Hindawi Publishing Corporation
Applied Computational Intelligence and Soft Computing
Volume 2013, Article ID 302573, 16 pages
http://dx.doi.org/10.1155/2013/302573

Research Article

Crossover Method for Interactive Genetic Algorithms to Estimate Multimodal Preferences

Misato Tanaka,[1] Yasunari Sasaki,[2] Mitsunori Miki,[3] and Tomoyuki Hiroyasu[4]

[1] *Graduate School of Engineering, Doshisha University, 1-3 Tatara Miyakodani, Kyotanabe-shi, Kyoto 610-0394, Japan*
[2] *Kanazawa Seiryo University Women's Junior College, 10-1 Ushi, Gosho-machi, Kanazawa-shi, Ishikawa 920-8620, Japan*
[3] *Faculty of Science and Engineering, Doshisha University, 1-3 Tatara Miyakodani, Kyotanabe-shi, Kyoto 610-0394, Japan*
[4] *Faculty of Life and Medical Sciences, Doshisha University, 1-3 Tatara Miyakodani, Kyotanabe-shi, Kyoto 610-0394, Japan*

Correspondence should be addressed to Misato Tanaka; mtanaka@mikilab.doshisha.ac.jp

Received 10 September 2013; Accepted 1 December 2013

Academic Editor: Shyi-Ming Chen

We apply an interactive genetic algorithm (iGA) to generate product recommendations. iGAs search for a single optimum point based on a user's Kansei through the interaction between the user and machine. However, especially in the domain of product recommendations, there may be numerous optimum points. Therefore, the purpose of this study is to develop a new iGA crossover method that concurrently searches for multiple optimum points for multiple user preferences. The proposed method estimates the locations of the optimum area by a clustering method and then searches for the maximum values of the area by a probabilistic model. To confirm the effectiveness of this method, two experiments were performed. In the first experiment, a pseudouser operated an experiment system that implemented the proposed and conventional methods and the solutions obtained were evaluated using a set of pseudomultiple preferences. With this experiment, we proved that when there are multiple preferences, the proposed method searches faster and more diversely than the conventional one. The second experiment was a subjective experiment. This experiment showed that the proposed method was able to search concurrently for more preferences when subjects had multiple preferences.

1. Introduction

At present, in the e-commerce of business-to-consumer, product recommendation is very important. The number of products sold on online shopping sites is increasing. Moreover, to improve sales, each site uses search techniques or recommendations to display its products. Because search techniques take into account user direct input, they return products that users expect. In contrast, because product recommendation techniques use action logs of users to analyze their needs, they display products that users do not expect. At present, the main recommendation techniques are contents based filtering [1, 2] and collaborative filtering [3–5]. The former recommends products by matching a user's profile and action logs with features of products, whereas the latter recommends products on the basis of frequency they are bought at the same time. Therefore, we aim at displaying products that fit a personal Kansei model. Kansei is a Japanese term that relates to human characteristics such as sensibility, perception, affection, or subjectivity. We assume that human Kansei is modeled as a function. The input parameters of the function are the features of objects or the factors of environment and the output parameters are subjective evaluations such as preference or impression. This internal model in human Kansei is able to be analyzed from a user's action log [6]. Hence, by searching for the maximum point of this function, we can find the objects that match the subjective evaluations. These search techniques are termed interactive evolutionary computations (iECs) [7]. The interactive evolutionary strategy (iES) [8] and interactive genetic programming (iGP) [9–11] are the two types of iEC techniques. In this study, we focus on interactive genetic algorithms (iGAs) [12–14] because they use multiple search points and the solutions of problem are encoded relatively easily.

iGAs are designed to search for a single optimum point. However, in the problem of product recommendation, there may be multiple optimum points having almost the same

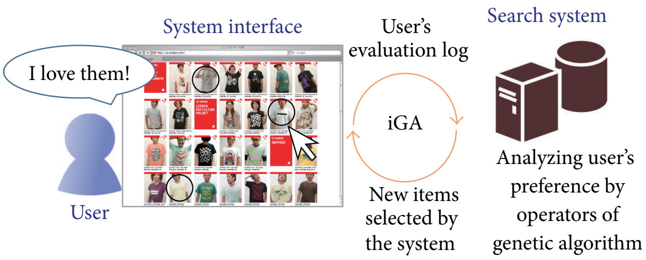

System interface
User's evaluation log
Search system
I love them!
iGA
User
New items selected by the system
Analyzing user's preference by operators of genetic algorithm

Figure 1: Overview of a product recommendation system using an iGA.

evaluation values. For example, several users of a recommendation system for T-shirts may like both blue and white colors, as well as dotted and striped patterns. Users often like more than two products that are distant to each other in the product parameter space. Therefore, if we apply iGA simply, we will not be able to correctly extrapolate the Kansei model of a user. To address this problem, we developed a new iGA crossover method that concurrently searches for multiple preferences of a user. This method consists of two phase's estimation of the locations of the optimum area and search for the maximum values of the area. The former is enabled by the clustering method. By dividing the highly valued solutions by a clustering method, several areas having high evaluation can be obtained. Moreover, by searching inside the areas by a probabilistic model reflecting the distribution of highly valued solutions, the more highly evaluated solutions can be found.

In Section 2, we provide an overview of iGAs. A detailed description of our proposed method is discussed in Section 3. In Section 4, we use a pseudouser to compare the performance of our proposed method with that of conventional methods. Finally, the effectiveness of our proposed method is confirmed in a subjective experiment in Section 5.

2. Interactive Genetic Algorithms

2.1. Outline. iGA is an optimization method based on GAs, which emulate evolution [15]. The optimal objects produced by iGAs match the Kansei of a user by replacing the objective function of a GA with the subjective evaluation of a user which is based on preference or impression. Therefore, iGAs are used in applications that should embed Kansei information in evaluations, such as fashion design [16, 17], user interface layout [18, 19], and hearing aid fitting [20, 21].

Figure 1 shows an overview of the recommendation system using iGAs. In this system, the products are evaluated according to the preferences of the user, likes or dislikes, wants or unwants, and so on. The system analyzes the evaluation log of the user and selects new products to display, which are also evaluated by the user. As this process of evaluation and analysis is repeated, the products displayed evolve toward those the user likes the most.

2.2. Algorithm. GAs use evolution factors such as natural selection and generation alternation to find an optimum object. These evolution factors are modeled as GA operator, namely, selection, crossover, and mutation. iGAs differ from GAs in that the evaluation values of solutions these operators use are the ratings provided by the user. Figure 2 shows a flowchart depicting the optimization process of iGAs. The selection operator extracts the solutions that are highly evaluated to be the parent solutions for next generations. The crossover operator emulates generation alternation and produces offsprings from selected parents. Finally, the mutation operator randomly changes the parts of the chromosome representing a solution to prevent solutions from converging on local optimum points.

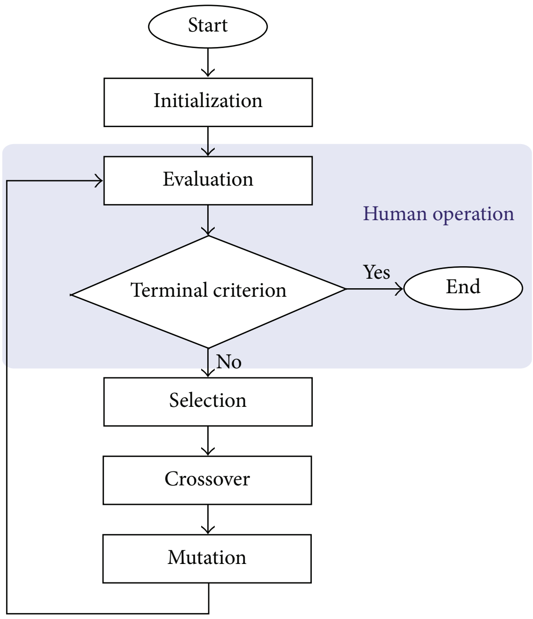


Figure 2: Flowchart of iGA optimization.

The representations of solutions are different between the evaluation and the other GA operations. The representation in the evaluation is called phenotype, which is a numeric set representing the features of a solution such as color, length, and shape. A user evaluates the solution made of this form. The other representation is called genotype, which is an encoded form of the phenotype to adapt to GA operators. Because the generated offspring are represented as genotype, the encoding and decoding between genotypes and phenotypes are required to be approximately linear with respect to human Kansei. In our approach, we use real-coded GAs [22–24] because the solutions can use the almost same representation as phenotype and genotype representations. Hereinafter, the representation of a solution will be referred to as design variable for descriptive purposes.

2.3. iGA for Multimodal Preferences. The conventional iGA is effective in searching for a single optimum on landscapes containing one or multiple peaks. A landscape is an objective function and visualizes a relationship between design variables and evaluation value by ranges of mountains. It expresses the features and complexity of a problem. In

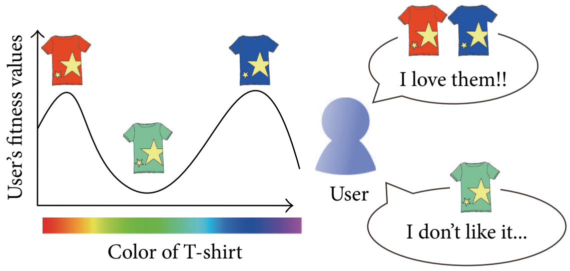


Figure 3: Landscape representation of multimodal Kansei.

this study, a landscape that uses the preferences of a user as evaluation values is defined as a Kansei landscape. In the domain of product recommendations, the preferences include the buying motivation, the tastes, and so on. Therefore, there may be multiple optimum solutions of different design variables. Moreover, a user often prefers those different solutions, especially in product recommendation. In this case, the Kansei landscape has multiple preference peaks whose heights are almost the same as shown in Figure 3. We define this as a multimodal Kansei landscape. We assume that if we concurrently search for multiple peaks, we will obtain the correct multimodal Kansei landscape and display solutions that meet the preferences of the user.

In a previous study, Ito et al. [25] attempted to extrapolate the locations of peaks by clustering solutions with high values in a specific generation. Although this clustering method was able to estimate the peaks, it could not search for the maximum values of the peaks. In this paper, we will expand this method and obtain maximum values of the peaks by extrapolating and searching every generation.

3. Proposed Method

To search for multiple optimum points on a multimodal Kansei landscape, we need to extrapolate the locations of the peaks and then search for the maximum values of each peak. We propose a new crossover method that consists of two steps. In the first step, we cluster solutions with high values. The location of each cluster is assumed to be a preference peak. In the second step, we generate new offsprings by using a multidimensional normal distribution constructed from members of each cluster. A detailed description of the two steps is presented below.

3.1. Clustering Method for Extrapolating Multiple Peaks. In the first step, we classify the parent solutions selected using the selection operator of the GA. Because we do not know to which peak a solution belongs, we use a clustering method that classifies solutions according to their pairwise distances in the design variable space. Figure 4 shows the example of clustering result obtained when applying our approach to T-shirts. Each data point represents a solution and each cluster is a candidate location of a peak.

The number of clusters is very important in this method, because it stands for the number of user's preference peaks. However, the number is unknown in advance. Therefore, we can use clustering methods that determine the optimum number of clusters [26, 27] automatically. Alternatively, we can use accuracy indices for clustering results, such as gap statistics [28, 29] and silhouette statistics [30, 31]. These indices can be used as follows. We recursively apply a clustering method such as K-medoids and K-means [32] by using a variable number of clusters. When the value of indices is maximum (or minimum), the number of clusters is regarded as optimum. In Section 4, we present an experiment in which we use K-medoids as the clustering method and silhouette statistics as the method of determining the number of clusters.

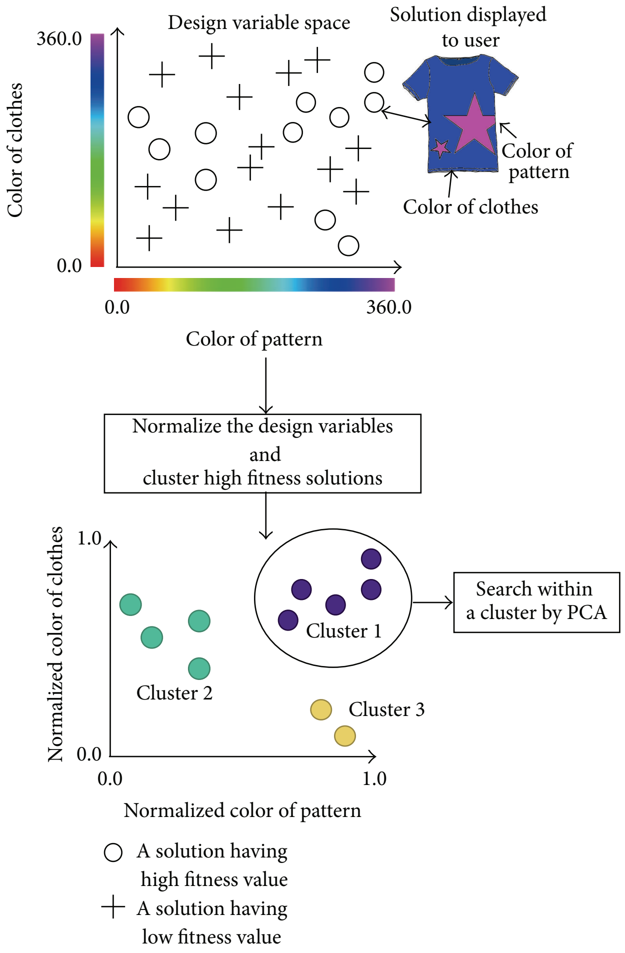

Design variable space
Solution displayed to user
360.0
Color of clothes
0.0
0.0
360.0
Color of pattern
Color of pattern
Color of clothes
Normalize the design variables and cluster high fitness solutions
1.0
Normalized color of clothes
0.0
0.0
1.0
Normalized color of pattern
Cluster 1
Cluster 2
Cluster 3
Search within a cluster by PCA
○ A solution having high fitness value
+ A solution having low fitness value

Figure 4: Example of results obtained when applying a clustering method to T-shirts solutions.

3.2. Searching for Peak Maximums Using Principal Component Analysis. To search for the maximum values of the extrapolated peaks, we construct a probabilistic model based on solutions and produce offspring from every cluster. In order to produce offsprings that maintain their correlations of solutions with the design variables, we apply the principal component analysis (PCA) [33] approach to each cluster [34, 35].

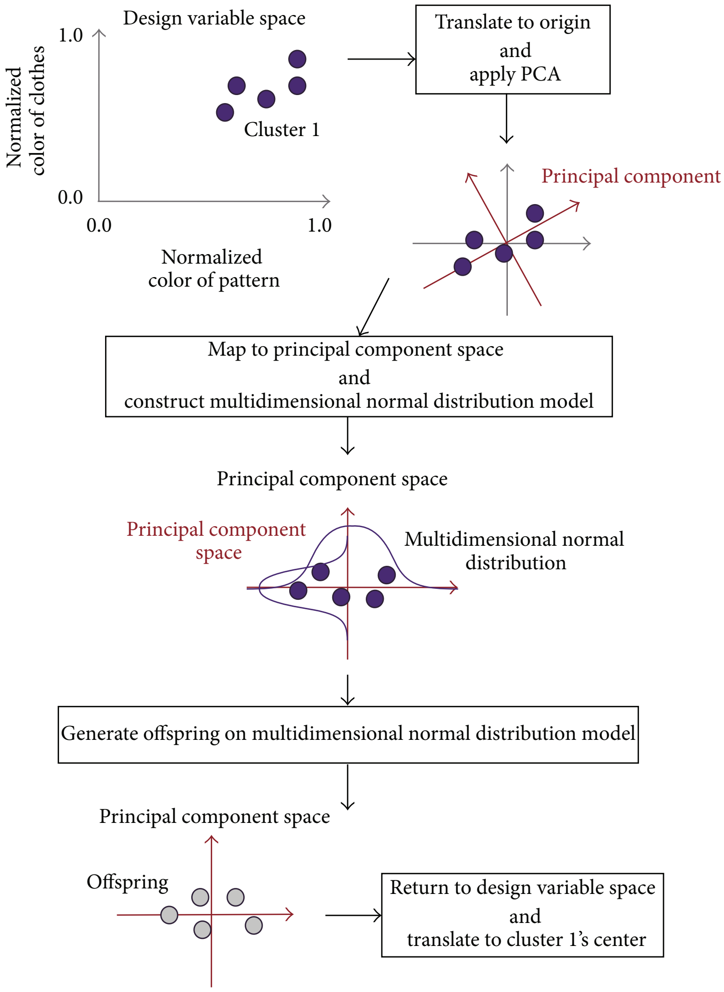

Design variable space
1.0
0.0
Normalized color of clothes
Cluster 1
0.0
1.0
Normalized color of pattern
Translate to origin and apply PCA
Principal component
Map to principal component space and construct multidimensional normal distribution model
Principal component space
Principal component space
Multidimensional normal distribution
Generate offspring on multidimensional normal distribution model
Principal component space
Offspring
Return to design variable space and translate to cluster 1's center

FIGURE 5: Procedure of offspring generation by principal component analysis.

The following shows the details of the procedure. At first, the number of offspring in each cluster must be set so that the total number of offspring is equal to the population size. We determine the number of offspring in each cluster on the basis of the ratio of the cluster member size to the sum. Subsequently, the following process is applied to each cluster. Figure 5 shows the procedure of generating offspring on multidimensional normal distribution constructed by principal component analysis.

(1) The solutions belonging to the cluster are represented by an $m \times n$ matrix X when the number of design variables is n and a cluster has m parents. Through the parallel translation of X, we obtain a matrix T whose entries are design variables with zero mean.

(2) We use T to compute a variance-covariance matrix S.

(3) We apply PCA and obtain the eigenvalues and eigenvectors of matrix S. The rotation matrix A is constructed by ranking the eigenvectors in a column according to the descending order of the absolute values of the eigenvalues.

(4) By multiplying the rotation matrix A and the translated parents matrix T, we map the parent solutions into a space whose dimensions are uncorrelated:

$$Y = TA. \tag{1}$$

(5) The multidimensional normal distribution is constructed from Y and we use this probabilistic distribution to generate the children solutions $Y_{\text{offspring}}$. However, if the raw distribution is used, the children solutions converge near the origin of the space. To resolve this problem, we multiply the variance of each dimension of the distribution by a parameter and generate offspring in a wider space. In the latter experiments, the parameter α is set to 1.4.

(6) By multiplying $Y_{\text{offspring}}$ by A^{-1}, the inverse matrix of A, $Y_{\text{offspring}}$ are mapped from the space with basis matrix A into the original space:

$$X_{\text{offspring}} = Y_{\text{offspring}} A^{-1}. \tag{2}$$

(7) The mean vector of X is added to each offspring $X_{\text{offspring}}$, which are included into the next population.

(8) We return to Step 1 until these operations are repeated for all clusters.

4. Pseudouser Experiment

Before an experiment with a real user, to confirm that our proposed method can search for multiple peaks and improve solutions within a peak, an experiment with a designed pseudo-user was performed. We developed an experimental system that simulates a product recommendation system and compared the effectiveness of the proposed and conventional crossover methods. The system consists of an optimization module and a pseudo-user module. A pseudo-user is a software having several pseudo-Kansei landscapes and evaluating solutions on behalf of a real user. The reason why we used a pseudo-user is that the user does not have the evaluation fluctuation that a human has. Therefore, this is appropriate to confirmation for algorithm behavior.

4.1. Experimental System. Figure 6 shows the flow of the experiment system. The conventional method we implemented was the blend crossover [24]. During a trial, we used one of the two available crossover methods, while the other operators (initialization, selection, and mutation) were the same. The details about the implementation of the evaluation module, selection module, proposed method, and conventional method are described below.

4.1.1. Evaluation Module. In this experiment, an evaluator is not a real user but a software. A software evaluator can make a large number of evaluations and the reproducibility of results is guaranteed. We call it a pseudo-user. The pseudo-user has some pseudo-Kansei landscapes for evaluation. It is thought that a Kansei landscape is not a sequence of rugged mountains but a superposition of gentle mountains which are almost as high. Therefore, a sum of Gaussian functions is appropriate to simulate a Kansei landscape.

We used 12 pseudo-Kansei landscapes. In detail, they are the combinations of 2, 4, and 6, dimensions and 2, 4, 6 and 8 peaks. Figure 7 shows the examples of 2-dimensional pseudo-Kansei landscapes with 2 or 4 peaks. The height of all Gaussian functions was set to 7. The variances need to differ in each landscape, to avoid combining some peaks into a single peak. They were determined to let 10% solutions of 1000 random samples exceed 5.5 evaluation values in a preliminary experiment.

Using these landscapes, the pseudo-user evaluated the top 6 solutions as good solutions.

4.1.2. Selection Module. Table 1 shows the shared parameters used. The population and generation sizes were limited to the sizes that did not burden a real user if the user operated this system [7].

The selection method adds the solutions evaluated as good into selection archive and extracts a required number of parent solutions from the archive. In this experiment, we set the selection size to 13, which is half the size of the population. In the 1st generation, random initialization is repeated until the size of the archive exceeds the selection size. From the 2nd generation onward, the selection method extracts solutions which are newer until the number of solutions is greater than the selection size.

Table 1: Experimental parameters.

Parameter	Value
Population size	25
Generation size	13
Selection size	13
Crossover rate	1.0
Mutation size	5
Mutation method	Uniform mutation

4.1.3. Proposed Method Module. In this experiment, we used the K-medoids clustering method and computed silhouette statistics to determine the number of clusters. The silhouette statistics provide an accuracy indicator for clustering results. For each data point i, we compute the mean of distances to all data points belonging to the same cluster $a(i)$ and nearest cluster $b(i)$. Next, we obtain the silhouette statistics of the clustering result by computing the mean of $s(i)$, which is defined as

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}. \tag{3}$$

The higher this value, the more precise the clustering results. In this experiment, the number of clusters used in the K-medoids method was varied from 2 to 8. The number of clusters that maximized the silhouette statistics was adopted in the trial.

4.1.4. Conventional Method Module. The conventional system uses the blend crossover operator to generate offsprings. Figure 8 illustrates the procedure of this method. Initially, two solutions are randomly extracted from parent solutions. Distances of the 2 solutions on each design variable are calculated. Next, the distances are expanded outward by α times and a super cuboid is constructed based on the distances. Finally, offspring are produced in the super cuboid randomly. In this experiment, α was set to 0.2.

4.2. Metrics. The effectiveness of the proposed method was evaluated from two standpoints.

(1) The proposed method can search for multiple peaks in a landscape.

(2) The proposed method can improve search within every peak.

The goal of the proposed method is to estimate locations of multiple peaks in a landscape. However, it is also necessary

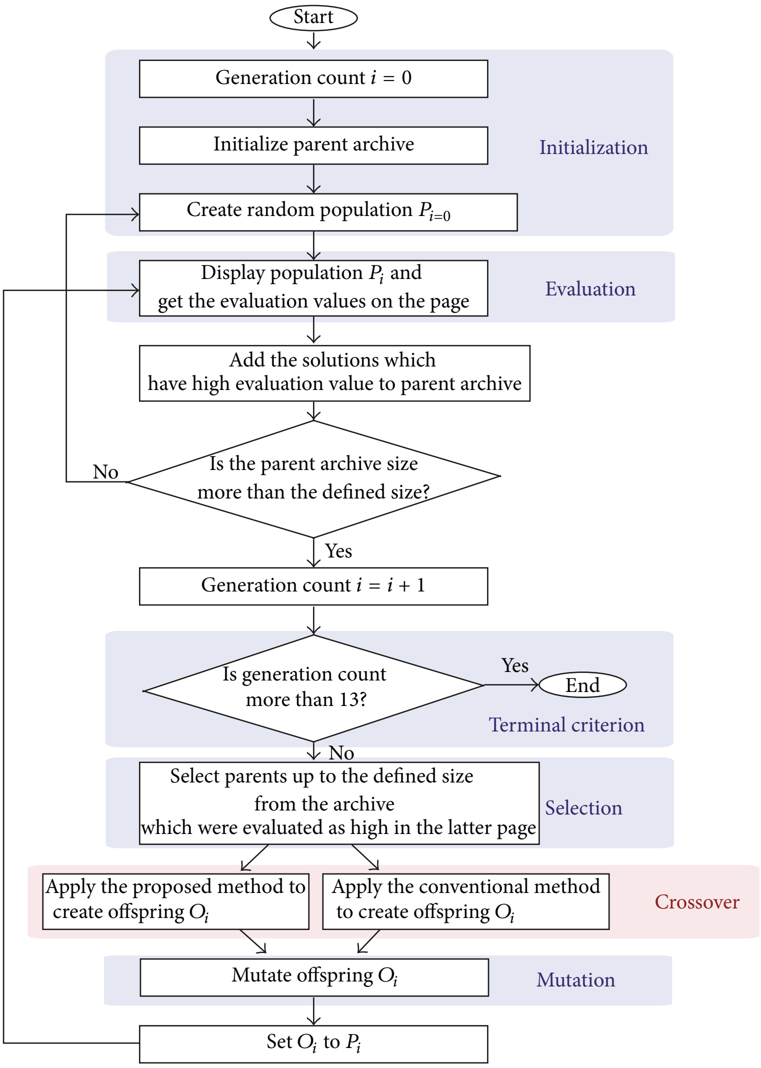
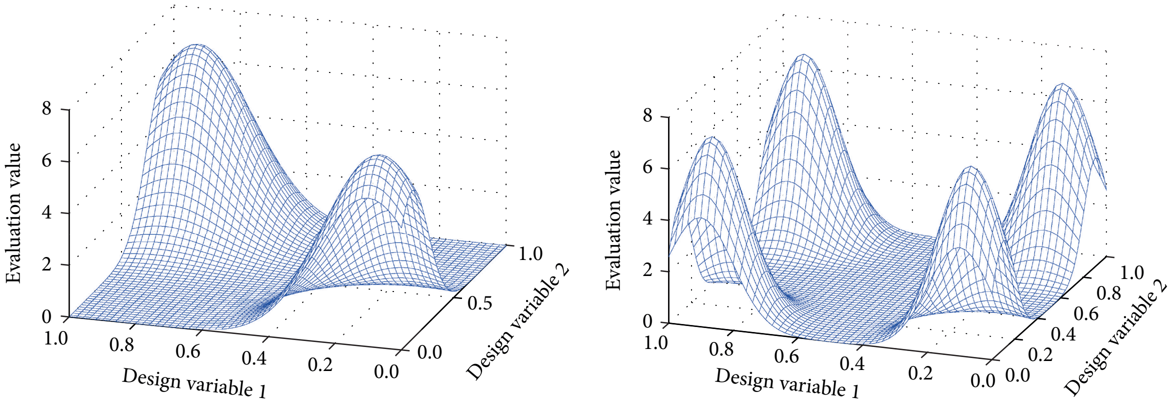

Start
Generation count $i = 0$
Initialize parent archive
Initialization
Create random population $P_{i=0}$
Display population P_i and get the evaluation values on the page
Evaluation
Add the solutions which have high evaluation value to parent archive
No
Is the parent archive size more than the defined size?
Yes
Generation count $i = i + 1$
Is generation count more than 13?
Yes
End
Terminal criterion
No
Select parents up to the defined size from the archive which were evaluated as high in the latter page
Selection
Apply the proposed method to create offspring O_i
Apply the conventional method to create offspring O_i
Crossover
Mutate offspring O_i
Mutation
Set O_i to P_i

Figure 6: Flowchart of experimental system.



(a) Pseudo-Kansei Landscape with 2 peaks

(b) Pseudo-Kansei Landscape with 4 peaks

Figure 7: Examples of 2-dimensional pseudo-user landscapes.

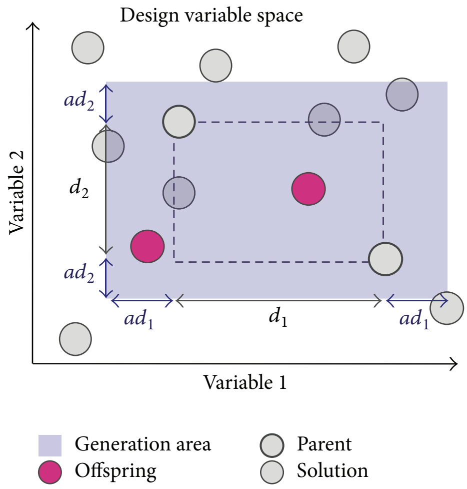


FIGURE 8: Process of generating offspring using the blend crossover operator.

to search for the maximum value of each peak. Therefore, we used the following two metrics to evaluate the method.

Variance. The sum of differences between the number of offspring generated in a peak and number of offspring that should be generated in all the peak.

Improvement. The mean of evaluation values of a population.

Variance is used to quantify the performance of evenly searching for all peaks. If the proposed method works effectively, it is hoped that each peak is searched by the same number of candidate solutions. This ideal number of offspring on each peak is computed by The following equation

$$O_{\text{ideal size}_i} = \frac{\text{peak area}_i}{\text{total peak area}} * \text{population size}. \quad (4)$$

The index of a peak in a Kansei landscape is denoted by i. A peak area stands for an area of a field of peak. Figure 9 shows the example of a field of peaks. Fields of peaks are extracted by a hypersurface. We define a field of peak as a continuous part of the landscape that is above the hypersurface. In this experiment, the height of the hypersurface is set to 5.5. This value was determined so that the total area of the field of peaks is approximately 10% of the entire surface area.

Because areas of all peaks in a pseudo-Kansei landscape are equal, all the ideal numbers of offspring $O_{\text{ideal size}}$ are approximately the same. The variance of the search results was evaluated by totaling the differences between the number of offspring generated and ideal number of offspring in each generation. This value is denoted as D_{ia} and is computed by the following equation, where the variable n in (5) represents the number of peaks:

$$D_{ia} = \sum_{i=1}^{n} \left| O_{\text{ideal size}_i} - O_{\text{actual size}_i} \right| \quad (5)$$

The performance of the search was confirmed by evaluating the value of *Improvement* for the populations generated.

4.3. Results. Figure 10 shows the results of *Variance*. Each graph shows the generational change of D_{ia} in each pseudo-Kansei landscape. The columns represent the dimension and the rows represent the number of peaks. If D_{ia} is smaller, the method is superior in *Variance*. The proposed method (solid blue line) is statistically better. However, D_{ia} tended to rise in the latter half of the generations in many cases.

Figure 11 shows the results of *Improvement*. The placement of graphs in Figure 10 is analogous to that in Figure 11. Because *Improvement* represents the mean of the evaluation values, the higher-quality results have larger values. The generational means of the proposed method were statistically better than those of the conventional method.

4.4. Discussion. In *Variance* evaluations, D_{ia} almost increased in the latter half. This was caused by pseudo-user's evaluation. The pseudo-user evaluated the static number of solutions as good in descending order of evaluation values. If the number of solutions belonging to a cluster is higher than the others, more offspring are produced in the cluster. Therefore, the search area tended to converge to one finally. From a practical perspective, real users do not have this tendency because they do not evaluate strictly based on the descending order of the evaluation values. Moreover, if users hope for the diverse view, they are expected to evaluate highly the solutions belonging to the multiple peaks. Therefore, this is not a problem.

The same tendency was observed in *Improvement*. In the latter half of the generations, the results of the conventional method were similar to those of the proposed method. This behavior is expected because the blend crossover method results in the strong convergence of solutions into one peak. However, the proposed method progressed faster and searched for multiple peaks.

Summarizing, the proposed method is appropriate, especially in the early generations.

5. Subjective Experiment

In order to verify effectiveness of the proposed method in a real user, we performed a subjective experiment. This experiment used the same experiment system presented in Section 4, with the exception of the evaluation phase. The evaluator is not a pseudo-user but a real user. Eight males and 4 females in their 20s participated in this experiment. We created three applications to be optimized and an evaluation interface for the experimental participants. In addition, while the Kansei landscape of a pseudo-user is known, the Kansei landscape of a human is not. We performed a preliminary experiment to obtain the approximate landscapes of the subjects participating in the experiment.

First, we describe the applications developed and the preliminary experiment performed to extract the approximate landscapes of the participants. The approximate landscapes obtained also verify that the participants had multiple

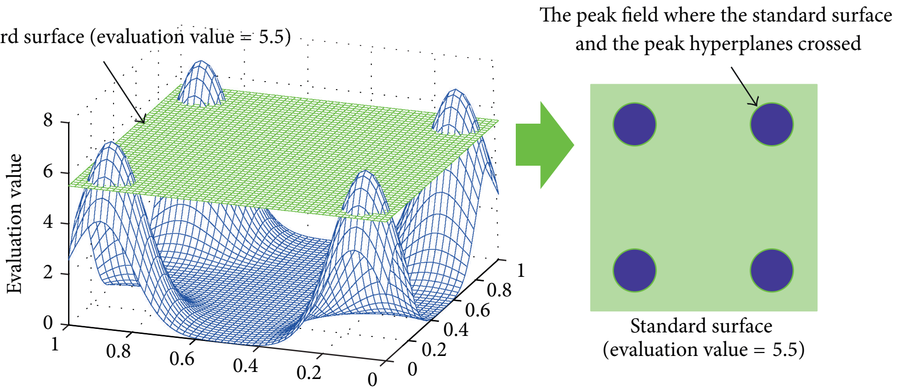

Standard surface (evaluation value = 5.5)
Evaluation value
8
6
4
2
0
1
0.8
0.6
0.4
0.2
0
1
0.8
0.6
0.4
0.2
0
The peak field where the standard surface and the peak hyperplanes crossed
Standard surface (evaluation value = 5.5)

FIGURE 9: Definition of peak field.

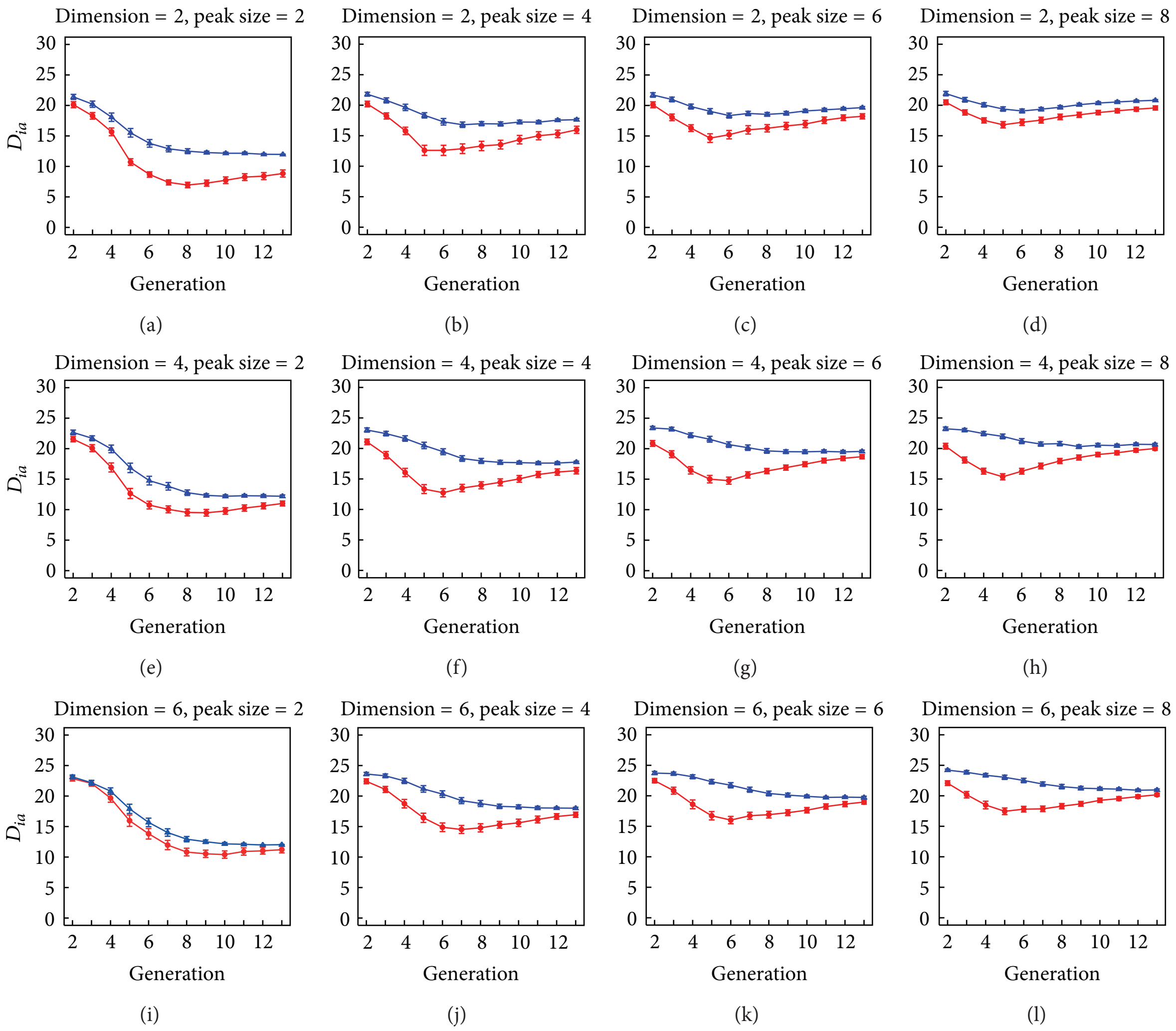
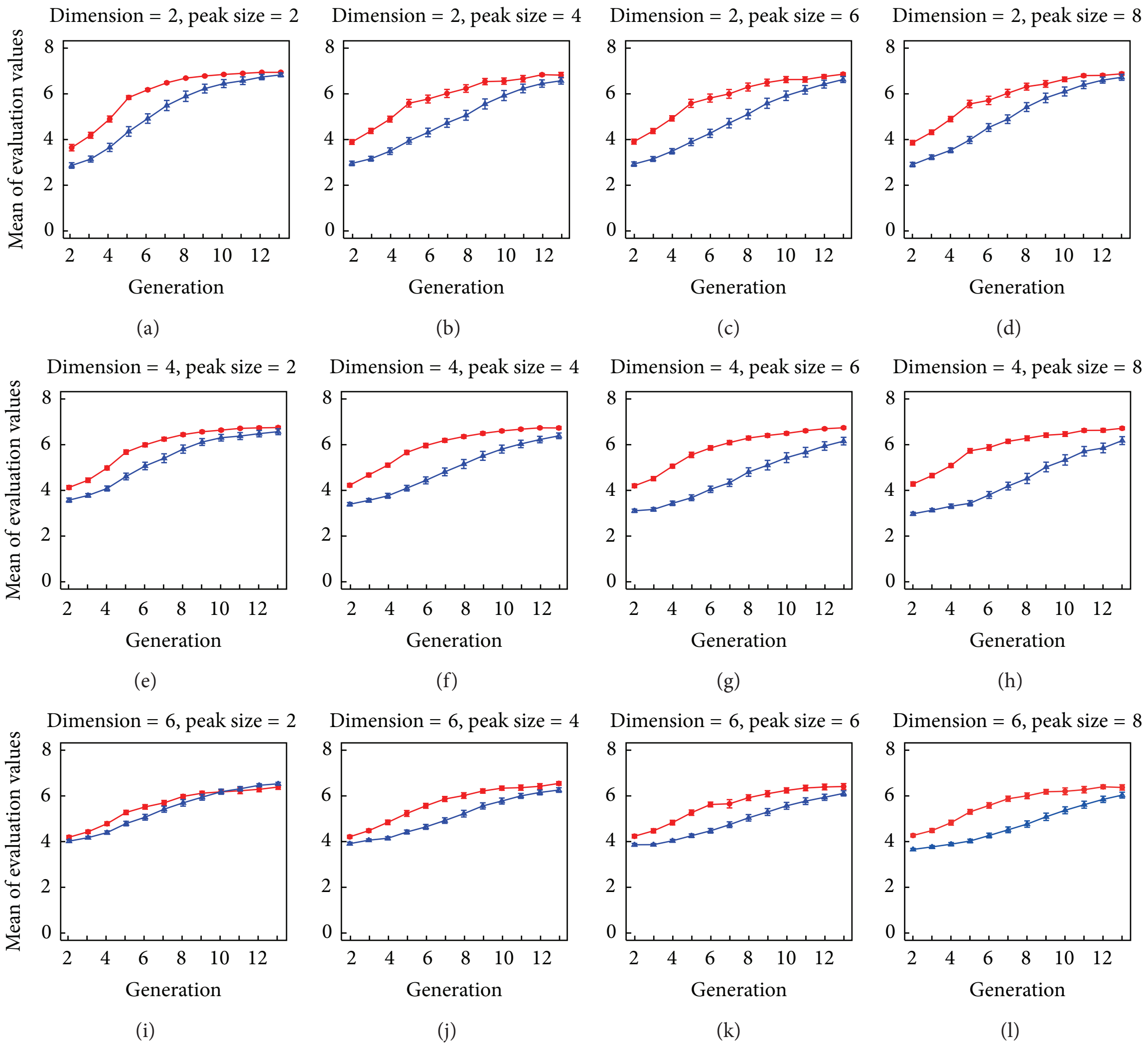


FIGURE 10: Value of D_{ia} in each generation. The horizontal axis represents the generations and the vertical axis is the mean of D_{ia} after 100 trials. The bar represents the standard error. The results of the proposed method are depicted by the solid red lines and those of the conventional method by the dotted blue lines. The graphs in the same row correspond to landscapes of equal dimensions and those in the same column correspond to landscapes with the same number of peaks.



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 11: Mean of the evaluation values of population in each generation. The horizontal axis represents the generations and the vertical axis is the mean of the evaluation values of population after 100 trials. The bar represents the standard error. The results of the proposed method are depicted by the solid red lines and those of the conventional method by the dotted blue lines. The graphs in the same row correspond to landscapes of equal dimensions and those in the same column correspond to landscapes with the same number of peaks.

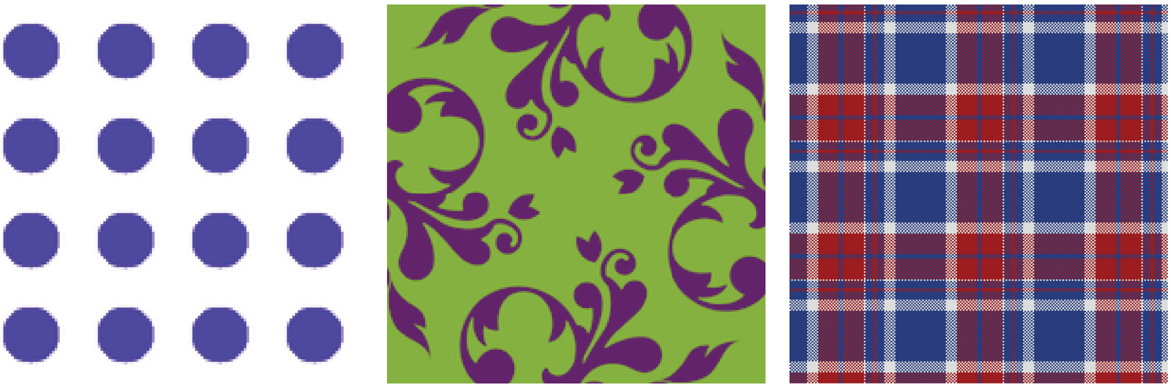
preferences. Next, we describe the iGA experimental system used in the subjective experiment. The results prove that the proposed method is effective on humans.

5.1. Applications. The experimental application for preference evaluation is a design of furniture's pattern. Especially, we used three patterns: dotted, arabesque, and plaid. Each pattern is treated as a single application. Additionally, when experimental participants operated one of the applications, they selected which furniture they design from curtain, sofa cover, and bed cover. We used this application domain because we believed that the evaluations would not fluctuate.

The examples of solutions are shown in Figure 12 and the design variables are shown in Table 2. The number of design variables of all the applications was 2. In the application of dotted pattern, we selected the design variables of color and size of dots. In the arabesque and plaid patterns, we selected two colors as design variables. For the plaid pattern, the two colors except white were parameterized using the hue, saturation, and brightness (HSB) color space [36]. The saturation and brightness values were kept constant and only hue values [0.0, 360.0) were parameterized. Because the hue values are expressed in degrees, the maximum 360.0 and minimum 0.0 values of hue are the same. In the dotted pattern, we varied the ratio of the radius of the dots over the distance between the neighboring dots [0.0, 1.0]. The number of dots was kept constant.

5.2. Preliminary Experiment

5.2.1. Experimental Procedure. Before an iGA experiment, to obtain the indicators required to evaluate the proposed method, we extracted the approximate landscapes of the participants. Landscapes were approximated by sampling the



(a) Dotted

(b) Arabesque

(c) Plaid

Figure 12: Examples of solutions displayed to users.

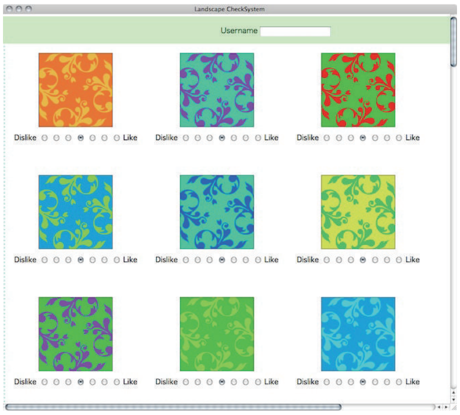

Landscape CheckSystem
Username
Dislike ○ ○ ○ ◉ ○ ○ ○ Like
Dislike ○ ○ ○ ◉ ○ ○ ○ Like
Dislike ○ ○ ○ ◉ ○ ○ ○ Like
Dislike ○ ○ ○ ◉ ○ ○ ○ Like
Dislike ○ ○ ○ ◉ ○ ○ ○ Like
Dislike ○ ○ ○ ◉ ○ ○ ○ Like
Dislike ○ ○ ○ ◉ ○ ○ ○ Like
Dislike ○ ○ ○ ◉ ○ ○ ○ Like
Dislike ○ ○ ○ ◉ ○ ○ ○ Like

Figure 13: Interface used in the experiment conducted to approximate the landscapes of participants.

Table 2: Design variables of experimental applications.

Application name	Design variable 1	Design variable 2
Dotted	Hue of dot	Rate of dot size
Arabesque	Hue of front color	Hue of background color
Plaid	Hue of first color	Hue of second color

design variable space of the applications mentioned above. Moreover, by verifying that a user has multiple preferences, we confirmed the need for our proposed method.

Each design variable was divided evenly by a grid and the grid points were treated as sample points. The participants rated solutions corresponding to the sample points. Because the grid size was 10 × 10, we evaluated 100 solutions for each application. We estimated the approximate landscapes by linearly interpolating the evaluation values. Prior to the experiment, the participants were instructed to consider buying a fabric pattern for a room being renovated and to evaluate how much they liked each fabric. The order of trials of the three applications was counterbalanced among participants.

5.2.2. Experimental System. Figure 13 shows the evaluation interface used in the experiment. The order of the 100 solutions displayed was randomized. Under each solution presented, there were seven radio buttons. The participants evaluated how much they liked each solution by selecting the appropriate button.

5.2.3. Results. In Figure 14, we present the examples of approximate landscapes obtained. Figure 14(a) shows the example of a multiple-peak landscape and Figure 14(b) shows the example of a single-peak landscape. In these figures, the axes are the design variables of the applications. The areas evaluated as highly preferred are colored red and those as least preferred are colored yellow.

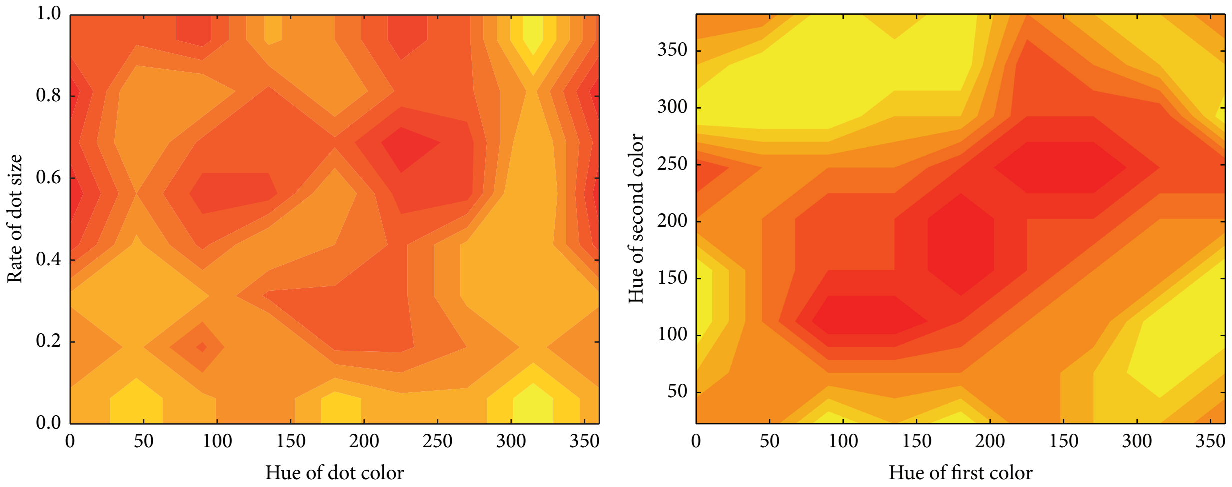
The 12 participants evaluated the 3 applications. Thus, 36 Kansei landscapes were obtained in total.

5.2.4. Discussion. We examined if the participants had multiple preferences according to the following process.

First, in order to count the number of peaks, we determined the field of peaks for each approximated Kansei landscape. Specifically, for every landscape, we plotted a histogram of the evaluation values for the solutions. Next, we set a threshold to select the top 25% of the evaluation values. A continuous area which has higher evaluation values than the threshold value was defined as a field of peak. Figure 15 presents the landscape image created by coloring all areas white in Figure 14(a) except for the peaks.

Next, we counted the peaks of each landscape. Figure 16 shows the frequency graph of the number of peaks. The horizontal axis is the number of peaks and the vertical axis is the number of landscapes that have a particular number of peaks. We confirmed that 27 out of the 36 landscapes had multiple peaks. Therefore, because the Kansei landscapes have multiple peaks in numerous instances, we require a new method that searches for them.

5.3. Experiment to Verify Effectiveness of the Proposed Method. In this section, we verify whether our proposed method



(a) Approximated multimodal landscape (for dotted pattern)

(b) Approximated unimodal landscape (for plaid pattern)

Figure 14: Examples of approximated Kansei landscapes.

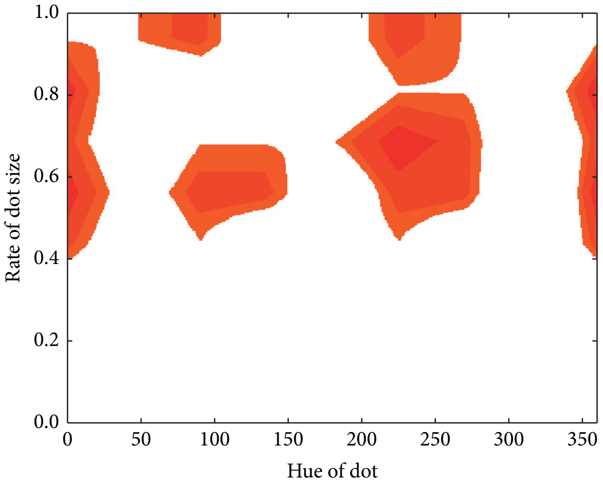


Figure 15: Peaks in the multimodal landscape shown in Figure 14(a).

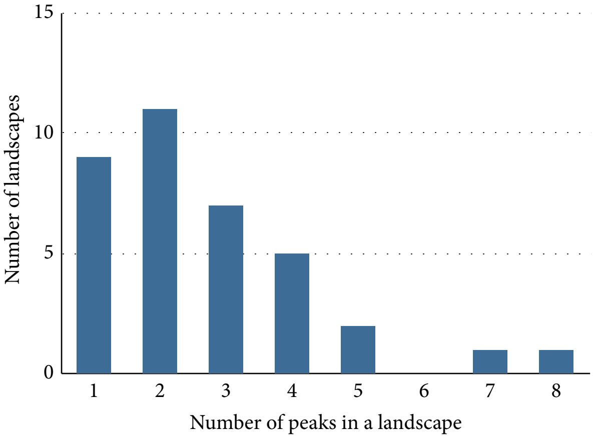


Figure 16: Frequency graph of the numbers of landscapes for all number of peaks.

searches for solutions in human multimodal Kansei landscapes more effectively than the conventional method. We used the same indicators as in the pseudo-user experiment to evaluate the performance of the methods.

5.3.1. Experimental Procedure. The participants operated the experimental systems, which applied one of the two available methods for each application. In total, six trials were performed. The order of trials was counterbalanced as described below. The participants were divided into 2 groups. The members of each group first operated the system implementing the proposed method module and then the system implementing the conventional method module. Moreover, within each group, the order of applications was randomized.

The participants were instructed to evaluate fabric patterns of the same furniture as the ones used in the preliminary experiment.

5.3.2. Experimental System. In this experiment, we used the same system as the one used in the pseudo-user experiment. And, we used the same experimental parameters as those in the pseudo-user experiment because the results presented in Section 4.3 show that the search performance of the proposed method in 2-dimensional landscapes was superior enough to that of the conventional method.

We replaced the pseudo-user module by the experiment interface shown in Figure 17, which was operated by the participants. The participants rated the 25 solutions displayed by clicking on their favorites. When they clicked on an image, the color of the frame changed to red. After rating all solutions, the participants clicked the Next Page button located at the bottom of the interface and 25 new solutions were displayed. Each sequence of 25 solutions is called a page. The participants continued evaluating the solutions until a pop-up window was displayed notifying them that the trial had ended.

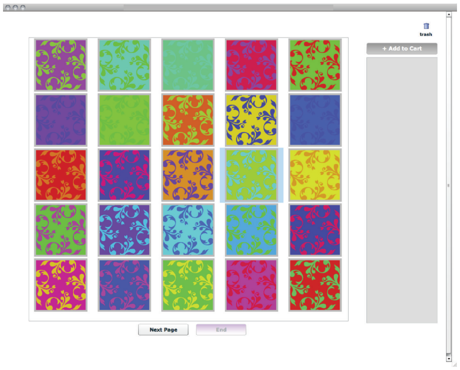


Figure 17: Interface used by participants during the experiment to rate solutions.

5.3.3. Results. Next, we present the results of the experiment with respect to the 2 indicators defined in Section 4.2.

First, we examine the metric of *Variance*, which is used to determine whether the proposed method can search for multiple peaks in a Kansei landscape. In Figure 18, we show the mean of D_{ia} for the landscapes with the same number of peaks. The procedure used to calculate the ideal number of generated offspring is the same as in Section 4.2. For each approximate landscape, we obtained the numbers of ideal generated offspring according to the fields of peaks. In Figure 18, the horizontal axis represents generations and the vertical axis is the mean of D_{ia}. We do not present the results of landscapes containing a single peak or more than 5 peaks because, for single-peak landscapes, D_{ia} is not meaningful, whereas, for landscapes with more than 5 peaks, we did not have a sufficient number of samples as Figure 16 shows.

The results for the landscapes with 2 and 3 peaks show that the proposed method obtained smaller D_{ia} values, which indicated that the method searched for more peaks. As mentioned in Section 5.2.4, the landscapes with 2 or 3 peaks are approximately half the landscapes. Hence the proposed method is efficient for approximately half the cases considered. In contrast, the conventional method was superior for the landscapes with 4 peaks. The reason for this behavior is discussed later.

Second, we consider the metric of *Improvement*, which is used to determine whether the proposed method searched for the maximum values of the peaks. In Figure 19, we show the mean of the evaluation values in each generation for cases where the landscapes either have a single peak or multiple peaks. The evaluation values of the solutions were estimated from the approximate landscapes. The horizontal axis represents the number of generations and the vertical axis is the mean of the evaluation values.

In Figure 19(a), the search results from the unimodal landscapes show that the conventional method achieved higher mean values. In contrast, in Figure 19(b), the search results from the multimodal landscapes show that the proposed and conventional methods achieve approximately the same evaluation values. These results indicate that the conventional method correctly searches the unimodal landscapes. Moreover, when a user had multimodal preferences, both methods achieve nearly equal performance.

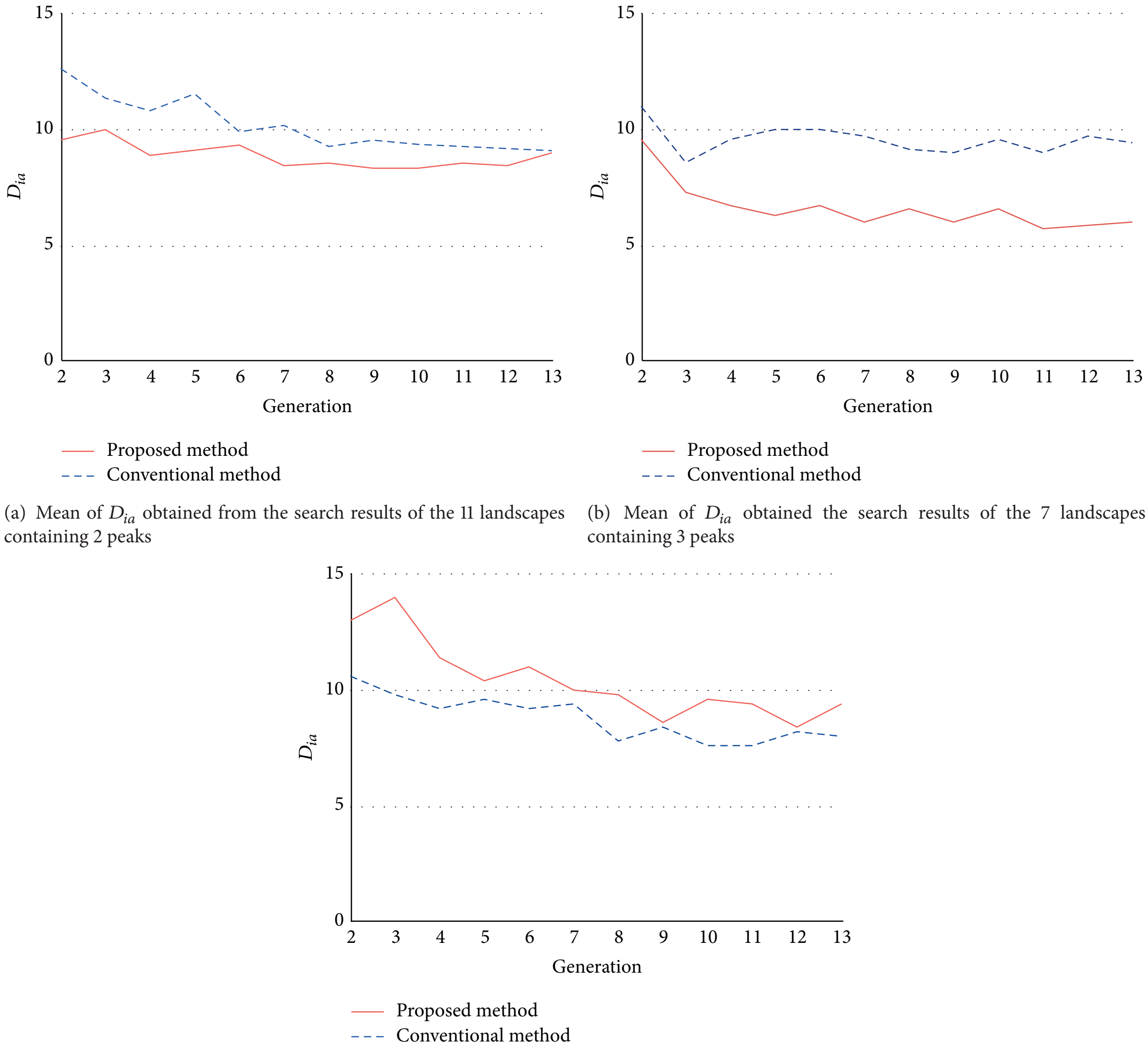
5.3.4. Discussion. For landscapes containing 4 peaks, the conventional method achieved higher *Variance* values. This behavior is attributed to the complexity of the shape of the landscapes caused by the presence of multiple peaks. Figure 20 shows the search logs for a 4-peak landscape. The axes of Figure 20 represent the design variables of the arabesque pattern. In this figure, we present only the field of peaks of the approximate landscape. The points plotted represent solutions that were selected as parents. In particular, in Figure 20(a), which shows the search logs of the proposed method, the points plotted with the same style correspond to solutions belonging to the same cluster.

Although the central peak in Figure 20 appears to consist of two peaks, it was actually treated as a single peak and the area is large. Therefore, the number of ideal offspring also became large. The conventional method used multiple solutions and searched the right half of this peak. The difference between the number of offspring of the conventional method and the ideal number of offspring was small.

In contrast, the proposed method considered the central peak as two individual peaks and searched them using two clusters. Therefore, the number of solutions which belonged to one of the clusters was small, which negatively affected the performance of the search. As a result, the difference between the number of offspring of the proposed method and the ideal number of offspring was large and the search performance of the proposed method looked like less than that of the conventional method. However, this behavior followed the purpose of the proposed method of finding user's multiple references by searching for multiple peaks. In fact, the proposed method was equal to finding the 2 areas which connected on the edge, and we thought that it was no problem.

Next, we examine the results that show the evaluation values of the conventional method to increase more rapidly. Figures 21 and 22 show examples of the search log for multimodal and unimodal landscapes. The blue circle roughly indicated the scope of search of each method. For the multimodal landscape, the proposed method used different clusters and searched the two peaks separately and efficiently.

On the other hand, in Figure 22, the conventional method tended to converge on solutions. However, the proposed method used too many clusters to search. This tendency is also observed in the discussion of *Variance*. Although it is in agreement with the goal of the proposed method, the search scope based on poor members of each cluster was narrow and the search performance became limited. The problem was discussed in the pseudo-user experiment (Section 4.4). To resolve this problem, it is necessary to either replace the crossover method with the conventional method or set the number of clusters to be smaller than that determined on the basis of silhouette statistics when finding that a user has a unimodal Kansei landscape.



(a) Mean of D_{ia} obtained from the search results of the 11 landscapes containing 2 peaks

(b) Mean of D_{ia} obtained the search results of the 7 landscapes containing 3 peaks

(c) Mean of D_{ia} obtained from the search results of the 5 landscapes containing 4 peaks

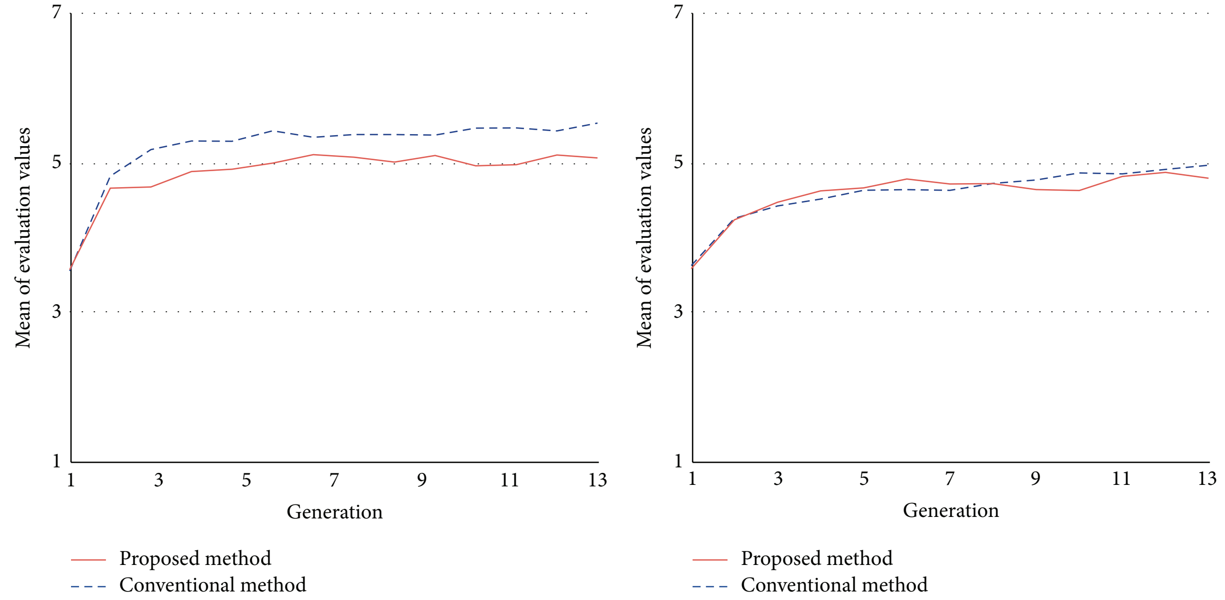
Figure 18: Value of D_{ia} in each generation. The results of the proposed method are depicted by the solid red line and those of the conventional method by the dotted blue lines. The number of landscapes is the number of trials by each method.

6. Conclusion

In this study, we developed a search method for multimodal preferences. Specifically, we applied iGA using Kansei landscapes and obtained optimal solutions in product recommendation. To obtain the maximum values of the peaks of Kansei landscapes, it is necessary to estimate the locations of the preference peaks and search within each peak efficiently. Therefore, we proposed a novel crossover method. Our method extrapolates the locations of the peak by clustering solutions that have high evaluation values and generates offspring by constructing a multidimensional normal distribution that accounts for the correlation among design variables.

We confirmed the efficiency of the proposed method by conducting a pseudo-user experiment and subject experiment. In these experiments, we compared the proposed method with a conventional method on the basis of two metrics. The first metric, *Variance*, is obtained by computing the difference between the real number of offspring generated in each peak and the ideal number of offspring. Using this metric, we examined whether our proposed method could identify more peaks. To verify the accuracy of the search within each peak, we defined the second metric *Improvement*, which measured the increase of the evaluation values.

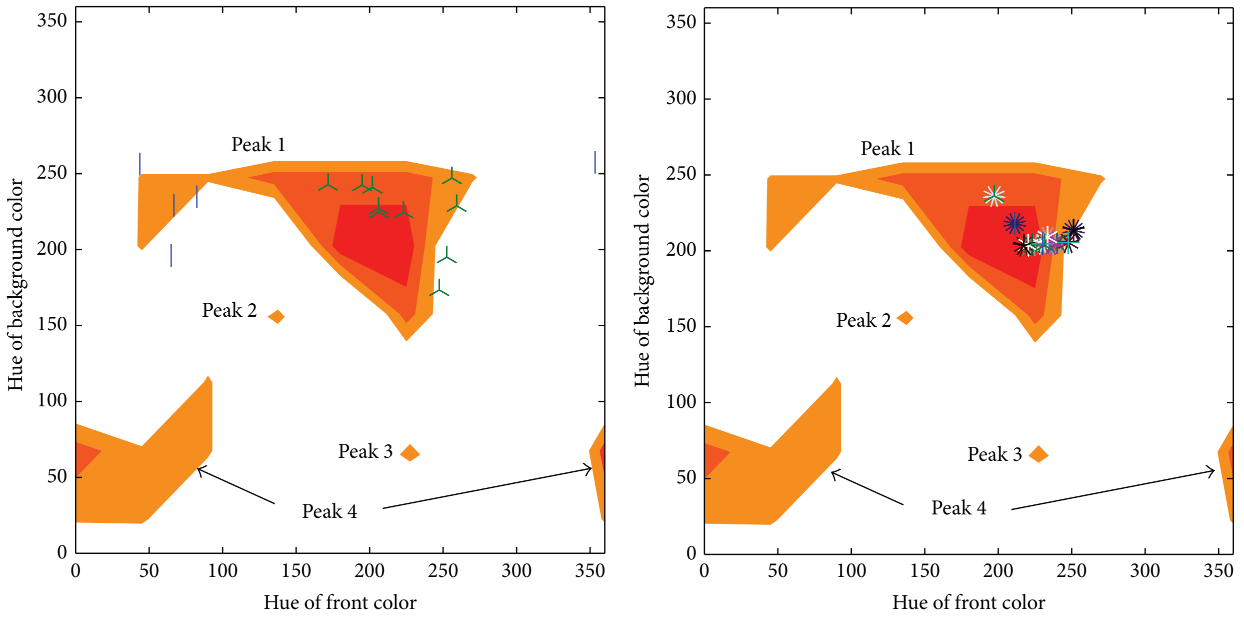
In the pseudo-user experiment, we investigated the behavior of the proposed method by varying the number of dimensions and peaks. For lower dimensions, the proposed



(a) Mean of the evaluation values estimated from the search results obtained from the 9 unimodal landscapes

(b) Mean of the evaluation values estimated from the search results obtained from the 27 multimodal landscapes

Figure 19: Mean of the evaluation values of population in each generation. The results of the proposed method are depicted by the solid red lines and those of the conventional method by the dotted blue lines. The number of landscapes is the number of trials by each method and means the number of samples.

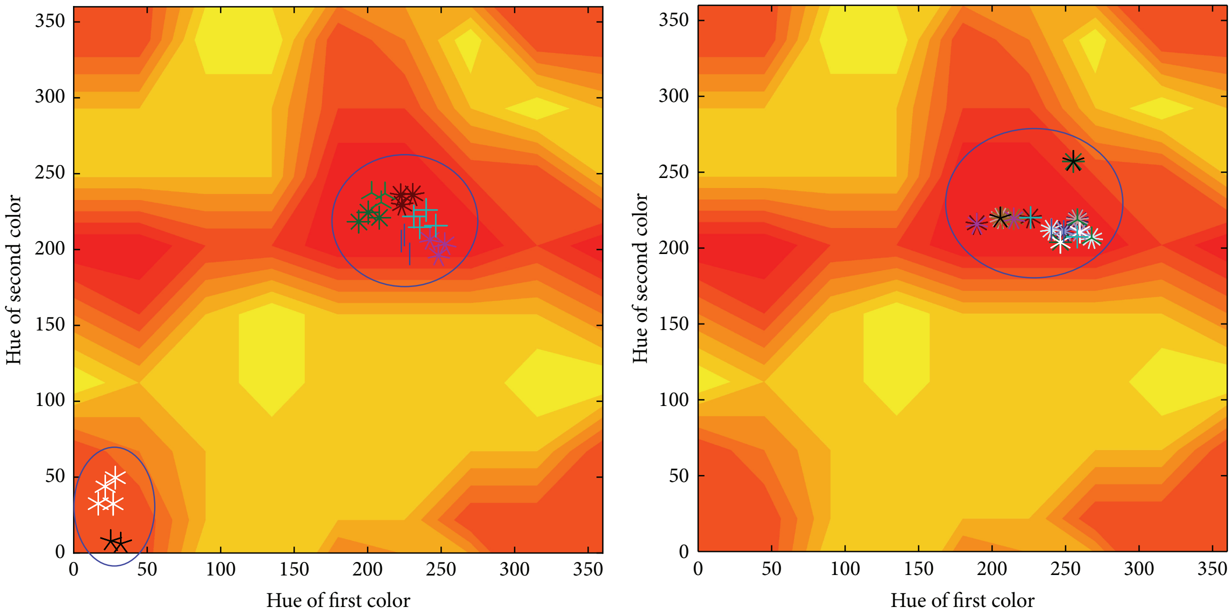


(a) Search result obtained using the proposed method

(b) Search result obtained using the conventional method

Figure 20: Examples of search results for landscapes containing 4 peaks (for arabesque pattern).

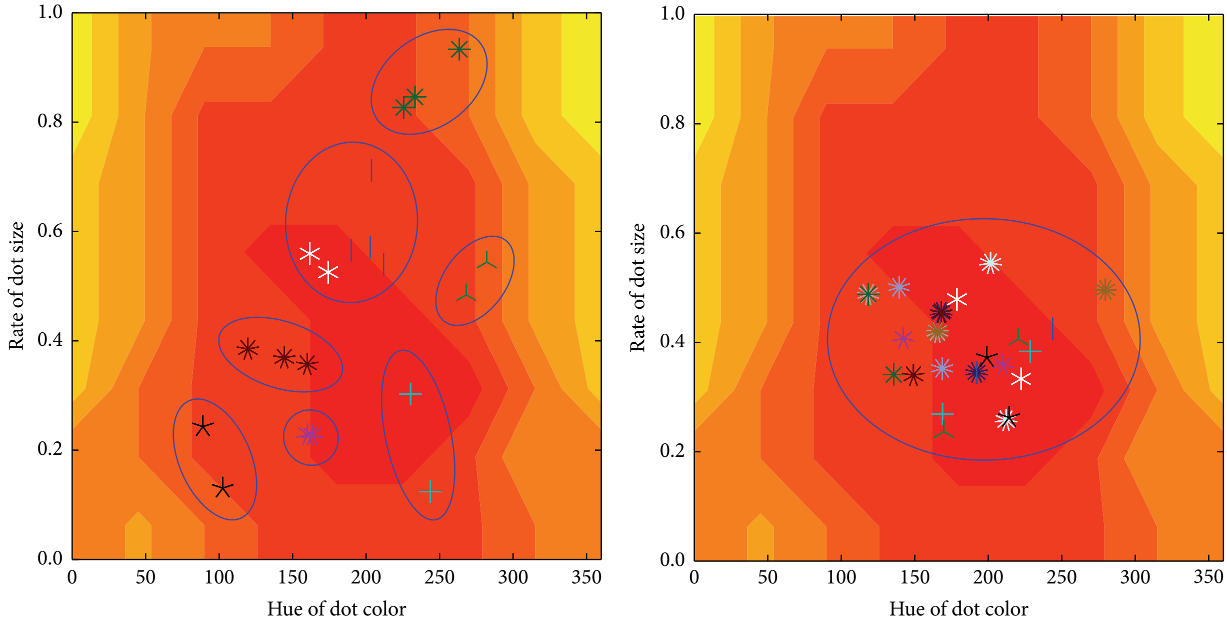
method performed better than the conventional method statistically on both metrics. In the subject experiment, the proposed method had higher *Variance* and the *Improvement* of the two methods was approximately the same when searching multimodal landscapes. Therefore, it was proved that the proposed method performed better when searching for multiple peaks. Because multimodal landscapes constituted 75% of all landscapes, we could say that the proposed method is appropriate for product recommendations which almost have multimodal preference.



(a) Search log of proposed method

(b) Search log of conventional method

Figure 21: Search logs of the last generation for the multimodal landscape (for the plaid pattern).



(a) Search log of proposed method

(b) Search log of conventional method

Figure 22: Search logs of the last generation for the unimodal landscape (for the dotted pattern).

In our future work, we will examine constraining the number of cluster members to improve the search performance in high dimensional spaces. Moreover, in unimodal landscapes, the proposed method tended to divide the solutions into too many clusters. To resolve this problem, we will consider switching to the conventional method in unimodal landscapes or controlling the number of clusters.

References

[1] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, *Recommender Systems Handbook*, Springer, 2011.

[2] M. J. Pazzani and D. Billsus, "Content-Based Recommendation Systems," in *The Adaptive Web*, pp. 325–341, 2007.

[3] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. 2009, Article ID 421425, 19 pages, 2009.

[4] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.

[5] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295, 2001.

[6] M. Tanaka, T. Hiroyasu, M. Miki, Y. Sasaki, M. Yoshimi, and H. Yokouchi, "Extraction and usage of Kansei meta-data in interactive Genetic Algorithm," in *Proceedings of 9th World Congress on Structural and Multidisciplinary Optimization (WCSMO9 '11)*, p. 505, 2011.

[7] H. Takagi, "Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation," *Proceedings of the IEEE*, vol. 89, no. 9, pp. 1275–1296, 2001.

[8] M. Herdy, "Evolution strategies with subjective selection," in *Parallel Problem SolvIng from Nature—PPSN IV*, H. M. Voigt, W. Ebeling, I. Rechenberg, and H. P. Schwefel, Eds., vol. 1141 of *Lecture Notes in Computer Science*, pp. 22–31, Springer, Berlin, Germany, 1996.

[9] T. Unemi, "SBArt4—Breeding abstract animations in realtime," in *Proceedings of the IEEE Congress on Evolutionary Computation (WCCI-IEEE CEC '10).*, IEEE, Barcelona, Spain, July 2010.

[10] K. Sims, "Artificial evolution for computer graphics," *Computer Graphics—ACM SIGGRAPH*, vol. 25, no. 4, pp. 319–328, 1991.

[11] J. Koza and R. Poli, "Genetic programming," in *Search Methodologies*, E. K. Burke and G. Kendall, Eds., pp. 127–164, Springer, New York, NY, USA, 2005.

[12] D.-W. Gong, J. Yuan, and X.-Y. Sun, "Interactive genetic algorithms with individual's fuzzy fitness," *Computers in Human Behavior*, vol. 27, no. 5, pp. 1482–1492, 2011.

[13] D. Gong, "Interactive genetic algorithms with individual fitness not assigned by human," *Journal of Universal Computer Science*, vol. 15, no. 13, pp. 2464–2480, 2009.

[14] X. Sun, D. Gong, and W. Zhang, "Interactive genetic algorithms with large population and semi-supervised learning," *Applied Soft Computing*, vol. 12, no. 9, pp. 3004–3013, 2012.

[15] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Professional, 1989.

[16] H.-S. Kim and S.-B. Cho, "Application of interactive genetic algorithm to fashion design," *Engineering Applications of Artificial Intelligence*, vol. 13, no. 6, pp. 635–644, 2000.

[17] S.-B. Cho, "Towards creative evolutionary systems with interactive genetic algorithm," *Applied Intelligence*, vol. 16, no. 2, pp. 129–138, 2002.

[18] J. C. Quiroz, S. J. Louis, A. Shankar, and S. M. Dascalu, "Interactive genetic algorithms for user interface design," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '07)*, pp. 1366–1373, September 2007.

[19] J. C. Quiroz, A. Banerjee, S. J. Louis, and S. M. Dascalu, "Document design with interactive evolution," in *New Directions in Intelligent Interactive Multimedia Systems and Services*, vol. 2, pp. 309–310, Springer, Berlin, Germany, 2009.

[20] P. Legrand, C. Bourgeois-Republique, V. Péan et al., "Interactive evolution for cochlear implants fitting," *Genetic Programming and Evolvable Machines*, vol. 8, no. 4, pp. 319–354, 2007.

[21] H. Takagi and M. Ohsaki, "Interactive evolutionary computation-based hearing aid fitting," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 3, pp. 414–427, 2007.

[22] F. Herrera, M. Lozano, and A. M. Sánchez, "Hybrid crossover operators for real-coded genetic algorithms: an experimental study," *Soft Computing*, vol. 9, no. 4, pp. 280–298, 2005.

[23] K. Deep and M. Thakur, "A new crossover operator for real coded genetic algorithms," *Applied Mathematics and Computation*, vol. 188, no. 1, pp. 895–911, 2007.

[24] L. J. Eshelman and J. D. Schaffer, "Real-coded genetic algorithms and interval-schemata," *Foundations of Genetic Algorithms*, vol. 2, pp. 187–202, 1993.

[25] F. Ito, T. Hiroyasu, M. Miki, and H. Yokouchi, "Discussion of offspring generation method for interactive genetic algorithms with consideration of multimodal preference," in *Proceedings of the 7th International Conference on Simulated Evolution and Learning*, vol. 5361, pp. 349–359, 2008.

[26] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, Article ID 026113, pp. 1–15, 2004.

[27] I. Derényi, G. Palla, and T. Vicsek, "Clique percolation in random networks," *Physical Review Letters*, vol. 94, no. 16, Article ID 160202, 2005.

[28] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society B*, vol. 63, no. 2, pp. 411–423, 2001.

[29] J. Handl and J. Knowles, "Multiobjective clustering with automatic determination of the number of clusters," Tech. Rep. COMPSYSBIO-2004-02, UMIST, Manchester, UK, 2004.

[30] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[31] A. Hotho, A. Maedche, and S. Staab, "Ontology-based text document clustering," *Künstliche Intelligenz*, vol. 4, pp. 1–13, 2002.

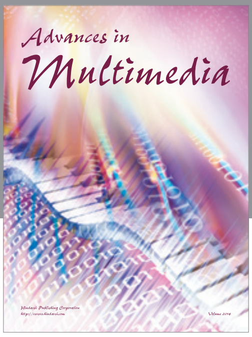
[32] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithms: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[33] I. T. Jolliffe, *Principal Component Analysis*, Springer, 2nd edition, 2002.

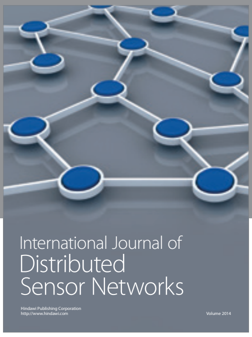
[34] T. Hiroyasu, M. Miki, M. Sano, H. Shimosaka, S. Tsutsui, and J. Dongarra, "Distributed probabilistic model-building genetic algorithm," in *Genetic and Evolutionary Computation—GECCO*, vol. 2723 of *Lecture Notes in Computer Science*, pp. 1015–1028, Springer, 2003.

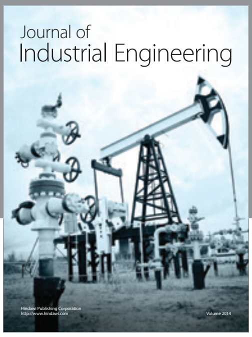
[35] M. Takahashi and H. Kita, "A crossover operator using independent component analysis for real-coded genetic algorithms," in *Proceedings of the Congress on Evolutionary Computation*, vol. 1, pp. 643–649, May 2001.

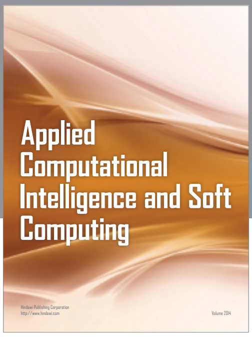
[36] M. D. Fairchild, *Color Appearance Models*, Wiley-IS & T, Chichester, UK, 2nd edition, 2005.

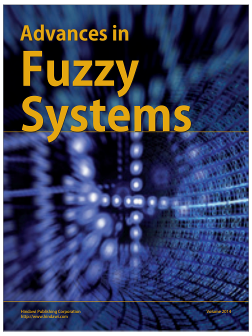

Advances in Multimedia

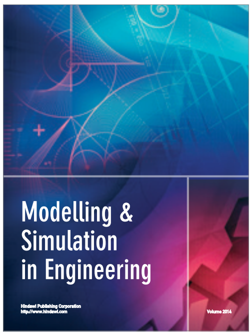
The Scientific World Journal

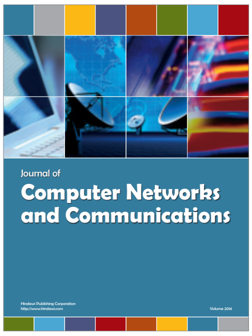

International Journal of Distributed Sensor Networks

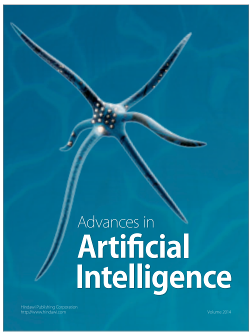

Journal of Industrial Engineering

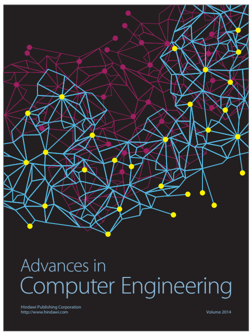

Applied Computational Intelligence and Soft Computing


Advances in Fuzzy Systems

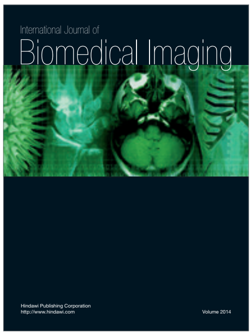

Modelling & Simulation in Engineering

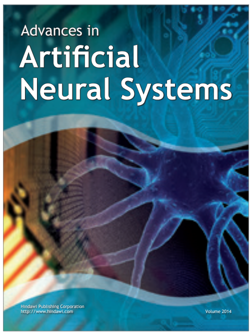

Journal of Computer Networks and Communications

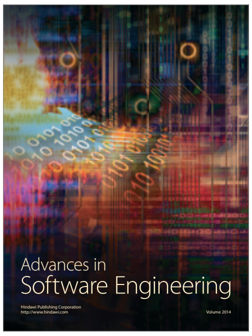

Advances in Artificial Intelligence


Advances in Computer Engineering

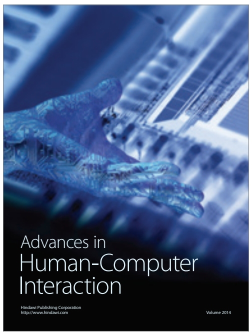
International Journal of Computer Games Technology

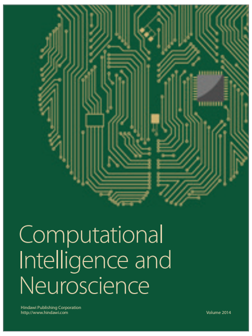

International Journal of Biomedical Imaging


Advances in Artificial Neural Systems


Advances in Software Engineering

Journal of Robotics


Advances in Human-Computer Interaction


Computational Intelligence and Neuroscience

International Journal of Reconfigurable Computing

Journal of Electrical and Computer Engineering