symmetry

MDPI

Article

An Interactive Estimation of the Distribution Algorithm Integrated with Surrogate-Assisted Fitness

Zhanzhou Qiao [1], Guangsong Guo [2,*] and Yong Zhang [3]

[1] Basis Department, Liaocheng Vocational & Technical College, Liaocheng 252002, China; qiaozzlctu@163.com
[2] School of Intelligent Engineering, Zhengzhou University of Aeronautics, Zhengzhou 450015, China
[3] School of Information and Control Engineering, China University of Mining & Technology, Xuzhou 221116, China; yongzh401@cumt.edu.cn
* Correspondence: guogs78@126.com

Abstract: To accurately model user preference information and ensure the symmetry or similarity between real user preference and the estimated value in product optimization design, an interactive estimation of a distribution algorithm integrated with surrogate-assisted fitness evaluation (SAF-IEDA) is proposed in this paper. Firstly, taking the evaluation information of a few individuals as training data, a similarity evaluation method between decision variables is proposed. Following that, a preference probability model is built to estimate the distribution probability of decision variables. Then, the preference utility function of individuals is defined based on the similarity of decision variables. Finally, the surrogate-assisted fitness evaluation is realized by optimizing the weight of the decision variables' similarities. The above strategies are incorporated into the interactive estimation of the distribution algorithm framework and applied to address the optimal product design problem and the indoor lighting optimization problem. The experimental results demonstrate that the proposed method outperforms the comparative method in terms of search efficiency and fitness prediction accuracy.

Keywords: evolutionary algorithm; estimation of distribution algorithm; interactive; fitness prediction

check for updates

Citation: Qiao, Z.; Guo, G.; Zhang, Y. An Interactive Estimation of the Distribution Algorithm Integrated with Surrogate-Assisted Fitness. *Symmetry* **2023**, *15*, 1852. https://doi.org/10.3390/sym15101852

Academic Editors: Hsien-Chung Wu and Sergei D. Odintsov

Received: 4 September 2023
Revised: 20 September 2023
Accepted: 27 September 2023
Published: 2 October 2023

1. Introduction

Product optimization design (or optimal product design, OPD) refers to the optimization of product recommendation, market share, and product line configuration by collecting user preference information for product concepts [1]. This problem is prevalent in various domains, including product design [2,3], the supply chain [4], and personalized recommendations [5]. Among these, the collection of user preference information is a key factor in solving this problem and has a direct influence on the optimization results. Employing interactive techniques to collect user preference information is the most direct and widespread method. However, facing an extensive space solution, excessive interaction can lead to user fatigue. Therefore, how to help users quickly find satisfactory solutions in a vast sea of information has become the focus of much research.

The evolutionary algorithm-based Top-N algorithm is a representative approach for solving this problem [6]. The distinguishing feature of such algorithms is that users only need to evaluate N-out-of-M individuals in the population, while the remaining individuals are evaluated using surrogate models. The most commonly used surrogate models include polynomial regression [7], support vector machines [8], radial basis function (RBF) [9], neural networks [10], Kriging [11,12], and so on. These surrogate models are all statistical methods, and their learning and predictive performance are highly dependent on the size of the training set. If the training set is relatively small, the accuracy of learning and prediction tends to decrease.

Generally, a user evaluation result shows high uncertainty within interactive evolutionary algorithms. A user's evaluation result on the same individual may not be consistent across different evolutionary generations, while different users may assign the same evaluation value to different individuals in different evolutionary generations. This will seriously affect the convergence accuracy of an algorithm, causing the population to oscillate back and forth near the optimal solution. In other words, the algorithm's global optimization results are not a singular point but rather a neighborhood. In response to this characteristic, using probability distributions to describe user preference information can more objectively represent the relationships and distribution information among the variables within the search space, directly depicting the evolutionary trends of the overall population distribution. The estimation of the distribution algorithm (EDA) can utilize the global information of the solution space and the historical information during the evolution process. Through statistical learning, EDA predicts feasible regions and generates excellent new individuals by randomly sampling from probability models. EDA combines the characteristics of both genetic algorithms and statistical learning, possessing stronger evolution guidance, chaining learning ability, and global search capability. It was applied to optimization problems such as supermarket scheduling [13], path planning [14], and crane scheduling [15].

With the robust search performance of EDA, the involvement of decision makers in EDA (IEDA) can more efficiently utilize preference information to guide evolution, demonstrating superior search capabilities in personalized searches [16]. The core of IEDA lies in the surrogate-assisted fitness evaluation strategy. In [13], a differential evolution strategy and an adaptive learning rate mechanism were incorporated into EDA. In [17] it was indicated that using probability models to predict fitness has the advantage of being insensitive to the size of the training set and achieving higher learning and prediction accuracy. Currently, there is some utilization of user preferences as prior knowledge in IEDA to address personalized search problems [18,19]. However, research on integrating knowledge into probabilistic models remains limited. Among a few related studies, [14] introduced a probability model with Mallows distribution. In [15], a probability model was introduced using a distance-based ranking model and the moth–flame algorithm. In [20], utility functions were introduced to characterize preferences and apply them to multi-objective optimization. However, the utility function in this method focuses only on the optimization functions and overlooks the probabilistic relationship between the decision variables and the preferences. In [21], a novel hybrid approach was introduced that enhanced the structure of EDA through the inclusion of a lottery procedure, an elitism strategy, and a neighborhood search. However, this method did not use utility functions. Table 1 provides a summary of the mentioned literature on the EDA probability model and the surrogate-assisted fitness evaluation.

Clearly, if we can establish the probabilistic relationship between decision variables and preferences, and subsequently utilize utility functions for fitness prediction, this has the potential to enhance the surrogate-assisted fitness evaluation capability in IEDA. However, few studies to date have integrated probability models with utility functions in the context of EDA. In fact, employing machine learning methods to analyze data and guide searches has become a research direction for designing new EDA algorithms. Nevertheless, the introduction of machine learning and statistical learning also brings high time and space costs to evolutionary computation. As a result, achieving a balance between machine learning and evolutionary searches becomes crucial to efficiently and accurately address real-world optimization problems.

Given this context, this paper proposes a novel preference probability model and surrogate-assisted fitness evaluation method, aiming to achieve an interactive evolutionary solution for OPD within the EDA framework. Firstly, the Top-N algorithm is employed to collect user evaluation information on the Top-Nc individuals. Taking this information as the training samples, a preference probability model based on the similarity of decision variables is then established. Next, individual utility functions are calculated using the similarity of the decision variable, and these utility functions are then used to estimate fitness values for new individuals. Finally, the sampling frequency is updated using the preference probability model, and new individuals are generated through the sampling method in the EDA. Specifically, the main contributions of this paper are as follows:

(1) Introducing a probabilistic model to EDA. Unlike machine learning methods, the proposed method is particularly well-suited for small training data and boosts the efficiency of EDA significantly;

(2) Presenting a fitness estimation approach that markedly improves the precision of fitness prediction;

(3) Proposing a novel interactive distribution estimation algorithm that enhances the quality of the interactive evolutionary algorithm.

Table 1. Literature on the EDA probability model and surrogate-assisted fitness evaluation.

References	Algorithms	Problems
Zhou, et al. [13]	DE/AEDA	Differential evolution strategy and adaptive learning rate mechanism are integrated into EDA
Pérez-Rodríguez, et al. [14]	HMEDA	Mallows distribution probability model
Pérez-Rodríguez [15]	QCEDA	Probability model uses a distance-based ranking model and the moth–flame algorithm.
Chen Yang, et al. [16]	RBF-IEDA	Surrogate-assisted fitness evaluation-based RBF
Wang Leuohong, et al. [17]	APL-IEC	Algorithmic probability IEC
Chen Yang, et al. [18]	LMIEDA	Language model-based IEDA
Bao Lin, et al. [19]	RBM-EDA	Surrogate-assisted fitness evaluation-based RBM
Dewancker, I. et al. [20]	Active Utility Function Preference Learning	Utility function learning
Amalia Utamima [21]	AEDA	EDA plus a lottery procedure, an elitism strategy, and a neighborhood search

2. The Proposed Method

Considering the following OPD problem:

$$\max f(X) = \widetilde{f}(X),\ X = x_1 x_2 \ldots x_c,\ x_i \in S \subseteq R^c$$

where $\widetilde{f}(X)$ is an optimization performance indicator that cannot be expressed as a well-defined function; x_i represents the i-th decision variable of the solution X, and S is its range of values. $\widetilde{f}(X)$ reflects the qualitative expression of user preferences for X, embodying the mapping from solution space to psychological space. To implement the method presented in this paper, three main problems need to be addressed: preference probability model, surrogate-assisted fitness evaluation, and EDA algorithm design.

2.1. Preference Probability Model

When employing evolutionary algorithms to solve the OPD problem, products or commodities are typically regarded as individuals formed by combining different attribute values of decision variables (attributes). Let the population size be N, and the individuals in the population be denoted as X_i, where $i = 1, 2, \ldots, N$. Each X_i is composed of c decision variables, i.e., $X_i = (x_1, x_2, \ldots, x_c)$, where each decision variable $x_i,\ i = 1,\ 2,\ \ldots,\ c$, encompasses m distinct attribute values, i.e., $x_i = (x_{i,i_1}, x_{i,i_2}, \ldots, x_{i,i_m})$. Hence, solving the OPD problem can be conceptualized as the process of searching for good combinations of attribute values within each decision variable that satisfy user preferences. Let $P = \left\{x_{1,1_h}, x_{2,2_h}, \ldots, x_{c,c_h} \middle| 1 \leq h \leq m\right\}$ represent the set of all individual attribute values, we utilize the utility function $U(p), \forall p \in P$ to express the degree of user preference for an individual X_i; therefore, the attribute combination that satisfies the user, X_*, is defined as the specific individual for which $U(X_*) = U(p*) \geq U(p), \exists p* \in P, \forall p \in P, p \neq p*$.

Clearly, if the algorithm is effective, the distribution probabilities of P should align with user preferences. The subsequent sections will determine the probabilities of P based on the user-evaluated information. Section 2.3 will then use these probabilities to calculate $U(p)$ and estimate the fitness $\widetilde{f}(X_i)$ of an individual X_i.

During the process of interactive evolutionary optimization, let the Top-Nc individuals in the population be $X_1, X_2, \ldots, X_{Nc}$, and their user evaluation values be $\widetilde{f}(X_1), \widetilde{f}(X_2), \ldots, \widetilde{f}(X_{Nc})$. These values can serve as training samples for fitness prediction. Due to cognitive fuzziness, the preference characteristics of unevaluated individuals in the population can be reflected through the phenotypic similarity of variables. To comprehensively utilize evaluation information, the phenotypic similarity of decision variables is described using a Gaussian function and weighted aggregation:

$$\mu(x_i = x_{i,i_h}) = \frac{\sum\limits_{j=1}^{Nc} \widetilde{f}(X_j) e^{-(\frac{(x_{i,i_h} - x_{i,i_j})}{t(X_j)})^2}}{\sum\limits_{j=1}^{Nc} \widetilde{f}(X_j)} \tag{1}$$

where x_{i,i_h} represents the h-th value of the decision variable $x_i \in X^K, i = 1, 2, \ldots, c$, x_{i,i_j} means the j-th value of decision variable x_i for the evaluated individual X_j, and $t(X_j)$ denotes the evaluation time of individual X_j. Based on the equation above, a matrix of decision variable similarity $\boldsymbol{pim}$ can be established:

$$\boldsymbol{pim} = \begin{bmatrix} \mu(x_1 = x_{1,1}) & \mu(x_1 = x_{1,2}) & \ldots & \mu(x_1 = x_{1,m_i}) \\ \mu(x_2 = x_{2,1}) & \mu(x_2 = x_{2,2}) & \ldots & \mu(x_2 = x_{2,m_i}) \\ \vdots & \vdots & \vdots & \vdots \\ \mu(x_c = x_{c,1}) & \mu(x_c = x_{c,2}) & \ldots & \mu(x_c = x_{c,m_i}) \end{bmatrix} \tag{2}$$

In Equation (2), the larger the value of $\mu(x_i = x_{i,i_h})$ is, the higher attribute value the decision variable x_i has, and the higher the probability of retaining the attribute value x_{i,i_h} in the search space. Therefore, $\boldsymbol{pim}$ can be used to establish a preference probability model about P. Furthermore, we normalize each column of $\boldsymbol{pim}$ to yield the normalized preference probability model $\boldsymbol{pim}'$.

$$\boldsymbol{pim}' = \begin{bmatrix} \mu'(x_1 = x_{1,1}) & \mu'(x_1 = x_{1,2}) & \ldots & \mu'(x_1 = x_{1,m_i}) \\ \mu'(x_2 = x_{2,1}) & \mu'(x_2 = x_{2,2}) & \ldots & \mu'(x_2 = x_{2,m_i}) \\ \vdots & \vdots & \vdots & \vdots \\ \mu'(x_c = x_{c,1}) & \mu'(x_c = x_{c,2}) & \ldots & \mu'(x_c = x_{c,m_i}) \end{bmatrix} \tag{3}$$

Utilizing user preference information as input, $\boldsymbol{pim}'$ can dynamically depict the probability distribution of attribute values satisfied by customers, thereby providing a basis for determining $U(p)$. The variable annotations mentioned above are shown in Table 2.

Table 2. The meanings of variables.

N	Population size
X_i	Individual i in the population
x_i	The i-th decision variable of individual X
c	Number of decision variables
x_{i,i_m}	The i_m-th attribute value of decision variable x_i

Table 2. *Cont.*

m	Number of attribute values
P	Set of individual attribute values
X_*	Optimal individual (specific individual)
$U(p)$	Utility function of attribute p
Nc	Number of user evaluations of individuals
$\widetilde{f}(X_i)$	Fitness value of individual X_i
$t(X_j)$	Evaluation time for individual X_j
$\mu(x_i = x_{i,i_h})$	Phenotypic similarity when the value of decision variable x_i is x_{i,i_h}

2.2. Surrogate-Assisted Fitness Evaluation

The essence of surrogate-assisted fitness evaluation is to predict the fitness of unevaluated individuals by using information from evaluated ones. If the evaluated individual information reflects preferences, the surrogate-assisted fitness evaluation is equivalent to predicting the utility function for the unevaluated individuals. In other words, the utility function should maximize the reflection of preference levels to the greatest extent possible. For the unevaluated individual $X_{oi},\ i = 1,2,\ldots N - Nc$ in the population, since the similarity of each decision variable of X_{oi} reflects user preference, an additive utility function is chosen to express the user preference for X_{oi}:

$$U(X_{oi}) = \sum_{i=1}^{c} \omega_{oi}\mu(x_i = x_{i,i_h}) \tag{4}$$

In Equation (4), the decision variables $x_1,\ x_2,\ \ldots,\ x_c$ are assumed to be independent. ω_{oi} represents the weights of decision variable similarity and $\sum_{i=1}^{c} \omega_{oi} = 1$. The larger the value of $U(X_{oi})$, the higher the preference level for X_{oi}. Thus, the fitness of X_{oi} is given by:

$$\widetilde{f}(X_{oi}) = U(X_{oi}) \cdot \widetilde{f}_{\max} = \sum_{i=1}^{c} \omega_{oi}\mu(x_i = x_{i,i_h})\widetilde{f}_{\max} \tag{5}$$

where $\widetilde{f}_{\max}$ represents the maximum user evaluation value and the weight ω_{oi} is determined by minimizing the distance between X_{oi} and the reference point $[X_1, X_2, \ldots, X_{Nc}]^T$.

$$\begin{aligned} &\min\left\{ \sum_{j=1}^{Nc} \left| \sum_{i=1}^{c} \omega_{oi}\mu(x_i = x_{i,i_h})\widetilde{f}_{\max} - \widetilde{f}(X_j) \right|^{\lambda} \right\}^{1/\lambda} \\ &\text{s.}t.\ X_j \in G \end{aligned} \tag{6}$$

where λ is a real number greater than 1, and G represents the solution space. The determination of weights through Equation (6) ensures that $U(X_{oi})$ maximally reflects the preferences. It can be observed that with the update of pim, both $U(X_{oi})$ and $\widetilde{f}(X_{oi})$ will be updated. Since there is no need for population clustering, the surrogate-assisted fitness evaluation strategy incurs lower computational complexity.

2.3. Interactive Estimation of the Distribution Algorithm

The proposed method described in this paper is implemented within the framework of EDA. Firstly, an initial population is generated randomly and then the frequency of attribute values within the population is calculated. A higher frequency indicates that the corresponding attribute value aligns more with the user preferences. The attribute value with the highest frequency is then chosen as the initial preference probability vector $\boldsymbol{p}_0$ for IEDA:

$$\begin{aligned}\boldsymbol{p}_0 &= (p_1^0, p_2^0, \ldots p_c^0) \\ &= \operatorname{argmax}\left[P(x_1 \middle| X^K), P(x_2 \middle| X^K), \ldots, P(x_c \middle| X^K)\right]\end{aligned} \tag{7}$$

Subsequently, the individuals in the population are ranked based on their Pareto dominance considering the frequency of the attribute values. The Top-Nc individuals are then recommended to the user for evaluation. These *Nc* individuals form the elite individual set, which is used to update $\boldsymbol{p}_0$. The updating process includes the update of the preference probability model and the surrogate-assisted fitness evaluation. Finally, the updated preference probability vector $\boldsymbol{p} = (p_1, p_2, \ldots p_c)$ is used to sample and generate offspring individuals, and excellent individuals will be recommended to users for evaluation. This process is iterated until the algorithm has found satisfactory solutions according to user preferences, or the maximum number of generations is reached. The algorithm's structure is depicted in Figure 1.

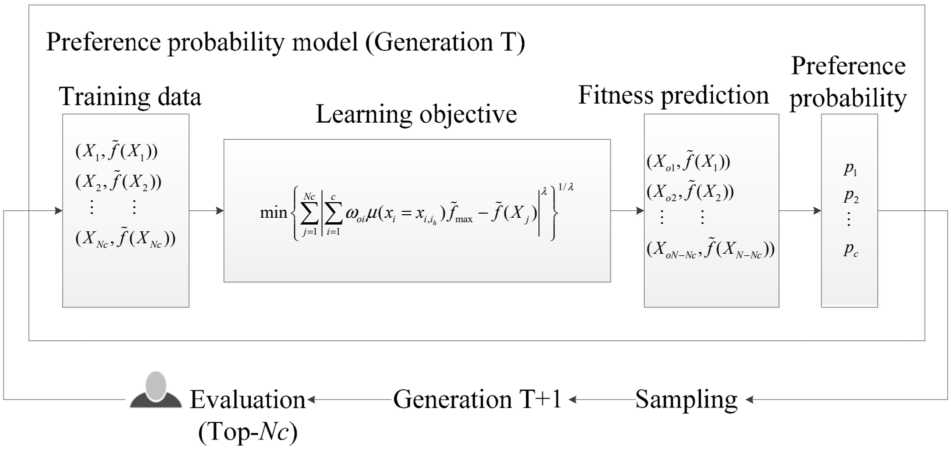

Preference probability model (Generation T)
Training data
$(X_1, \tilde{f}(X_1))$
$(X_2, \tilde{f}(X_2))$
$\vdots \quad \vdots$
$(X_{Nc}, \tilde{f}(X_{Nc}))$
Learning objective
$\min\left\{\sum_{j=1}^{Nc}\left|\sum_{i=1}^{c}\omega_{oi}\mu(x_i = x_{i,i_h})\tilde{f}_{\max} - \tilde{f}(X_j)\right|^{\lambda}\right\}^{1/\lambda}$
Fitness prediction
$(X_{o1}, \tilde{f}(X_1))$
$(X_{o2}, \tilde{f}(X_2))$
$\vdots \quad \vdots$
$(X_{oN-Nc}, \tilde{f}(X_{N-Nc}))$
Preference probability
p_1
p_2
$\vdots$
p_c
Evaluation (Top-*Nc*)
Generation T+1
Sampling

Figure 1. The framework of SAF-IEDA.

The computational complexity of SAF-IEDA is determined by the preference probability model and the surrogate-assisted fitness evaluation. If the number of evaluated individuals is Nc, the number of decision variables is c, and the number of attribute values is m, then the computational complexity for updating the preference probability model is $O(Nc)$. For the surrogate-assisted fitness evaluation, this requires calculating c utility functions for $N - Nc$ unevaluated individuals and Nc evaluated individuals, and the corresponding computational complexity is $O((N - Nc) \times Nc)$. Therefore, SAF-IEDA can complete one search and recommendation with a computational complexity of $O(Nc + (N - Nc) \times Nc)$. Since $(N - Nc) \times Nc >> Nc$, the computational complexity can be simplified to $O((N - Nc) \times Nc)$.

3. Experimental Study

3.1. Experimental Setup

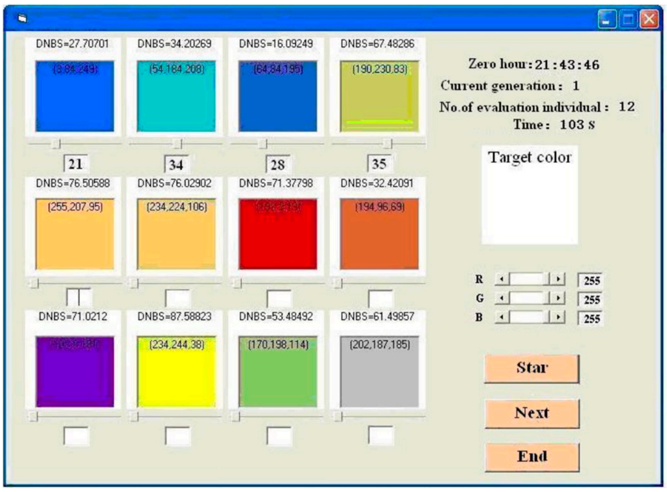
To validate the effectiveness of SAF-IEDA, this paper selects the RGB color one-max optimization problem [22] as a test case. The one-max optimization problem is a specific type of binary function; its objective is to maximize the number of gene positions containing a gene value of 1 within the binary string. The RGB color attributes are from 0 to 255, and its chromosome is encoded as a 24-bit binary string. The first eight bits represent the red attribute (R), the middle eight bits represent the green attribute (G), and the last eight bits represent the blue attribute (B). Each color attribute corresponds to a binary encoding range from 00000000 to 11111111. Clearly, the goal of the RGB color one-max optimization problem is to achieve white color, with attribute values of (255, 255, 255). When solving this problem using interactive evolutionary algorithms, higher user satisfaction implies individuals are closer to the target, making it a typical OPD problem. In this case, there are three decision variables (c = 3) and 256 attribute values (m = 256).

The evolutionary optimization system was developed using Visual Basic 6.0 and executed in the environment with an Intel(R) Xeon(R) E5-2660 V3 CPU at 2.60 GHz and 48 GB RAM. The interactive interface of SAF-IEDA is shown in Figure 2. The interface presents users with 12 color blocks, i.e., Nc = 12. Below each color block, an input textbox for fitness values is provided. Moreover, the system employs slider bars below each color block to record user evaluation time for calculating $t(X_j)$. Given that the RGB color space is visually non-uniform, the Munsell color space is introduced to measure the similarity between two colors. The color components are represented by Hue (H), Lightness (L), and Chroma (C). If the National Bureau of Standards unit (NBS) distance of an HLC color pair is less than 3.0, human vision perceives them as similar; if the NBS distance is greater than 6.0, they are regarded as significantly different [22]. Assuming two HLC color pairs, $X = (H_1, L_1, C_1)$ and $Y = (H_2, L_2, C_2)$, their NBS distance (DNBS) is defined as follows:

$$D_{\text{NBS}}(X,Y) = 1.2 \cdot \sqrt{2C_1C_2\left[1-\cos\left(\frac{2\pi}{100}\Delta H\right)\right] + (\Delta C)^2 + (4\Delta L)^2} \tag{8}$$

where $\Delta H = |H_1 - H_2|$, $\Delta L = |L_1 - L_2|$, $\Delta C = |C_1 - C_2|$. The process of converting from the RGB color space to the HLC color space is as follows. Firstly, the transformation from the RGB space to the xyz space is carried out [22]:

$$\begin{aligned} x &= 0.608R + 0.174G + 0.200B \\ y &= 0.299R + 0.587G + 0.144B \\ z &= 0.000R + 0.066G + 1.112B \end{aligned} \tag{9}$$



DNBS=27.70701
(3,84,249)
21

DNBS=34.20269
(54,184,208)
34

DNBS=16.09249
(64,84,195)
28

DNBS=67.48286
(190,230,83)
35

DNBS=76.50588
(255,207,95)

DNBS=76.02902
(234,224,106)

DNBS=71.37798
[illegible]

DNBS=32.42091
(194,96,69)

DNBS=71.0212
[illegible]

DNBS=87.58823
(234,244,38)

DNBS=53.48492
(170,198,114)

DNBS=61.49857
(202,187,185)

Zero hour: 21:43:46
Current generation: 1
No.of evaluation individual: 12
Time: 103 s

Target color

R 255
G 255
B 255

Figure 2. The system's interactive interface.

Subsequently, the transformation from the xyz space to the $p'q'$ space is performed:

$$p' = f(x) - f(y), q' = 0.4[f(z) - f(y)] \tag{10}$$

where $f(x) = 11.6x^{1/3} - 1.6$. Further, the transformation is performed to convert the $p'q'$ space to the $s't'$ space:

$$s' = (a + b\cos\theta)p, t' = (c + d\sin\theta)q \tag{11}$$

where $\theta = \arctan(p'/q'), a = 8.880, b = 0.966, c = 8.025, d = 2.558$. Finally, the resulting transformation is:

$$H = \arctan(s'/t'), L = f(y), C = \sqrt{s'^2 + t'^2} \tag{12}$$

In the experiments, if $D_{\text{NBS}} \leq 3.0$, we believe two colors are similar. A smaller D_{NBS} value indicates a higher similarity between the two color pairs. Thus, this value can also measure the prediction accuracy. A target color block is set on the periphery of each color block, displaying the similarity distance value. The user clicks the "Start" button, and the system generates an initial population randomly. The user evaluates the attributes of the Top-Nc individuals, which forms the initial preference probability vector $\boldsymbol{p}_0$ for EDA. Then, by clicking the "Next" button, the system establishes a preference probability model $\boldsymbol{pim}'$ based on the Top-Nc individuals and their evaluations according to Equation (3) in the background and estimates the fitness values for other unevaluated individuals based on Equation (5). Finally, a new population is generated by sampling according to $\boldsymbol{pim}'$. For this new population, frequencies of attribute values are calculated, and Pareto dominance sorting is performed based on these frequencies. The Top-Nc individuals are recommended as excellent individuals for further evaluation by the user and will serve as a dominant set used to update the EDA preference probability vector. The cycle continues until the termination condition is satisfied, upon which the "End" button can be clicked to end the evolution. The termination conditions for the algorithm are as follows: (1) $D_{\text{NBS}} \leq 3.0$, which indicates two colors are within the minimum just-noticeable difference, and the population has produced colors that are very close to the target color or (2) the user has discovered a satisfactory individual, or half of the individuals to be evaluated by users are the same, or the user feels fatigued.

In this study, four representative algorithms are selected as comparison algorithms to validate the effectiveness of SAF-IEDA. These algorithms include:

(1) The traditional interactive genetic algorithm (IGA);
(2) The Kano-integrated interactive genetic algorithm (Kano-IGA), proposed in [23];
(3) The interactive genetic algorithm with BP neural network-based user cognitive surrogate model (BP-IGA), proposed in [24];
(4) An interactive estimation of the distribution algorithm with RBF neural network-based fitness evaluation (RBF-IEDA), proposed in [16].

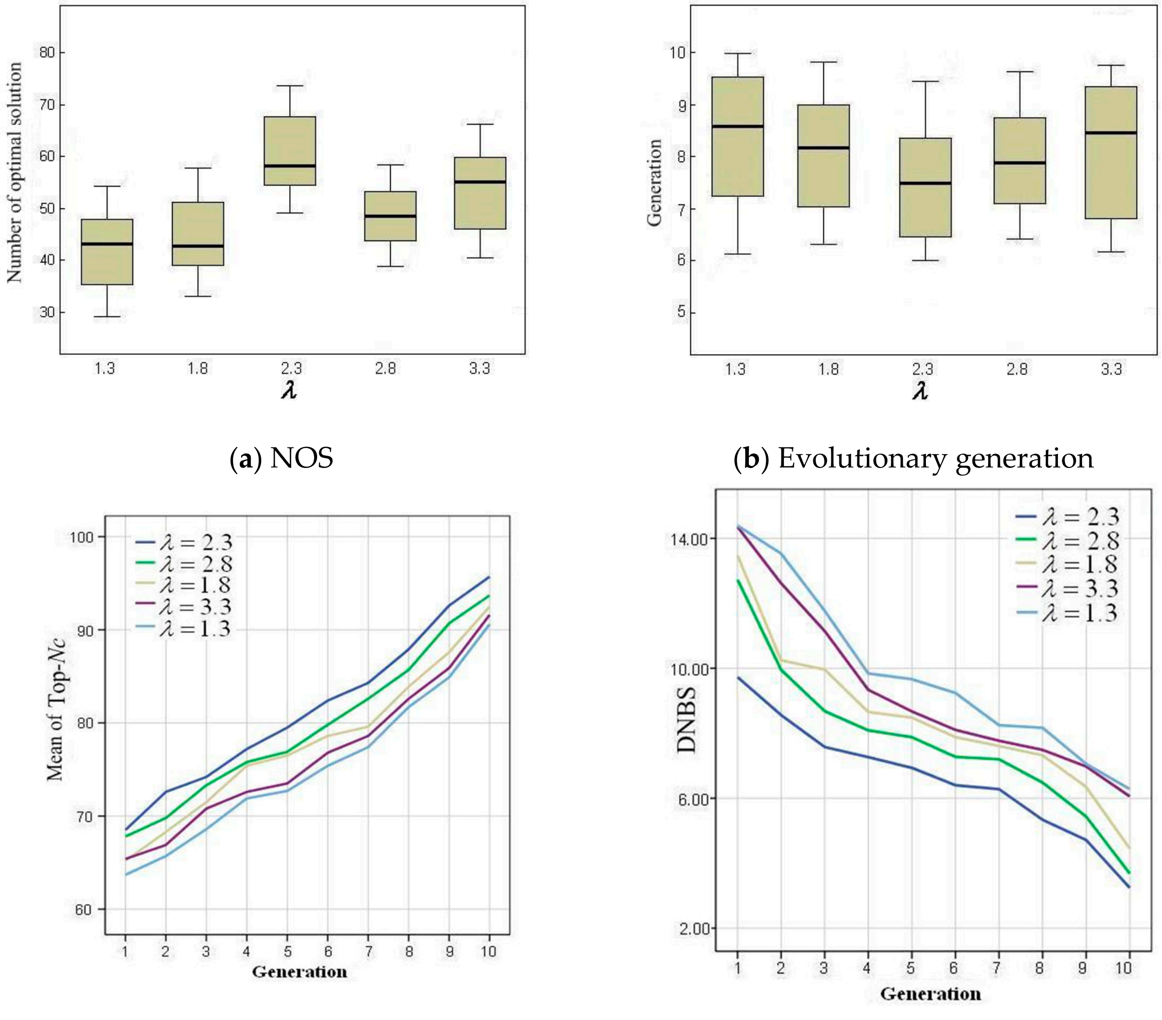
To perform the comparison, five male and five female university students without visual impairments are chosen as test users, labeled as User 1 to User 10. The optimization system is applied, and SAF-IEDA along with the comparison methods is run separately 20 times. The average results are calculated for each run and subjected to comparative analysis.

In all methods, the population size is set to be N = 200 and the maximum number of generations is set to be T = 10. For SAF-IEDA, the user's fitness evaluation range is set as integers from 1 to 99. IGA adopts the k-means clustering strategy, the roulette wheel selection, the single-point crossover with a probability of 0.85, and the mutation with a probability of 0.05. BP-IGA takes a three-layer BP network with a configuration of 12-2-7 as the learning model for surrogate-assisted fitness evaluation and uses the same single-point crossover and mutation as IGA. Kano-IGA utilizes the Kano model to calculate preference values for individual attributes and recommends individuals that align with user satisfaction. This is a relatively new interactive evolutionary algorithm, and its single-point crossover and mutation probabilities are set as 1 and 0.05, respectively. RBF-IEDA constructs a preference surrogate-assisted model by using an RBF network and implements evolutionary optimization based on IEDA, according to the suggestions in reference [14].

3.2. Results and Analysis

3.2.1. The Parameter λ

The parameter λ in Equation (6) directly impacts the computation of the utility function. Thus, it is essential to investigate the influence of λ on SAF-IEDA performance. Figure 3a,b show the NOS (the number of optimal solutions) and NG (the number of generations) values obtained using SAF-IEDA under different λ (1.3, 1.8, 2.3, 2.8, and 3.3). We can see that both excessively small and large λ lead to a decrease in NOS and an increase in the number of generations. The fitness values of the Top-Nc individuals are analyzed statistically, as shown in Figure 3c. A high value indicates better quality for the recommended individuals. From Figure 3c, it can be observed that excessively small or large λ lead to a decline in the quality of recommended individuals. Figure 3d illustrates the variations of D_{NBS} under different λ. As the evolution progresses, D_{NBS} gradually decreases. Based on the comprehensive experimental results, λ = 2.3 is selected as the optimal value. Table 3 provides the algorithm's metric values, including means and standard deviations (STD). The success rate (SR) is defined as the number of successful experiments divided by the total number of experiments. As seen in Table 1, NOS for all users is greater than 160, and the SR is higher than 95%. This indicates that SAF-IEDA effectively solves the RGB color one-max optimization problem.



(**a**) NOS

(**b**) Evolutionary generation

(**c**) The average fitness of Top-Nc

(**d**) D_{NBS} variation curve

Figure 3. The influence of λ on the proposed method.

Table 3. The metric values obtained using SAF-IEDA when $\lambda = 2.3$.

Users	Evolutionary Generation	Fitness of Top-Nc	D_{NBS}	Runtime (s)	NOS	SR (%)
Use 1	7.7 (1.033)	95.70 (3.561)	4.33 (0.687)	181.30 (20.548)	167 (11.253)	96.24 (0.405)
Use 2	7.3 (0.816)	94.56 (2.351)	5.71 (0.592)	188.56 (22.562)	162 (13.052)	95.62 (0.561)
Use 3	7.0 (0.876)	95.21 (2.803)	4.04 (0.754)	179.63 (19.554)	165 (12.455)	96.53 (0.684)
Use 4	7.4 (0.948)	96.89 (3.149)	4.31 (0.653)	168.37 (18.615)	182 (14.027)	96.37 (0.515)
Use 5	7.0 (0.994)	96.43 (3.153)	3.65 (0.626)	175.48 (18.327)	172 (13.587)	95.75 (0.632)
Use 6	7.6 (0.949)	95.82 (2.362)	4.28 (0.654)	187.56 (22.355)	166 (14.396)	95.83 (0.426)
Use 7	7.4 (0.843)	96.87 (3.257)	5.23 (0.735)	194.25 (22.452)	189 (12.473)	95.85 (0.654)
Use 8	7.5 (0.738)	97.02 (1.758)	3.36 (0.593)	186.56 (20.354)	192 (12.951)	96.58 (0.488)
Use 9	7.7 (0.865)	96.26 (2.354)	4.56 (0.827)	179.19 (24.228)	176 (11.558)	95.51 (0.584)
Use 10	7.3 (0.952)	97.28 (12.528)	4.27 (0.774)	167.63 (15.736)	187 (13.825)	96.46 (0.752)

3.2.2. The Result Analyses of SAF-IEDA with the Comparison Methods

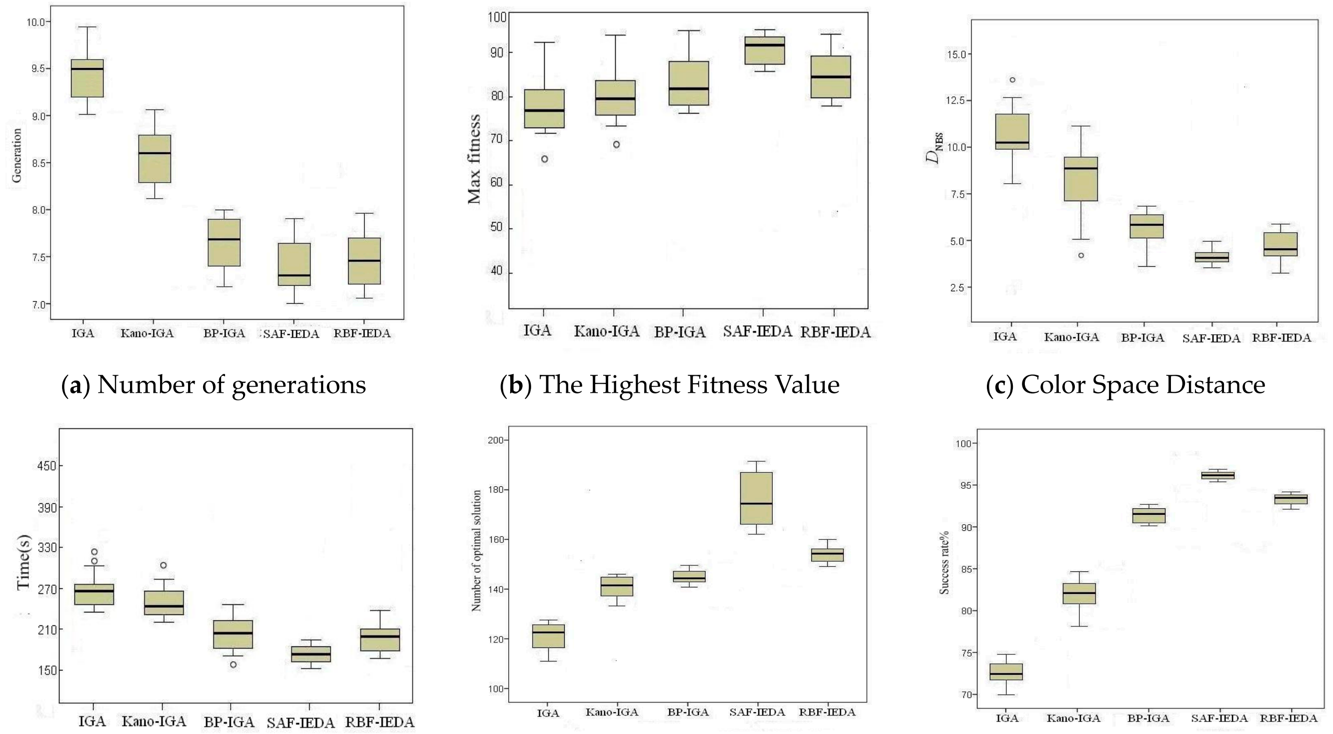
Figure 4 shows the comparison results between SAF-IEDA with the four comparative methods on the six metrics. Here, the circle "o" represents exception values. These six metrics include the number of generations, the highest fitness value, the color space distance, the runtime, NOS, and SR. We can see that SAF-IEDA displays the fewest generations, the highest fitness value, the smallest color space distance, the shortest runtime, as well as the highest NOS and SR values, highlighting the superiority of our method. The reason is that the fitness prediction model based on the preference probability model adopted in this paper is more accurate compared to other methods. Additionally, compared to GA, the EDA-based search engine used in SAF-IEDA has better search performance. In Figure 4c, it is evident that SAF-IEDA demonstrates the highest fitness prediction accuracy. This can be attributed to the insensitivity of SAF-IEDA to the number of training samples, compared with Kano, BP, RBF, etc. During the initial stages of evolution, the low number of training samples for Kano, BP, and RBF leads to poorer fitness prediction accuracy. In contrast, SAF-IEDA retains good predictive performance with fewer training samples. As the number of iterations increases, the number of training samples also increases; therefore, the predictive accuracy of all methods gradually improves.

Using the Wilcoxon signed-rank test, the results are shown in Table 4. By observing the positive or negative nature of the *t*-values, we can see that SAF-IEDA outperforms the comparison methods in terms of all metrics. On a two-sided significance indicator, although SAF-IEDA does not exhibit significant differences from BP-IGA and RBF-IEDA in terms of the number of generations, there are significant differences between SAF-IEDA and the comparative methods.

Table 4. The results of Wilcoxon signed-rank test.

Algorithms	Evolutionary Generation	Fitness of Top-Nc	D_{NBS}	Runtime (s)	NOS	SR (%)
IGA vs. SAF-IEDA	+	+	+	+	+	+
Kano-IGA vs. SAF-IEDA	+	+	+	+	+	+
BP-IGA vs. SAF-IEDA	+	+	+	+	+	+
RBF-IEDA vs. SAF-IEDA	+	+	+	+	+	+

SAF-IEDA can achieve high-quality solutions with the lowest errors under the same number of iterations, compared with BP-IGA and RBF-IEDA. This further validates the superiority of SAF-IEDA. Although Kano-IGA also updates the population based on individual attribute preference values dynamically, its metric values are less than SAF-IEDA, suggesting that SAF-IEDA is more accurate in characterizing attribute preferences, and the quality of recommended individuals is higher.



(**a**) Number of generations (**b**) The Highest Fitness Value (**c**) Color Space Distance

(**d**) Runtime (**e**) SR (**f**) NOS

Figure 4. Metric comparisons of 5 methods.

Figure 5 illustrates the convergence curves of the five methods. It is evident that all methods are able to converge, but SAF-IEDA achieves the highest fitness values and shows better convergence.

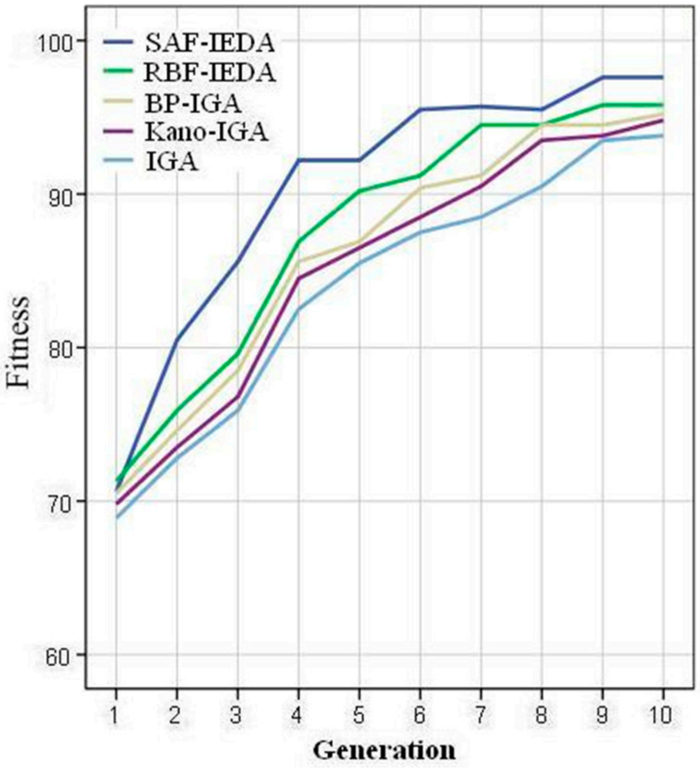


Figure 5. The convergence curves of the 5 methods.

3.2.3. Application to the Indoor Lighting Optimization

To further validate the effectiveness of SAF-IEDA, SAF-IEDA is applied to optimize the indoor lighting problem. In this scenario, an individual represents a combination of three LED color lights, each comprising three color channels: red (R), green (G), and blue (B). The optimization objective is to create the most favorable light and shadow effects by combining these three-color lights. The RGB variables collectively form a chromosome with a total of nine color attributes. Each RGB variable can take on values in the range of 0 to 255, and these attribute values are encoded using 8-bit binary code. Consequently, a chromosome comprises 72 bits of binary code. Refer to Figure 6 for a visual representation of the chromosome encoding process. It is important to note that the indoor lighting optimization problem involves a multitude of individual attributes and attribute values, rendering it a complex problem suited for resolution using our method.

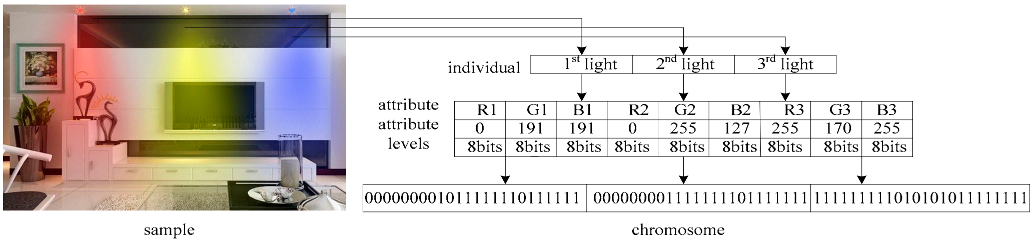


Figure 6. Individual and chromosome encoding.

Ten users are tasked with optimizing indoor lighting schemes to achieve a 'warm' style. These users are not constrained by a maximum number of evolution generations. Comparative methods are run independently ten times, and the results of independent sample analyses for different evolution indicators are presented in Table 5. The Levene's test results for each indicator indicate that there are no significant differences in variances among the samples, as all p-values were greater than 0.05. However, there is a notable significant difference in the number of evolution generations required and the running time between SAF-IEDA and the comparative methods. This suggests that when there is no restriction on the maximum number of evolution generations, Kano-IGA, BP-IGA, and RBF-IEDA require more evolution generations and more time to achieve high-quality satisfactory solutions, likely due to the high training costs associated with their models. IGA performs the worst and requires more evolution generations to obtain satisfactory solutions. There is a significant difference in terms of the highest fitness value, color space distance, and optimal solution quantity between SAF-IEDA and the comparative methods, indicating that it is suitable for personalized optimization objectives. Therefore, SAF-IEDA can achieve higher-quality solutions and better interactivity.

Table 5. Analysis of evolutionary indicators using different methods.

Independent Sample Test p-Value (Two-Sided)	**Evolutionary Generation**	**Fitness of Top-Nc**	D_{NBS}	**Runtime (s)**	**NOS**
	SAF-IEDA	**SAF-IEDA**	**SAF-IEDA**	**SAF-IEDA**	**SAF-IEDA**
Kano-IGA	0.018	0.023	0.028	0.017	0.001
BP-IGA	0.037	0.033	0.020	0.003	0.012
RBF-IEDA	0.028	0.032	0.042	0.035	0.014
IGA	0.013	0.028	0.039	0.016	0.028
Levene test for equality of variances	{0.352, 0.637, 0.256, 382}	{0.385, 0.417, 0.228, 0.673}	{0.753, 0.525, 0.257, 0.702}	{0259, 0.450, 0.441, 0.335}	{0.204, 0.198, 0.657, 0.284}

Based on the above analysis, it is evident that SAF-IEDA outperforms the comparative methods for various optimization objectives. In all comparison indicators, SAF-IEDA consistently delivers the most satisfactory optimization solutions. Therefore, it can be concluded that SAF-IEDA is superior to comparative methods in addressing the indoor lighting optimization problem effectively.

4. Conclusions

This paper proposed a preference probability model and a surrogate-assisted fitness evaluation strategy under the framework of an interactive distribution estimation algorithm. In traditional EDA, individual fitness is an explicit function value, and new individuals are generated using sampling based on the probability model of individual attributes. In an interactive evolutionary environment, individual fitness needs to be set by the user, and it reflects the user's preferences. In this case, the distribution of individual attributes is consistent with the preference distribution. User preferences depend on the similarity of individual phenotypes, which can be described by Gaussian functions for decision variables. In this way, a preference probability model can be established based on the similarity of decision variables, and generating new individuals using this preference probability model will better align with user preferences. On the other hand, in order to alleviate fatigue, utility functions are used to express attribute preferences and weighted estimates of fitness values to better reflect consistency between attributes and preference distributions. The strategy proposed in this article is an extension of EDA applied in an interactive evolutionary environment. This method is based on distribution estimation algorithms, uses a preference probability model as sampling probabilities, and utilizes Top-Nc individuals to construct a preference probability vector for evaluating fitness values based on the utility function.

The experimental results on the RGB color one-max optimization problem and the indoor lighting optimization problem show that users can obtain the most satisfactory optimization solution using SAF-IEDA, compared with five existing comparison algorithms. This paper only deals with single-objective optimal problems; in-depth research is needed for cases with multiple objectives. In addition, applying the proposed method to more practical optimal product design problems is also an open issue.

Author Contributions: Writing, conceptualization, Z.Q.; Software, conceptualization, supervision, G.G.; Review and editing, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key Research and Development Program of China (No.2020YFB1708200).

Data Availability Statement: The datasets generated and analyzed during the current study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pantourakis, M.; Tsafarakis, S.; Zervoudakis, K.; Altsitsiadis, E.; Andronikidis, A.; Ntamadaki, V. Clonal selection algorithms for optimal product line design: A comparative study. *Eur. J. Oper. Res.* **2022**, *298*, 585–595. [CrossRef]
2. Guido, M. Optimal Product Design: Implications for Competition and Growth under Declining Search Frictions. *Natl. Bur. Econ. Res.* **2021**, *91*, 28638.
3. Zheng, R.; Zhang, Y.; Sun, X.; Wang, F.; Yang, L.; Peng, C.; Wang, Y. Multi-objective particle swarm optimization of complex product change plan considering service performance. *CAAI Trans. Intell. Technol.* **2022**, *8*, 1058–1076. [CrossRef]
4. Xue, K.; Sun, G.; Wang, Y.; Niu, S. Optimal Pricing and Green Product Design Strategies in a Sustainable Supply Chain Considering Government Subsidy and Different Channel Power Structures. *Sustainability* **2021**, *13*, 12446. [CrossRef]
5. Wang, R.; Jiang, Y.; Lou, J. Attention-based dynamic user preference modeling and nonlinear feature interaction learning for collaborative filtering recommendation. *Appl. Soft Comput.* **2021**, *110*, 107652. [CrossRef]
6. Lee, J.; Hwang, W.S.; Parc, J.; Lee, Y.; Kim, S.-W.; Lee, D. Injection: Toward Effective Collaborative Filtering Using Uninteresting Items. *IEEE Trans. Knowl. Data Eng.* **2019**, *31*, 3–16. [CrossRef]
7. Kim, T.; Han, W.S.; Piao, J.; Kang, P.K.; Shin, J. Predicting remediation efficiency of LNAPLs using surrogate polynomial chaos expansion model and global sensitivity analysis. *Adv. Water Resour.* **2022**, *163*, 104179. [CrossRef]

8. Mohammadi, Y.; Salarpour, A.; Leborgne, R.C. Comprehensive strategy for classification of voltage sags source location using optimal feature selection applied to support vector machine and ensemble techniques. *Int. J. Electr. Power Energy Syst.* **2021**, *124*, 106363. [CrossRef]
9. Wang, W.Y.; Akhtar, T.; Shoemaker, C.A. Integrating ε-dominance and RBF surrogate optimization for solving computationally expensive many-objective optimization problems. *J. Glob. Optim.* **2022**, *82*, 965–992. [CrossRef]
10. Stuckner, J.; Piekenbrock, M.; Arnold, S.M.; Ricks, T.M. Optimal experimental design with fast neural network surrogate models. *Comput. Mater. Sci.* **2021**, *200*, 110747. [CrossRef]
11. Chao, R.; Younes, A.; Didier, L. Ensemble of surrogates combining Kriging and Artificial Neural Networks for reliability analysis with local goodness measurement. *Struct. Saf.* **2022**, *96*, 102186.
12. Ji, X.; Zhang, Y.; Gong, D.; Sun, X.; Guo, Y. Multisurrogate-assisted multitasking particle swarm optimization for expensive multimodal problems. *IEEE Trans. Cybern.* **2023**, *53*, 2516–2530. [CrossRef] [PubMed]
13. Zhou, B.H.; Tan, F. A self-adaptive estimation of distribution algorithm with differential evolution strategy for supermarket location problem. *Neural Comput. Appl.* **2020**, *32*, 5791–5804. [CrossRef]
14. Pérez-Rodríguez, R.; Hernández-Aguirre, A. A hybrid estimation of distribution algorithm for the vehicle routing problem with time windows. *Comput. Ind. Eng.* **2019**, *130*, 75–96. [CrossRef]
15. Pérez-Rodríguez, R. A hybrid estimation of distribution algorithm for the quay crane scheduling problem. *Math. Comput. Appl.* **2021**, *26*, 64. [CrossRef]
16. Chen, Y.; Sun, X.Y.; Gong, D.W.; Zhang, Y.; Choi, J.; Klasky, S. Personalized Search Inspired Fast Interactive Estimation of Distribution Algorithm and Its Application. *IEEE Trans. Evol. Comput.* **2017**, *21*, 588–600. [CrossRef]
17. Wang, L.H.; Liao, J.D. An Interactive Evolutionary Computation Framework: A Case Study of the Optimal Product Design. *J. E-Bus.* **2011**, *13*, 77–98.
18. Chen, Y.; Jin, Y.; Sun, X. Language model based interactive estimation of distribution algorithm. *Knowl. Based Syst.* **2020**, *200*, 105980. [CrossRef]
19. Bao, L.; Sun, X.; Chen, Y.; Man, G.; Shao, H. Restricted boltzmann machine-assisted estimation of distribution algorithm for complex problems. *Complexity* **2018**, *2018*, 609014. [CrossRef]
20. Dewancker, I.; McCourt, M.; Ainsworth, S. Interactive Preference Learning of Utility Functions for Multi-Objective Optimization. In Proceedings of the Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016.
21. Utamima, A. A comparative study of hybrid estimation distribution algorithms in solving the facility layout problem. *Egypt. Inform. J.* **2021**, *22*, 505–513. [CrossRef]
22. Gong, Y.H. Advancing Content-based Image Retrieval by Exploiting Image Color and Region Features. *Multimed. Syst.* **1999**, *7*, 449–457. [CrossRef]
23. Dou, R.; Zhang, Y.; Nan, G. Application of combined Kano model and interactive genetic algorithm for product customization. *J. Intell. Manuf.* **2019**, *30*, 2587–2602. [CrossRef]
24. Lv, J.; Zhu, M.; Pan, W. Interactive Genetic Algorithm Oriented toward the Novel Design of Traditional Patterns. *Information* **2019**, *10*, 36. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.