi: 10.53106/199115992022083304007

The Configuration Design of Electronic Products Based on Improved NSGA-III with Information Feedback Models

Yi-Hui Chen, Heng-Zhou Ye*, Feng-Yi Huang

Guangxi Key Laboratory of Embedded Technology and Intelligent System, Guilin University of Technology, P. R. China
2381928951@qq.com, 2002018@glut.edu.cn

Received 18 September 2021; Revised 2 February 2022; Accepted 2 March 2022

Abstract. The configuration of electronic products is an important means to meet the diverse and personalized needs of users and achieve mass customization, and one of its goals is to recommend an excellent bill of material to users according to users' individualized needs and preferences. Current research describes the configuration of electronic products as a single-objective optimization model, which suffers from the problems of single recommended configuration and difficulty in meeting the dynamic adjustment of user preferences. Therefore, we describe it as a multi-objective optimization model and propose an NSGA-III-FR algorithm to solve the model. In order to balance the convergence and diversity of the algorithm, NSGA-III-FR has made two improvements on the basis of NSGA-III with information feedback models: introducing adaptive parameters to balance NSGA-III-F1 and NSGA-III-R1; and using Angle Penalized Distance (APD) to improve the niche technology. The experimental results show that our modified method can achieve better performance compared with the other three algorithms.

Keywords: information feedback, NSGA-III, product configuration, Angle Penalized Distance

1 Introduction

With the popularity of the Internet and the rapid development of e-commerce, electronic products have become an essential tool for people. With the wide range of industries involved in electronic products and the rapid replacement of electronic products, the demand for electronic products has been diversified and personalized at an accelerated rate, and it is an urgent need for enterprises to design electronic products which meet the needs of users quickly and efficiently. Product configuration, as an important means to realize mass customization, can meet the diversification and individualization of user needs, and has received more and more attention from enterprises. The implementation of product configuration depends on flexible configuration models and efficient solution methods [1], and electronic products often involve more accessories, each of which has a variety of brands and models, increasing the difficulty of product configuration.

To enable active user participation in the product configuration process, some interactive evolutionary computation (IEC) is used for product customization [2-3], but they require user participation in mass individual adaptation and are prone to user fatigue. In the Internet era, the exponential growth of information on electronic products has led to information overload. As a result, users have to spend a lot of time looking for the information they need [4]. The parameters of electronic products are specialized and incomplete. It is difficult for users to have enough expertise to choose the electronic product they are satisfied with [5], so a fast and professional product configuration facilitates users to improve the efficiency of product selection. The current electronic product configuration design ignores how to specify users' personalized needs. Therefore, Han [6] constructed a single-objective optimization model for CTO order recommendation of electronic products, but it is not enough to meet the variability of user preferences, and often can only recommend a priority scheme. Nowadays, most product configuration design is a multi-objective optimization problem. For different types of products, product configurations [7] are described according to their characteristics, which include form, function, quality, etc. Wei [8] constructs a configuration model with performance, cost, and delivery time as objectives. Song [9] constructs a product configuration model with service performance, cost, and response time as objectives.

The NSGA-III proposed by Deb [10] has received much attention in solving multi-objective optimization models. The basic framework of NSGA-III is similar to NSGA-II [11]. The main difference between NSGA-III and NSGA-II is individual selection, i.e., replacing the congested distance with a selection based on reference points to obtain a more uniform Pareto front. This effect is confirmed by practical and empirical results in the literature [12]. NSGA-III or its improved algorithms have been used to solve multi-objective optimization problems

* Corresponding Author

successfully in many fields, such as power systems [13], satellite image detection [14], cargo distribution [15], desalination [16], esterification processes [17], and intrusion detection [18]. However, the performance of NSGA-III in solving configuration problem of electronic products is rarely verified.

Since NSGA-III has good performance on multi-objective models, this paper has applied it to solve multi-objective electronic product configuration models. The main contributions of this paper are as follows:

1) Based on the literature [6], the product configuration problem of electronic products is described as a multi-objective optimization model that considers three optimization objectives: functional target positioning closeness, cost and energy consumption.

2) An improved NSGA-III algorithm with information feedback (NSGA III-FR) is proposed to solve a multi-objective optimization model for electronic product configuration. NSGA III-FR balances NSGA III-F1 and NSGA III-R1 by adaptive parameter settings to better exploit their advantages and dilute their deficiencies; NSGA III-FR also introduces an angular penalty distance to improve the niche technology.

3) According to the dataset [6], the performance of NSGA-III, NSGA-III-F1, NSGA-III-R1 and NSGA-III-FR in solving multi-objective optimization models for electronic product configurations is compared from several perspectives through experiments.

In this paper, the configuration design of electronic products is described as a multi-objective optimization model and is solved using an improved NSGA-III algorithm with information feedback models. The rest of the paper is as follows. Section 2 describes the related work. Section 3 provides a multi-objective optimization model for the configuration of electronic products. Section 4 proposes NSGA-III-FR (improved NSGA-III with information feedback models) and applies APD (angular penalty distance) in NSGA-III-FR to improve the niche technology. Section 5 shows the experimental results. Finally, Section 6 provides our conclusions.

2 Related Work

In the electronics industry, mass customization is an effective way to improve operations, but it requires extensive development of product architecture to achieve and implement mass customization [19]. Product configuration is one of the key technologies of mass customization [8, 20]. Product configuration is the process of designing individual customer requirements based on a variable product model based on user needs [21]. Wang *et al*. reviewed the research status of product configuration design from the aspects of customer demand analysis, configuration modeling, and configuration solution, and pointed out further research directions [22].

Du [23] developed a multi-objective optimization model considering cost and user satisfaction to optimize product configurations using a hybrid ant colony algorithm. Long [24] developed a multiclass vector machine model to configure a specific product service system (PSS) to meet customer needs based on product service system configuration. Lei [25] constructed a multi-objective optimization model with product configuration cost, carbon emission and product reliability as optimization objectives, where the core is product reuse. In [26], the correlation between the attribute parameters of the configuration unit of the product instantiation order was focused. Dong [27] proposed an ontology-based modeling approach for service product configuration. Zhao [28] integrated Kano model-based customer requirements into product configuration design and transformed the model results into a multi-objective function in a hybrid nonlinear programming model. Wang [29] focused on demand-driven product configuration design, but used a hierarchical attention network approach. Although the above research focused on product configuration, it's different from our focus. Our focus is consistent with the literature [6] which uses a genetic algorithm to find the optimal solution and recommends it to users by building a single-objective optimization model. However, the single-objective model is not suitable for recommending multiple excellent solutions, and it is not convenient for users to adjust the preference weight at any time. Therefore, we try to describe the product configuration problem as a multi-objective optimization model.

In recent years, many scholars have solved multi-objective models for product configurations by multi-objective genetic algorithms. Zhan [30] constructed a product configuration model with the objectives of performance, cost and delivery time, and designed an improved non-dominated ranking genetic algorithm NSGA-II to solve the multi-objective optimization model and recommend configuration scheme preferences based on customers. Cheng [31] used the non-dominated ranking genetic algorithm NSGA-II to solve a multi-objective optimization model of collaborative manufacturing chain based on time series constraints to achieve sharing and optimal allocation of manufacturing resources. Altiparmak [32] designed a supply chain network model with the objectives of cost, service level and resource utilization, and compared the model solution results with a simulated annealing algorithm based on an improved genetic algorithm. Kuriakose [33] used a multi-objective optimization approach based on non-dominated ranking genetic algorithm (NSGA-II) to optimize the wire-cutting process. Several studies have proposed some improved NSGA-III algorithms, and Sang [34] proposed an improved genetic algorithm NSGA-III-APEV to solve a multi-objective shop floor scheduling model. Khettabi [35] designed a multi-objective reconfigurable manufacturing system model, experimentally compared four genetic algorithms, and analyzed the effect of genetic operators on the convergence of NSGA-III. Liu [36] proposed the NSGA-III-GKM algorithm using genetic K-mean clustering algorithm for clustering reference points, which has better diversity and convergence. Xue [37] used a maximum ranking strategy to improve the NSGA-III algorithm. Among others, this improved NSGA-III balances convergence and diversity by adaptive parameter settings to achieve better performance. However, these improved methods do not take into account the potentially useful information before generating new parent populations.

Information feedback models have been increasingly applied to intelligent algorithms. In recent years, many researchers have done extensive studies in this area, such as teaching learning based optimization (TLBO) [38], particle swarm optimization (PSO) [39], artificial bee colony (ABC) [40], and dandelion algorithm (DA) [41]. These algorithms which apply the information feedback models have been proved experimentally to have better performance. The core of the information feedback models is to use the previous useful information to guide the subsequent behaviors, making up for the shortcomings of wasting a lot of useful information. Wang [42] introduced information feedback to heuristic algorithms for the first time. Zhang [43] added information feedback models to the multi-objective algorithm for the first time, which uses information from previous iterations to update the current information through a weighted fitness function. Gu [44] applied information feedback models to NSGA-III and compared six information feedback models showing that NSGA-III-F1 and NSGA-III-R1 have better performance.

For the multi-objective optimization problem of electronic product configuration design, we propose an information feedback model NSGA-III-FR. NSGA-III-FR improves the convergence and diversity of algorithm by using previous information to update the current information. The improved algorithm can provide more accurate and uniformly spaced solutions for the configuration design of electronic products. The specific details will be explained later in the paper.

3 The Multi-objective Optimization Model of Electronic Product Configuration Design

The single-objective optimization model was constructed by [6]. We describe the electronic product configuration design as a multi-objective optimization model, that is, maximizing function positioning closeness (μ), minimizing cost (sc), and minimizing power consumption (sp).

μ represents the number of indicators for product function positioning closeness, which is calculated by

$$\mu=\sum_{i=1}^{u} w_i \cdot \mu_i \quad . \tag{1}$$

where w_i is the weight vector of the i-th indicator. μ_i is the closeness of the i-th indicator, which adopts the measurement of [6].

sc is the normalized cost of the electronic products, which is given by

$$sc = \frac{cost - cost_{min}}{cost_{max} - cost_{min}} \quad . \tag{2}$$

where $cost_{min}$ and $cost_{max}$ are the minimum and maximum values of all possible electronic product costs, respectively. The cost of an electronic product is given by

$$cost = \sum_{i=1}^{n}\sum_{j=1}^{m_i} x_{ij} \cdot price_{ij} + price_0 \quad . \tag{3}$$

where n is the total number of accessories, m_i is the total number of the i-th accessory, x_{ij} (i =1, 2,..., n ; j =1, 2,…, n) is a 0-1 variable, $x_{ij}=1$ represents that the j-th model is selected for the i-th accessory, $price_{ij}$ is the cost of the j-th model selected of the i-th accessory, and $price_0$ is the fixed cost.

sp is the normalized power consumption of electronic products, which is given by

$$sp = \frac{power - power_{min}}{power_{max} - power_{min}} . \tag{4}$$

where $power_{min}$ and $power_{max}$ are the minimum and maximum power consumption of all possible electronic products, respectively. The power consumption of an electronic product is given by

$$power = \sum_{i=1}^{n}\sum_{j=1}^{m_i} x_{ij} \cdot power_{ij} + power_0 . \tag{5}$$

where $power_{ij}$ is the cost of the j-th model selected of the i-th accessory, and $power_0$ is the adjusted power consumption.

4 Improving NSGA-III with Information Feedback Models

The configuration design model of electronic products in this paper is essentially a multi-objective combinatorial optimization problem. The NSGA-III algorithm has achieved good performance in solving such problems. With the introduction of information feedback models, NSGA-III can balance convergence and diversity better. Based on the multi-objective optimization model of electronic product configuration design, we establish a new solution model NSGA-III-FR by improving the information feedback strategy and the niche technology.

4.1 NSGA- III-FR

The NSGA-III-FR algorithm used to solve the configuration model for electronic products is an improvement on the information feedback NSGA-III algorithm, which is consistent with the NSGA-III framework (Fig. 1) and involves the main steps as following:

Step 1: Initialize the population and create reference points. The chromosomes of the population are coded with n real numbers (n is the number of accessories of electronic products). The i-th real number, an integer value from $[0, m_i]$ (m_i is the number of models available of i-th accessory), represents the code of the model selected of i-th accessory. The initial population consists of a set of random sample values which obey a standard normal distribution, and its size is N. Considering that there are three optimization objectives, we divide each objective into four parts, and set a total of 15 reference points, that is, Z={1, 2, …, 15}.

Step 2: Generate a new population Q_t. The generation of a new population R_t requires two steps: first, P_t generates chromosome U_t by the crossover and mutation operation. Crossover and mutation operations are conventional, that is, to simulate binomial crossover [45] and polynomial mutation [46]. Second, Q_t is generated by P_t and U_t using information feedback strategy. R_t is generated by the offspring population Q_t and the parent population P_t, that is, $\left(Q_t \bigcup P_t = R_t\right)$. The information feedback strategy is discussed in detail in Section 4.2.

Step 3: Screen out a new population P_{t+1}. Firstly, perform on-dominated sorting on R_t and divide it into different non-dominated solution sets $\left(F_1, F_2, ..., F_w\right)$. The next step is to move one non-dominated solution set to S_t each time, start with F_1, until $\left|S_t\right| \geq N$ is satisfied for the first time. ($\left|S_t\right|$ is the number of solutions in S_t).

Choose $K = N - |P_{t+1}|$ individuals from F_1 to add to P_{t+1} according to the selection. The specific selection is detailed in Section 4.3.

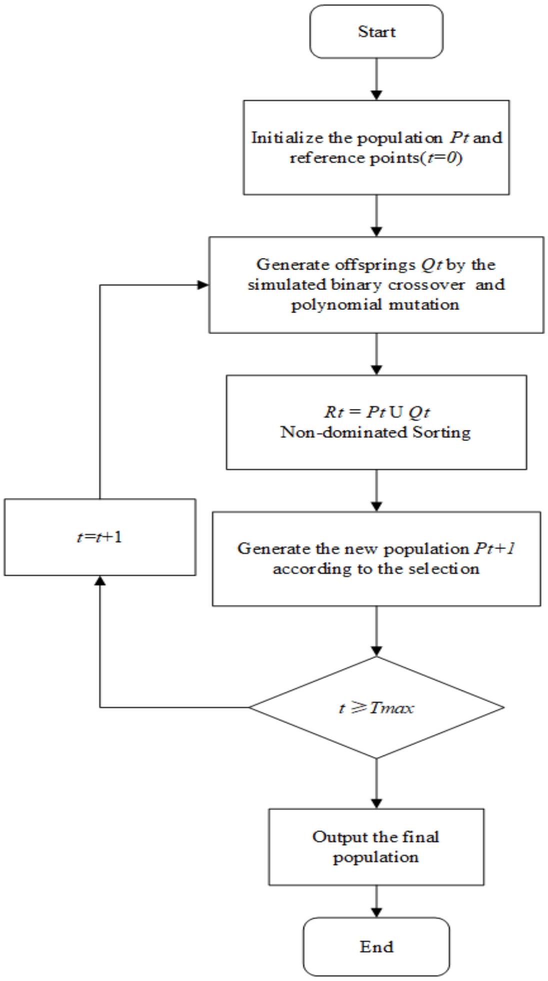

Fig. 1. The flowchart of NSGA-III

4.2 Information Feedback Models and Its Improvement

An information feedback model refers to an optimization strategy that extracts information from the previous population and feeds it back to the offspring through weight coefficients, so as to reuse the previous useful information. Gu [44] showed that the model works best when only considering the information of the *t*-th population to update the (*t*+1)-th population in the iterative process. At this time, the individual of the (*t*+1)-th population can be expressed by

$$s_i^{t+1} = \frac{f_i^t}{f_i^{t+1} + f_i^t} u_i^{t+1} + \frac{f_i^{t+1}}{f_i^{t+1} + f_i^t} x_k^t \ . \tag{6}$$

where u_i^t represents the *i*-th individual of the *t*-th population generated by the original NSGA-III. s_i^t and f_i^t represent the *i*-th individual of the *t*-th population and its fitness, respectively. The final fitness of an individual *s* is the weight average of the fitness of the three optimization objective. $f_1(s) = 1 - \mu$, $f_2(s) = sc$ and $f_3(x) = sp$ are the fitness of the individual's three optimization objectives, respectively, that is, function target closeness μ, cost

sc and price sp. The final fitness of the individual s is expressed by

$$f(s)=\sum_{i=1}^{3}\varphi_i\cdot f_i(s) \quad . \tag{7}$$

where φ_1, φ_2, φ_3 are weight coefficients, reflecting the user's preference for each optimization objective, which must satisfy $\varphi_1+\varphi_2+\varphi_3=1$. According to the value of *k*, it can be divided into the model NSGA-III-F1 or the model NSGA-III-R1. When *k* =*i*, it is NSGA-III-F1; when *k* is a random integer from 1 to *N*, it is NSGA-III-R1. Generally speaking, NSGA-III-R1 has fast convergence speed, but it is highly likely to fall into partial convergence. NSGA-III-F1 has better diversity, but the convergence speed is slow a little. In order to balance the convergence and diversity better, the adaptive parameter $\theta\in[0,1]$ is proposed. When a new individual is generated each time, the model NSGA-III-F1 is selected with probability θ, but the model NSGA-III-R1 is selected with probability 1-θ. The value of θ is expressed by

$$\theta=\frac{\alpha}{1+e^{\beta(1-\frac{t}{T})}} \quad . \tag{8}$$

where *t* and *T* are the current number of iteration and the total number of iterations, respectively. α and β are adjustable parameters. α is chosen from [0,1], and $\beta>0$. At the beginning of the iteration, *t* is relatively small and θ is also small. Therefore, the model tends to be NSGA-III-F1 to prevent partial convergence and improve the diversity of NSGA-III. At the end of the iteration, *t* is relatively large and θ is also large. Thus, the model tends to be NSGA-III-R1 so as to reduce the damage of convergence.

4.3 Improved Niche Technology

NSGA-III uses niche technology to select *K* individuals from F_1 to add to P_{t+1}. The original strategy is to preferentially select the individuals with the smallest vertical distance from the associated reference lines. However, this strategy may cause some bad consequences. At the beginning of the iteration, the population has insufficient convergence. At the later stage of the iteration, the population is too concentrated in a small partial area near the Pareto Front, resulting in poor diversity. For this problem, Penalty-based Boundary Intersection (PBI) distance [47], Angle Penalized Distance (APD) [48] and other methods have been proposed to replace vertical distance. In comparison, APD considers the impact of the iterative process; thus, it can balance convergence and diversity of the population dynamically.

Combined with the configuration design model of electronic products, for individual *s* in F_l, APD(*s*) can be expressed by the following steps:

Step 1: Calculate the degree of convergence of the individual. For the individual *s* from S_t, the distance $d(s)$ from the objective function to the ideal points is used to measure its convergence. Therefore, $d(s)$ is given by

$$d(s)=\sqrt{\sum_{i}^{3}(f_i(s)-Z_i^{\min})^2} \quad . \tag{9}$$

where $Z_i^{\min}$ represents the minimum value of the *i*-th optimization objective in the population S_t.

Step 2: Calculate the distribution of individuals. The distribution of the individual s is measured by the angle $\gamma(s)$ between the individual s and the associated reference lines. For comparison, we normalize it as $\gamma'(s)$ using Eq. (10). Therefore $\gamma'(s)$ is given by

$$\gamma'(s) = \gamma(s) / \gamma^{\min}(s) \quad . \tag{10}$$

where $\gamma^{\min}(s)$ is the minimum value of the angle between the individual s and all reference lines.

Step 3: Calculate the individual's APD, the APD of the individual s is given by

$$\mathrm{APD}(s) = (1 + 3 \cdot (\frac{t}{T})^{\varepsilon} \cdot \gamma'(s)) \cdot d(s) \quad . \tag{11}$$

where ε is an adjustable parameter, which can controls the change of APD(s) in the iterative process. Eq. (11) shows at the beginning of the iteration, APD is influenced mainly by the degree of convergence of the individual, that is, $d(s)$. As the iteration progresses, the impact of APD on the distribution of individuals is increasing. Therefore, to select individuals by APD can balance convergence and diversity of the algorithm dynamically.

ρ_j represents the niche count of the j-th reference point from the reference point set Z. We need to calculate the niche count ρ_j value. Next, we give the detailed algorithm about niche technology using APD as follows.

```
program niche (K, Pt+1, F1, Z)
  Calculate the niche count ρi of reference point i from Z;
  begin
   k := 1;
   repeat
     k := k + 1;
     Find the reference point j with the lowest niche count ρj value and
     choose one randomly if there are more than one members;
     Find individuals in set Fl associated with the reference point j and
     put them into the set Ij;
       if (ρj ==0) {
          if(Ij≠∅) {
           Find individual s with the lowest APD(s) value from Ij;
           Add s to Pt+1, and remove s from F1;
           ρj := ρj + 1;
         }
         else
           Remove j from set Z;
       }
       else {
         Find the individual s from Ij to minimize f(s);
         Add s to Pt+1, and remove s from F1;
         ρj := ρj + 1;
       }
   until (k>K)
end.
```

4.4 Discussion of the Limitations of the Algorithm

To balance the convergence and diversity of the algorithm, we introduce three adjustable parameters α, β and ε. The values of these three parameters will affect the performance of the algorithm, and their reasonable values are related to the specific problems. In our subsequent experiments, the values of these three parameters are given based on experience and experimental comparison results. Next, we will pay more attention to the problem of taking the values of these three parameters.

5 Experiment Analysis

In this Section, we do some experiments by using the NSGA-III-FR algorithm, the original NSGA-III [10], NSGA-III-F1 [44], and NSGA-III-R1 [44] to solve multi-objective optimization configuration design model of electronic products. Then, these experimental results will be compared and analyzed their performance.

5.1 Experimental Setting

We use the dataset of [6]. In this dataset, the desktop PC is considered as the research object and there are 348 records. Each record involves 10 accessories such as CPUs, GPUs, hard disks, keyboards, headsets, etc. In the experiment, the relevant parameters are set as follows: the population size N is 100, the crossover probability P_c and the mutation probability P_m are both 1, the crossover parameter η_c and mutation parameter η_m are both 20, and the adjustable parameter α, β and ε are 0.5, 3, 2, respectively.

5.2 Performance Metrics

The fitness value is used to measure the user satisfaction of the electronic product configuration results [6, 49], and we measure the fitness value of product configuration scenarios by Eq. (7). Since Eq. (7) uses negative factors, the lower the fitness value, the higher the user satisfaction.

In order to evaluate the convergence and diversity of the algorithm, Spacing [11], Generation Distance (GD) [50], Inversion Generation Distance (IGD) [51] and Hypervolume (HV) [52] are denoted as classical performance metrics. Assuming that P is a certain solution set, N is the size of the solution set P, P^* is the reference point set of the population P^*, d_i is the minimum Euclidean distance from individual i in P to P^*, and d^- is the mean value of all d_i. Moreover, Spacing, GD, IGD, and HV are explained as follows:

Spacing represents the standard deviation of the minimum distance from each solution to other solutions. Spacing is a classic performance metric that measures uniformity of the solution set. The smaller the value, the more uniform the solution set. The Spacing value of the solution set P is given by

$$\text{Spacing}(P)=\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(d^- - d_i)^2}. \tag{12}$$

GD represents the average minimum distance from each individual in the solution set to the reference points. The smaller the GD value, the better the convergence. The GD value is given by

$$GD(P,P^*)=\frac{\sum_{i=1}^{N} d_i^2}{N}. \tag{13}$$

IGD represents the average value of the distance from each reference point to the nearest solution. The smaller IGD value, the better overall performance of the algorithm. The IGD value can be given by

$$IGD(P^*,P)=\frac{\sum_{i=1}^{N} d_i^2}{N^*}. \tag{14}$$

where N^* is the number of reference points.

HV represents the volume of the area in the target space enclosed by the non-dominated solution set and the reference points obtained by the algorithm. The larger the value, the better the overall performance of the algorithm. The HV value can be given by

$$HV(P^*, P) = \delta(\bigcup_{i=1}^{N} v_i) \quad . \tag{15}$$

where δ represents the Lebesgue metric used to measure the volume, and v_i represents the hypervolume formed by P^* and the *i*-th individual in P.

5.3 Comparisons of NSGA-III-FR with Three Other Algorithms

Fig. 2 compares the fitness values of the four algorithms (average of 10 iteration results). From the figure, we can see that NSGA-III-FR has the lowest fitness value in ten different iterations. There is little difference between NSGA-III-F1 and NSGA-III-R1, which is about 9% higher than NSGA-III-FR and about 5% lower than NSGA-III. In terms of the effect of the number of iterations, NSGA-III shows a trend of decreasing fitness as the number of iterations increases. The other three algorithms have some fluctuations, but there is no significant rising and falling trend. This indicates that NSGA-III-FR can obtain understandable results after 100 iterations.

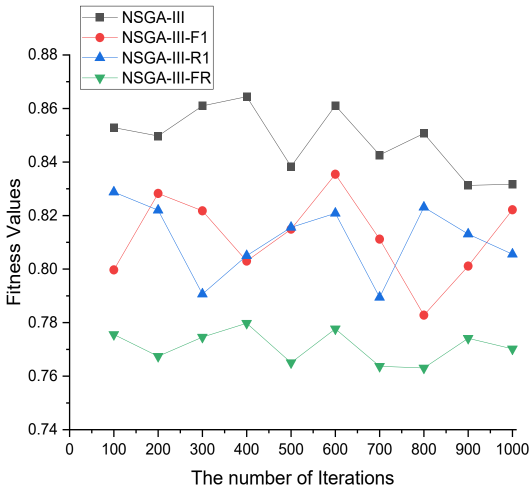

Fig. 2. Comparison chart of fitness values of four algorithms

Fig. 3 further compares the standard deviation of the fitness value of the four algorithms. As can be seen from the figure, the standard deviation of NSGA-III-FR is better than the other three algorithms, which indicates that NSGA-III-FR performs better than the other three algorithms.

Fig. 4 shows the comparison results of the four algorithms in terms of Spacing values. It can be seen that NSGA-III has a higher Spacing value than the other three algorithms in ten different iterations. Therefore, it means that the algorithms with information feedback models have a more uniform solution set than NSGA-III. The information feedback models use the previous useful information to update the current information, making up for the shortcomings of wasting useful information, which makes the solution set distributed more evenly.

Table 1 shows the comparison results of the four algorithms in terms of GD values. The average and standard deviation of GD values are presented by AVG and SD, respectively. The bold font indicates that the algorithm has the best performance in this iteration. It can be observed from the table that NSGA-III-FR achieves the best results for different number of iterations, whether it is the average or standard deviation of GD. From the effect of the number of iterations, NSGA-III-F1 shows a slight increase trend with the increase of the number of iterations, while the other three algorithms only show slight fluctuations. It shows that NSGA-III-F1 is easier to fall into local convergence than the other three algorithms.

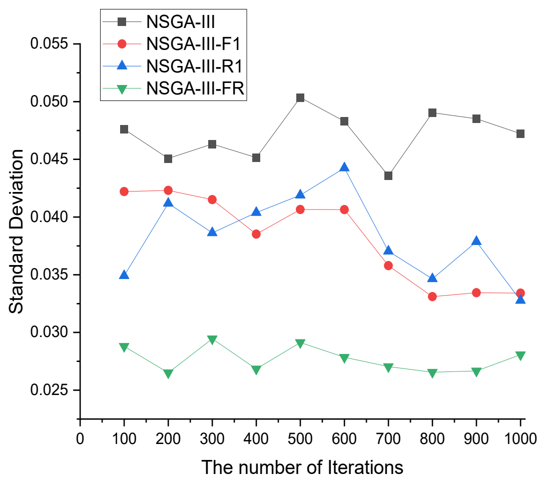

Fig. 3. Comparison chart of standard deviation of fitness values of four algorithms

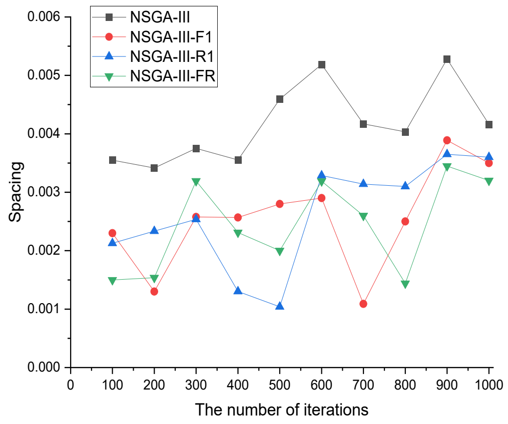

Fig. 4. Comparison chart of Spacing values of four algorithms

Table 2 shows the comparison results of the four algorithms in terms of IGD values. The average and standard deviation of IGD values are presented by AVG and SD, respectively. The bold font indicates that the algorithm has the best performance in this iteration. As can be seen from Table 2, NSGA-III-FR performs best in most iterations. When the number of iterations is 300 and 900, respectively, NSGA-III-R1 has the best performance, followed by NSGA-III-FR. NSGA-III-R1 outperforms NSGA-III-F1 in six iterations (when the number of iterations is 100, 300, 400, 600, 900 and 1000, respectively), and NSGA-III-F1 and NSGA-III-R1 have the same performance when the number of iterations is 800. IGD is a classical metric for evaluating the convergence and diversity of the algorithm. From the experimental results, it can be seen that NSGA-III-FR has the best performance, followed by NSGA-III-R1. In terms of the effect of the number of iterations, we can see that the fluctuations of the AVG values of the four algorithms, NSGA-III-FR, NSGA-III-R1, NSGA-III-F1, and NSGA-III, range from [0.289, 0.296], [0.286, 0.299], [0.294, 0.306], and [0.301, 0.311], respectively. It can be seen that the IGD value of NSGA-III-FR is least affected by the number of iterations, followed by NSGA-III. This indicates that the performance of NSGA-III-R1 and NSGA-III-F1 is unstable, while NSGA-III-FR improves this.

Table 1. Comparison results on GD values of NSGA-III-FR with three other algorithms

The number of iterations		NSGA-III	NSGA-III-F1	NSGA-III-R1	NSGA-III-FR
100	AVG	2.31e-2	1.88e-2	1.88e-2	**1.79e-2**
	SD	2.37e-3	3.00e-3	2.53e-4	**1.17e-3**
200	AVG	2.09e-2	2.10e-2	2.05e-2	**1.74e-2**
	SD	2.08e-3	2.61e-3	2.19e-3	**1.79e-3**
300	AVG	2.22e-2	1.99e-2	1.86e-2	**1.77e-2**
	SD	2.53e-3	2.18e-3	1.09e-3	**8.77e-4**
400	AVG	2.06e-2	2.20e-2	2.06e-2	**1.85e-2**
	SD	2.49e-3	4.54e-3	3.78e-3	**5.22e-4**
500	AVG	2.24e-2	2.08e-2	2.17e-2	**1.73e-2**
	SD	2.62e-3	2.78e-3	2.58e-3	**1.91e-3**
600	AVG	2.34e-2	2.05e-2	1.86e-2	**1.81e-2**
	SD	6.93e-5	2.14e-3	4.19e-3	**8.38e-4**
700	AVG	2.18e-2	1.95e-2	1.87e-2	**1.74e-2**
	SD	2.27e-3	8.53e-4	3.88e-3	**1.87e-3**
800	AVG	2.22e-2	2.01e-2	1.97e-2	**1.87e-2**
	SD	2.14e-3	2.87e-4	4.52e-3	**3.76e-4**
900	AVG	2.43e-2	2.16e-2	2.04e-2	**1.77e-2**
	SD	2.49e-4	2.80e-3	2.93e-3	**1.51e-3**
1000	AVG	2.33e-2	2.33e-2	1.98e-2	**1.78e-2**
	SD	1.92e-3	4.64e-4	2.51e-3	**2.25e-3**

Table 2. Comparison results on IGD values of NSGA-III-FR with the other three algorithms

The number of iterations		NSGA-III	NSGA-III-F1	NSGA-III-R1	NSGA-III-FR
100	AVG	3.01e-1	2.96e-1	2.94e-1	**2.90e-1**
	SD	1.23e-2	5.76e-3	5.04e-3	**8.28e-3**
200	AVG	3.11e-2	2.95e-1	2.99e-1	**2.89e-1**
	SD	3.69e-3	7.14e-3	6.68e-3	**4.35e-3**
300	AVG	3.06e-1	2.94e-1	**2.86e-1**	2.89e-1
	SD	5.06e-3	8.87e-3	**3.42e-3**	9.89e-3
400	AVG	3.05e-1	3.06e-1	2.98e-1	**2.92e-1**
	SD	7.27e-3	7.80e-3	5.70e-3	**3.06e-3**
500	AVG	3.01e-1	2.94e-1	3.02e-1	**2.90e-1**
	SD	1.02e-2	1.28e-2	1.35e-2	**6.44e-3**
600	AVG	3.06e-1	2.98e-1	2.93e-1	**2.89e-1**
	SD	3.03e-3	5.87e-3	8.58e-3	**2.94e-3**
700	AVG	3.09e-1	2.96e-1	2.99e-1	**2.91e-1**
	SD	5.48e-3	1.06e-2	9.97e-3	**6.86e-3**
800	AVG	3.08e-1	2.95e-1	2.95e-1	**2.92e-1**
	SD	7.65e-3	4.31e-3	8.70e-3	**2.85e-3**
900	AVG	3.11e-1	3.04e-1	**2.94e-1**	2.96e-1
	SD	6.08e-4	1.21e-2	**1.09e-2**	9.24e-2
1000	AVG	3.06e-1	3.01e-1	2.95e-1	**2.93e-1**
	SD	3.97e-3	4.59e-3	8.46e-3	**5.38e-3**

Table 3 shows the comparison results of the four algorithms in terms of HV values. The average and standard deviation of HV values are presented by AVG and SD, respectively. The bold font indicates that the algorithm has the best performance in this iteration. From the table, we can see that NSGA-III-FR performs best in eight iterations among ten different iterations (from 100 to 1000). When the number of iterations is 300, NSGA-III-R1 performs the best, followed by NSGA-III-FR. When the number of iterations is 800, NSGA-III-F1 performs better than NSGA-III-FR. Overall, NSGA-III-FR performs the best, NSGA-III-F1 and NSGA-III-R1 have similar performance, but NSGA-III performs the worst. In terms of the effect of the number of iterations, the fluctuations of the HV values of these four algorithms are highly similar to those of IGD, indicating that NSGA-III-FR is more stable than NSGA-III-F1 and NSGA-III-R1.

Table 3. Comparison results on HV values of NSGA-III-FR with the other three algorithms

The number of iterations		NSGA-III	NSGA-III-F1	NSGA-III-R1	NSGA-III-FR
100	AVG	1.44e-1	1.55e-1	1.59e-1	**1.69e-1**
	SD	2.31e-2	7.65e-3	1.10e-2	**1.97e-2**
200	AVG	1.28e-1	1.55e-1	1.48e-1	**1.74e-1**
	SD	6.83e-3	1.75e-2	1.11e-2	**8.34e-3**
300	AVG	1.34e-1	1.59e-1	**1.75e-1**	1.72e-1
	SD	9.85e-3	1.95e-2	**8.78e-3**	1.99e-2
400	AVG	1.39e-1	1.37e-1	1.50e-1	**1.65e-1**
	SD	1.32e-2	1.74e-2	1.00e-2	**8.32e-3**
500	AVG	1.43e-1	1.58e-1	1.44e-1	**1.74e-1**
	SD	2.01e-2	2.57e-2	2.63e-2	**1.11e-2**
600	AVG	1.32e-1	1.49e-1	1.65e-1	**1.71e-1**
	SD	4.67e-3	1.17e-2	2.15e-2	**9.75e-3**
700	AVG	1.31e-1	1.56e-1	1.54e-1	**1.73e-1**
	SD	1.22e-2	1.77e-2	2.20e-2	**5.80e-3**
800	AVG	1.31e-1	**1.67e-1**	1.60e-1	1.64e-1
	SD	1.11e-2	**8.64e-3**	2.46e-2	4.65e-3
900	AVG	1.23e-1	1.41e-1	1.59e-1	**1.61e-1**
	SD	7.42e-4	2.27e-2	2.45e-2	**1.63e-2**
1000	AVG	1.33e-1	1.48e-1	1.55e-1	**1.65e-1**
	SD	3.94e-3	5.41e-3	1.65e-2	**1.37e-2**

6 Conclusion

The configuration design is the process of finding the optimal configuration list according to user needs and preferences. For electronic products, this often means selecting the appropriate instance for each core component. In this paper, we describe this process as a multi-objective optimization model that considers three optimization objectives of functional target positioning closeness, cost, and power consumption. To solve this model, we propose an improved algorithm NSGA-III-FR based on the existing information feedback NSGA-III algorithm by introducing adaptive parameters and APD. Extensive experiments make a conclusion that our method can achieve better user satisfaction and have better convergence and stability.

Our method also has some deficiencies: (1) the correlation between product accessories and inventory information are not considered when building the product configuration design model; (2) NSGA-III-FR we proposed involves three adjustable parameters, and the way they are regulated still needs to be discussed. These will be the goals of our next work.

7 Acknowledgement

This research was supported by the Guangxi Science and Technology Major Project [grant number AA19046004], and the Guangxi Key Laboratory Director Fund of Embedded Technology and Intelligent Information Processing [grant number 2020-1-7].

References

[1] Q. Yang, J.-G. Li, C. Yang, D.-Z. Wei, A product configuration method based on DCSP and CBR, Modern Manufacturing Engineer 15(8)(1975) 91-95.

[2] J.-D. Zhu, S.-H. Su, C. Chen, G.-Y. Liu, Improved Interactive Genetic Algorithm for Product Configuration Design, China Mechanical Engineering 29(20)(2018) 2474-2478.

[3] R.-L. Dou, C. Zong, G.-F. Nan, Multi-stage Interactive Genetic Algorithm for Collaborative Product Customization, Knowledge-based Systems 92(2015) 43-54.

[4] Y.-K. Cao, Y.-F. Li, An intelligent fuzzy-based recommendation system for consumer electronic products, Expert Systems with Applications 33(2007) 230-240.

[5] C.-Q. Tan, L.-M. Hu, Recommendation Method for Consumer Electronic Products under Fuzzy Environment, Operations Research and Management Science 27(4)(2018) 126-137.

[6] H.-F. Han, H.-Z. Ye, L. Zhang, CTO order recommentdation of electronic products based on improved genetic algorithm, Journal of Guilin University of Technology 40(2)(2020) 177-183.
[7] H.-E. Tseng, C.-C. Chang, S.-H. Chang, Applying case-based reasoning for product configuration in mass customization environments, Expert Systems with Applications 29(2005) 913-925.
[8] W. Wei, W.-H. Fan, Z.-K. Li, Multi-objective optimization and evaluation method of modular product configuration design scheme, International Journal of Advanced Manufacturing Technology 75(2014) 1527-1536.
[9] W.-Y. Song, F.T.S. Chan, Multi-objective configuration optimization for product-extension service, Journal of Manufacturing Systems 37(2015) 113-125.
[10]K. Deb, H. Jain, An Evolutionary Many-Objective Optimization Algorithm Using Reference-point Based Non-dominated Sorting Approach, Part I Solving Problems with Box Constraints, IEEE Transactions on Evolutionary Computation 18(4)(2014) 577-601.
[11]K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6(2)(2002) 182-197.
[12]H. Li, K. Deb, Q.-F. Zhang, P.N. Suganthan, L. Chen, Comparison between MOEAD and NSGA-III on a set of novel many and multi-objective benchmark problems with challenging difficulties, Swarm and Evolutionary Computation 46(2019) 104-117.
[13]X.-H. Yuan, H. Tian, Y.-B. Yuan, Y.-H. Huang, R.M. Ikram, An extended NSGA-III for solution multi-objective hydro-thermal-wind scheduling considering wind power cost, Energy Conversion and Management 96(2015) 568-578.
[14]R. Gupta, S.J. Nanda, Many-objective B/NSGA-III for band selection in cloud contaminated hyper-spectral images, in: Proc. 2019 International Conference on Information Technology, 2019.
[15]K. Lavangnananda, P. Wangsom, Multi-Objective Shipment Allocation using Extreme Nondominated Sorting Genetic Algorithm-III (E-NSGA-III), in: Proc. 18th IEEE International Conference on Machine Learning and Applications, 2019.
[16]B.-Y. Liu, B.-W. Zhou, D.-S. Yang, J. Cao, T. Littler, Optimal Operation of Coastal Source-Load-Storage System with Seawater Desalination based on NSGA-III, in: Proc. 2020 IEEE Power & Energy Society General Meeting, 2020.
[17]X.-L. Zhu, K.-R. Hao, X.-S. Tang, T. Wang, Y.-C. Hua, X.-Y. Liu, The multi-objective optimization of esterification process based on improved NSGA-III algorithm, in: Proc. 12th Asian Control Conference, 2019.
[18]Y.-Y. Zhu, J.-W. Liang, J.-Y. Chen, Z. Ming, An improved NSGA-III algorithm for feature selection used in intrusion detection, Knowledge-Based Systems 116(2017) 74-85.
[19]J. Partanen, H. Haapasalo, Fast production for order fulfillment: Implementing mass customization in electronics industry, International Journal of Production Economics 90(2004) 213-222.
[20]B. Li, L.-P. Chen, Z.-D. Huang, Y.-F. Zhong, Product configuration optimization using a multiobjective genetic algorithm, International Journal of Advanced Manufacturing Technology 30(2006) 20-29.
[21]J.-M. Zhan, J.-J. Wang, X.-P. Wei, Research state and development directions of mass customization, Journal of Dalian University 23(6)(2002) 31-37.
[22]J. Wang, R. Li, H.-Y. Wu, G.-F. Ding, Y.-F. Zheng, Review of configuration design for complicated engineer-to-order products, Computer Integrated Manufacturing Systems 26(2)(2020) 410-425.
[23]H.-M. Du, A.-X. Zhang, W.-H. Ruan, X.-Y. Jing, Ant Colony Algorithm of Product Configuration in MassCustomization, Journal of Shanghai Dianjin University 15(3)(2012) 409-414.
[24]H.-J. Long, L.-Y. Wang, J. Shen, M.-X. Wu, Z.-B. Jiang, Product service system configuration based on support vector machine considering customer perception, International Journal of Production Research 51(18)(2013) 5450-5468.
[25]C.-M. Lei, M.-N. Zhou, Research on Multi-objective Product Configuration Optimization Considering EoL Product Reuse, Modular Machine Tool & Automatic Manufacturing Technique 4(2020) 9-13.
[26]S.-Y. Zhang, W.-Q. Ge, Z.-L. Wang, L.-M. Qiu, H.-F. Zhou, A heuristic configuration solving process planning method for mechanical product configuration by imitating the crystal crystallization process, International Journal of Advanced Manufacturing Technology 116(1-2)(2021) 611-628.
[27]M. Dong, D. Yang, L.-Y. Su, Ontology-based service product configuration system modeling and development, Expert Systems with Applications 38(2011) 11770-11786.
[28]S.-Y. Zhao, Q. Zhang, Z.-L. Peng, Y. Yan, Integrating customer requirements into customized product configuration design based on Kano's model, Journal of Intelligent Manufacturing 31(2020) 597-613.
[29]Y. Wang, W.-L. Zhao, W.-X. Wan, Needs-Based Product Configurator Design for Mass Customization Using Hierarchical Attention Network, IEEE Transactions on Automation Science and Engineering 18(1)(2021) 195-204.
[30]J.-K. Zhan, Y.-Q. Shi, B.-Z. Chen, Y.-K. Cai, Research on Optimization of Multi-Objective Product Configuration, Machinery Design & Manufacture 8(2020) 40-44.
[31]F.-Q. Cheng, F.-F. Ye, J.-G. Yang, Multi-objective optimization of collaborative manufacturing chain with time-sequence constraints, The International Journal of Advanced Manufacturing Technology 40(9)(2009) 1024-1032.
[32]F. Altiparmak, M. Gen, L. Lin, T. Paksoy, A genetic algorithm approach for multi-objective optimization of supply chain networks, Computers & Industrial Engineering 51(1)(2006) 196-215.
[33]S. Kuriakose, M.S. Shunmugam, Multi-objective optimization of wire-electro discharge machining process by Non-Dominated Sorting Genetic Algorithm, Journal of Materials Processing Technology 170(1)(2005) 133-141.
[34]Y.-W. Sang, J.-P. Tan, W. Liu, Research on Many-Objective Flexible Job Shop Intelligent Scheduling Problem Based on Improved NSGA-III, IEEE Access 8(2020) 157676-157690.

[35]I. Khettabi, L. Benyoucef, M.A. Boutiche, Sustainable reconfigurable manufacturing system design using adapted multi-objective evolutionary-based approaches, International Journal of Advanced Manufacturing Technology 115(11) (2021) 3741-3759.
[36]Q.-G. Liu, X.-X. Liu, J. Wu, Y.-X. Li, An Improved NSGA-III Algorithm Using Genetic K-Means Clustering Algorithm, IEEE Access 7(2019) 185239-185249.
[37]F. Xue, D. Wu, NSGA-III algorithm with maximum ranking strategy for many-objective optimization, International Journal of Bio-Inspired Computation 15(1)(2020) 14-23.
[38]Z. Wang, R.-Q. Lu, D.-B. Chen, F. Zou, An Experience Information Teaching–Learning-Based Optimization for Global Optimization, IEEE Transactions on Systems, Man, and Cybernetics: Systems 46(9)(2016) 1202-1214.
[39]J. Li, J.-Q. Zhang, C.-J. Jiang, M.-C. Zhou, Composite Particle Swarm Optimizer With Historical Memory for Function Optimization, IEEE Transactions on Cybernetics 45(10)(2015) 2350-2363.
[40]W.-F. Gao, F.T.S. Chan, L.-L. Huang, S.-Y. Liu, Bare bones artificial bee colony algorithm with parameter adaptation and fitness-based neighborhood, Information Sciences 316(2015) 180-200.
[41]S.-F. Han, K. Zhu, M.-C. Zhou, Competition-Driven Dandelion Algorithms With Historical Information Feedback, IEEE Transactions on Systems, Man, and Cybernetics: Systems 52(2)(2022) 966-979.
[42]G.-G. Wang, Y. Tan, Improve metaheuristic algorithms with information feedback models, IEEE Transactions on Cybernetics 49(2)(2019) 542-555.
[43]Y. Zhang, G.-G. Wang, K.-Q. Li, W.-C. Ye, M.-W. Jian, J.-Y. Dong, Enhancing MOEA/D with Information Feedback Models for Large-Scale Many-Objective Optimization, Information Sciences 522(2020) 1-16.
[44]Z.-M. Gu, G.-G. Wang, Improving NSGA-III algorithms with information feedback models for large-scale many-objective optimization, Future Generation Computer Systems 107(2020) 49-69.
[45]K. Deb, R. B. Agrawal, Simulated binary crossover for continuous search space, Complex Systems 9(4)(1995) 115-148.
[46]K. Deb, M. Goyal, A combined genetic adaptive search (GeneAS) for engineering design, Computer Science and Informatics 26(4)(1996) 30-45.
[47]X.-J. Bi, C. Wang, An improved NSGA-IIIS algorithm based on objective space decomposition for many-objective optimization, Soft Compute 21(15)(2017) 4269-4296.
[48]R. Cheng, Y.-C. Jin, B. Sendhoff, A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization, IEEE Transactions on Evolutionary Computation 20(5)(2016) 773-791.
[49]J.-D. Zhu, S.-H. Su, C. Chen, G.-Y. Liu, Improved Interactive Genetic Algorithm for Product Configuration Design, China Mechanical Engineering 29(20)(2018) 2474-2478.
[50]D.A.V. Veldhuizen, G.B. Lamont, Evolutionary Computation and Convergence to a Pareto Front, in: Proc. of the 1998 Genetic Programming Conference, 1998.
[51]P.A.N. Bosman, D. Thierens, The Balance Between Proximity and Diversity in Multiobjective Evolutionary Algorithms, IEEE transactions on evolutionary computation 7(2)(2003) 174-188.
[52]E. Zitzler, L. Thiele, Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach, IEEE transactions on evolutionary computation 3(4)(1999) 257-271.