

RESEARCH

Open Access



Improving the interactive genetic algorithm for customer-centric product design by automatically scoring the unfavorable designs

Zahra Sheikhi Darani and Marjan Kaedi*

*Correspondence:
kaedi@eng.ui.ac.ir
Faculty of Computer
Engineering, University
of Isfahan, Hezar-Jerib St.,
Isfahan 81746-73441, Iran

Abstract

One of the effective factors in increasing sales is the consistency of products with the preference of the customers. Designing the products consistent with customer needs requires the engagement of customers in the product design process. One way to achieve this goal is the use of interactive evolutionary algorithms. During the running of such algorithms, the customer acts as a fitness function and imparts his/her opinion directly to the design process. Since these algorithms are usually iterated frequently, the user fatigue problem during interaction with them is a major challenge. The present study develops a method to tackle the user fatigue problem in the interactive genetic algorithm using the candidate elimination algorithm. In this method, customer preferences are gradually learned by applying the candidate elimination algorithm on the designs evaluated by the user in the early stages of algorithm. Using the learned preferences, designs which may not meet the customer preferences are discovered and automatically receive a predefined low score from the algorithm. The proposed method has been evaluated on the customer-centric design of book covers and its results have been compared with those of the two simple interactive genetic algorithm and multi-stage interactive genetic algorithm. The results are indicative of a considerable reduction of the number of algorithm generations, the number of chromosomes being evaluated by user, and the evaluating time in comparison with the two aforementioned methods. Reduction of these criteria leads to decrease of user fatigue. In addition, the proposed method has increased the user satisfaction.

Keywords: Interactive design, Customer-centric product design, Interactive genetic algorithm, User fatigue, Candidate elimination algorithm

Introduction

Appearance of products is one of the factors of customer attraction. This necessitates customer-centric design of products and product customization [1–3]. One of the methods proposed in this area is the direct use of designs proposed by users for designing the products. However, offering a design requires specialized knowledge and necessary skills for working with design software and tools and customers lack such requirements. Besides, strong presence in competitive markets requires quick development of

customer-centric designs for products and services. These factors have paved the way for the emergence of interactive evolutionary algorithms in the design area. Interactive evolutionary algorithms are a class of problem solving methods in which the human factor serves as a fitness function [4]. One type of interactive evolutionary algorithms is the interactive genetic algorithm which is explained in “[Genetic algorithm and interactive genetic algorithm](#)” section.

Despite the aforementioned merits, the interaction of humans with evolutionary algorithms also creates challenges which have been investigated in different studies. One of the most important of these challenges is user fatigue which is caused by the evaluation of candidate solutions (where each solution represents a candidate design) in each algorithm iteration [4, 5]. The approach adopted in the present study is prevention of undesirable solutions using the candidate elimination algorithm in combination with the interactive genetic algorithm and which reduces the number of generations of the algorithm, prevents the evaluation of designs which are probably not desirable for the users, and considerably reduces the time of achievement of a design which is desirable for the user.

In what follows, first, in “[Genetic algorithm and interactive genetic algorithm](#)” section, the genetic algorithm and the interactive genetic algorithm are briefly discussed. In “[Candidate elimination algorithm](#)” section, the candidate elimination algorithm is introduced. In “[Review of literature](#)” section, the related literature is introduced. In “[The proposed method](#)” and “[Evaluation](#)” sections, the proposed method is introduced and evaluated in comparison with two former methods. Finally, in “[Conclusion](#)” section, some conclusions are drawn.

Genetic algorithm and interactive genetic algorithm

The genetic algorithm, which is one of the best-known evolutionary algorithms, has been inspired by Darwin’s theory of natural selection [6]. In this algorithm, the initial population is produced from the candidate solutions for the intended problem. Each of the candidate solutions is called a ‘chromosome’. The population members are evaluated by fitness function and are scored based on their degree of fitness. During the process of selection, a number of chromosomes with highest scores are selected as parents to generate the next generation, and, through a crossover operation between these chromosomes, a new population is generated. Crossover operation is a process whereby the combination of two parent chromosomes generates new solutions each of which inherits some features from the first parent and some from the second. Also involved in generating the second generation are the two processes of mutation and elitism. In mutation, akin to what happens in nature, it is probable that a gene belonging to an offspring chromosome undergoes random change. This alteration affects population variety and achievement of optimal solutions. In the processes of elitism, based on elitism rate, the chromosomes which have received the highest scores from the fitness function are conveyed directly to the next generation. The generation process is iterated until optimal solutions are achieved [5, 6]. In the interactive genetic algorithm, the genes of each chromosome together describe a possible design [5]. For example, if the interactive genetic algorithm is applied for design of a book cover, the background color, image location, text font, text size and so forth are each specified by a gene. Similar to the genetic

algorithm, the first population is randomly generated. The genes of each chromosome are converted into predefined graphical features and the book cover design corresponding to that chromosome is shown to the user through a graphical interface. Afterwards, the user, serving as the fitness function, scores the candidate solutions. Then, selection, crossover, and mutation operators are applied and a new population is generated. This process is iterated until the user accesses his/her favorite book cover design.

Candidate elimination algorithm

In the method proposed in the present study, the candidate elimination algorithm is used to reduce user fatigue in the interactive genetic algorithm. Hence, in this section, this algorithm is discussed.

The candidate elimination algorithm falls into the category of concept learning algorithms where the goal is to predict whether or not a sample is a member of a set [6, 7].

In the candidate elimination algorithm, the training process is carried out on a number of samples whose labels are known as “positive” or “negative” indicating the membership or non-membership in the set, respectively. In this training process, during some stages which will be explained later on, two boundaries are determined named lower and upper boundaries. Each boundary contains some hypotheses for rejecting or accepting the samples.

The stages of the candidate elimination algorithm are as follows [6, 7]

- First, upper boundaries (G) and lower boundaries (S) are initialized in such a way that the upper boundaries are in the most general form and accept every sample and the lower boundaries are in the most specific form and accept no samples.
- Afterwards, the following process is iterated for each training sample:
 - In cases where the sample has a positive label, each hypothesis in G which rejects the sample is eliminated and boundaries in S become generalized enough to accept the sample. Modification of boundaries in S must be conducted in such a way that S becomes generalized as minimally as possible and boundaries in S stay more specific than boundaries in G .
 - In cases where the sample has a negative label, each hypothesis in S which accept the sample is eliminated and boundaries in G become specialized enough to accept the sample. Modification of boundaries in G must be conducted in such a way that G becomes specialized as minimally as possible and boundaries in G stay more general than boundaries in S .

For further explanation of the algorithm, its pseudo-code is given in Fig. 1. The main loop in this pseudo code consists of two “If” blocks, the first block is for the positive samples and the second one is for the negative samples. If sample d is positive, each boundary in G or S that is inconsistent with d is removed. The boundaries in S become generalized enough so that they accept d and be more specific than boundaries in G . In the last “Remove” statement in the first “If” block, all boundaries in S that are more specific than another boundaries in S are removed. In a similar manner, the negative

```

Initialize  $G$  to the set of maximally general hypotheses in  $H$ 
Initialize  $S$  to the set of maximally specific hypotheses in  $H$ 
For each training example  $d$ , do
  • If  $d$  is a positive example
    • Remove from  $G$  any hypothesis inconsistent with  $d$ 
    • For each hypothesis  $s$  in  $S$  that is not consistent with  $d$ 
      • Remove  $s$  from  $S$ 
      • Add to  $S$  all minimal generalizations  $h$  of  $s$  such that
        •  $h$  is consistent with  $d$ , and some member of  $G$  is more general than  $h$ 
      • Remove from  $S$  any hypothesis that is more general than another hypothesis in  $S$ 
    • If  $d$  is a negative example
      • Remove from  $S$  any hypothesis inconsistent with  $d$ 
      • For each hypothesis  $g$  in  $G$  that is not consistent with  $d$ 
        • Remove  $g$  from  $G$ 
        • Add to  $G$  all minimal specializations  $h$  of  $g$  such that
          •  $h$  is consistent with  $d$ , and some member of  $S$  is more specific than  $h$ 
        • Remove from  $G$  any hypothesis that is less general than another hypothesis in  $G$ 

```

Fig. 1 The pseudo-code of candidate elimination algorithm [7]

samples are managed in the second “If” block. The output of this algorithm is one or more boundaries in G and S .

After completion of the training process and the determination of the upper and lower boundaries, new samples falling between the two boundaries are labeled as positive (i.e., members of the set) and other samples are labeled as negative (i.e., non-members of the set).

Review of literature

The issue of interactive evolutionary computation started approximately from 1980 and it was first applied to art, graphics and animation and then to various areas such as industrial design, information retrieval, games, robotics, etc. [8]. In “[Review of the applications of the interactive evolutionary algorithm for design](#)” section, some of the applications of the interactive evolutionary algorithms for design in different areas are addressed, and, in “[Review of literature on reducing user fatigue in interactive evolutionary algorithm](#)” section, recent studies on the improvement of user fatigue in the interactive evolutionary algorithm is reviewed.

Review of the applications of the interactive evolutionary algorithm for design

Different studies have used different interactive algorithms, particularly the interactive genetic algorithm, for design in different areas. For example, one of the initial applications of this algorithm was its use in fashion design [9] in which different parts of a piece of clothing are codified as the genes (i.e., characteristics) of a chromosome and users can create their design by evaluating the designs of each generation. In a different study, to generate mobile phone designs, seven different parts of a cell phone were taken into consideration. These parts formed the structure of chromosomes, and the hierarchical interactive genetic algorithm was used to create the optimal design [10]. Another interesting area is the use of this algorithm in designing the virtual environment of games and social media. In this application, characteristics such as water level, sunlight direction, and

cloud motion are taken into consideration and codified in chromosomes. In this application, the designs existing in each population are graphically shown to the user and the user chooses three images from the presented images [11]. In another study, in an image retrieval system, the interactive genetic algorithm was used to improve the results obtained from the search. The initial results obtained from searching for an image were considered as the initial population, and the three features of background, color, and image margin were incorporated into each chromosome. Afterwards, the population was gradually improved [12]. In other studies, applications such as website appearance design [13], sign sound design [14], and greeting card design [15], carpet pattern design [16], and the design of office layout design for polygonal space [17] were applied using the interactive evolutionary algorithm.

Review of literature on reducing user fatigue in interactive evolutionary algorithm

As mentioned previously, user fatigue is one of the most important challenges to interactive evolutionary algorithms and different methods have been developed to tackle this challenge so far. In one study, the interactive evolutionary algorithm was used to design book covers [18], and the k-means clustering method was used to reduce user fatigue so that only the centroid of each cluster was shown to the user. After evaluation of cluster centroids by the user, the other members of each cluster were scored based on their similarity to the cluster centroid. This system was evaluated by three users and the results indicated a faster algorithm convergence and hence lower user fatigue. The fuzzy c-means clustering is another common method for tackling user fatigue which has been studied in recent decades. In this connection, a study has developed a system for solving unequal facility layout problem [19]. In this study, chromosomes are clustered using fuzzy c-means algorithm, and only the cluster centroids are evaluated by the user. Other cluster members are scored based on the cluster centroids scores. To evaluate the algorithm, the square and rectangular factories were used and it was demonstrated that the designs expected by the experts have been achieved after a reasonable number of algorithm iterations. Later on, in order to improve on the aforementioned study, another study was conducted in 2015 in which attempts were made to preserve population variety [20]. For this purpose, niching methods [21] were used in the interactive genetic algorithm. Comparison of this study to previous studies showed a decrease in standard deviation and in the mean number of times required to reach the optimal design. Akase and Okada in 2014 proposed a system that uses interactive evolutionary computation (IEC) to easily generate the 3D furniture layouts [22]. They proposed two prototypes; in the first prototype, the fitness function is defined based on the ergonomic constraints to reduce the burden of user evaluation. The fitness of each chromosome is calculated by summing up the inverse of the ergonomic cost function and the user's score. The user can delegate several early generations to the system. In this case, this system uses the inverse of cost function as the default fitness of chromosomes. In the second prototype, at first, the user chooses an elite chromosome. Then, in each generation 6 designs are shown to the user and the user selects the top three of them. The top two chromosomes are transferred to the next generation. Then, the one-point crossover is applied on the top two chromosomes and also on the third chromosome and elite chromosome. Mutation probability is set to 50% to prevent the initial convergence. It is shown that these operations reduce the

number of generations. Another method is proposed by Quiroz et al. for user interface design [23], in which the user just selects the best and worst designs in each generation. Then, the other designs are scored based on their similarity to these chromosomes. The chromosomes fitness is calculated by combining “the user evaluations” and “the compatibility of design with the user interface design guidelines”. Furthermore, the user evaluates the chromosomes every t generations. Results show that for the appropriate values of t , this method reduces the human fatigue and the time spent by the user on a session. Quiroz et al. in another study enhanced their method using a collaborative approach [24]. They tested their model on floorplanning. This method supports collaboration of users by allowing individual designers to view each other’s designs during the evolutionary process and share their designs via case injection. The results showed that the floorplans which are created collaboratively receive higher scores compared to the floorplans created individually. In addition, it was shown that this method reduces the user fatigue.

In another study that was conducted in 2013 to design the web pages [25], instead of evaluating the design as a whole, the user evaluates each section of a design. In other words, each gene of chromosomes is scored separately. Crossover operation is implemented in such a way that the gene which has a high score is more likely to be assigned to a child. The experiments showed that this method improves the population and reduces the number of generations.

In 2016, the “multi-stage interactive genetic algorithm” was developed to tackle user fatigue. The aim was to bring the process of interactive genetic algorithm closer to the process of design done by professional designers. In this method, the focus is on the problem of low user knowledge in the initial stages of the design process. For that purpose, first, a component of the design (e.g., the background design) is shown to the user and it is improved upon through interactive genetic algorithm operations so as to achieve user satisfaction of that component. In the next step, different settings of the next component of the design (e.g., the cover image) are added to the optimal solution of the previous evolutionary stage and the interactive genetic operations are iterated again until an optimal design for this component is made. This process continues until all components have been added to the design and the design has been completed. In this method, chromosomes become gradually more complex during the process of evolution [26]. For example, if the multi-stage interactive genetic algorithm is applied for design of a book cover and the design items include the background color, the image location, and the text location, the following steps will be performed in this algorithm:

- (1) The initial population is generated randomly. In this population each chromosome has just one gene (e.g., the background color).
- (2) The user evaluates each chromosome. Afterwards, the new population is generated by applying the selection, crossover, and mutation operators.
- (3) Step 2 is iterated until the user achieves his/her favorable design component (e.g., the background color).
- (4) Previous component (e.g., the background color) will be fixed and the new component (e.g., the image) is added to the design.
- (5) Steps 2, 3, and 4 are iterated until all components are added to the design and the user achieves his/her desired book cover design.

It was claimed that this trend in multi-stage interactive genetic algorithm gradually increases users' knowledge of the design and diminishes the problem of low user knowledge for evaluating the whole design in the early stages of the algorithm (which lengthens the algorithm and increases user fatigue) [26].

Considering that the multi-stage interactive genetic algorithm is new and efficient, the method proposed in the present paper is compared with the results of the multi-stage interactive genetic algorithm.

The proposed method

In the proposed method, the candidate elimination algorithm is used in combination with the interactive genetic algorithm in order to identify the user preference in scoring the designs and to prevent the evaluation of inappropriate solutions (i.e., designs which do not match user preference) by user.

In this method, first, a population of candidate solutions (i.e., candidate designs) is generated randomly by interactive genetic algorithm and the graphical presentation of each solution is shown to the user. The user scores each design in the population based on his own preferences. Having been scored by the user, each design is then sent to the candidate elimination algorithm as a training sample. For this purpose, designs with scores higher than average are considered as positive samples and others are considered as negative samples.

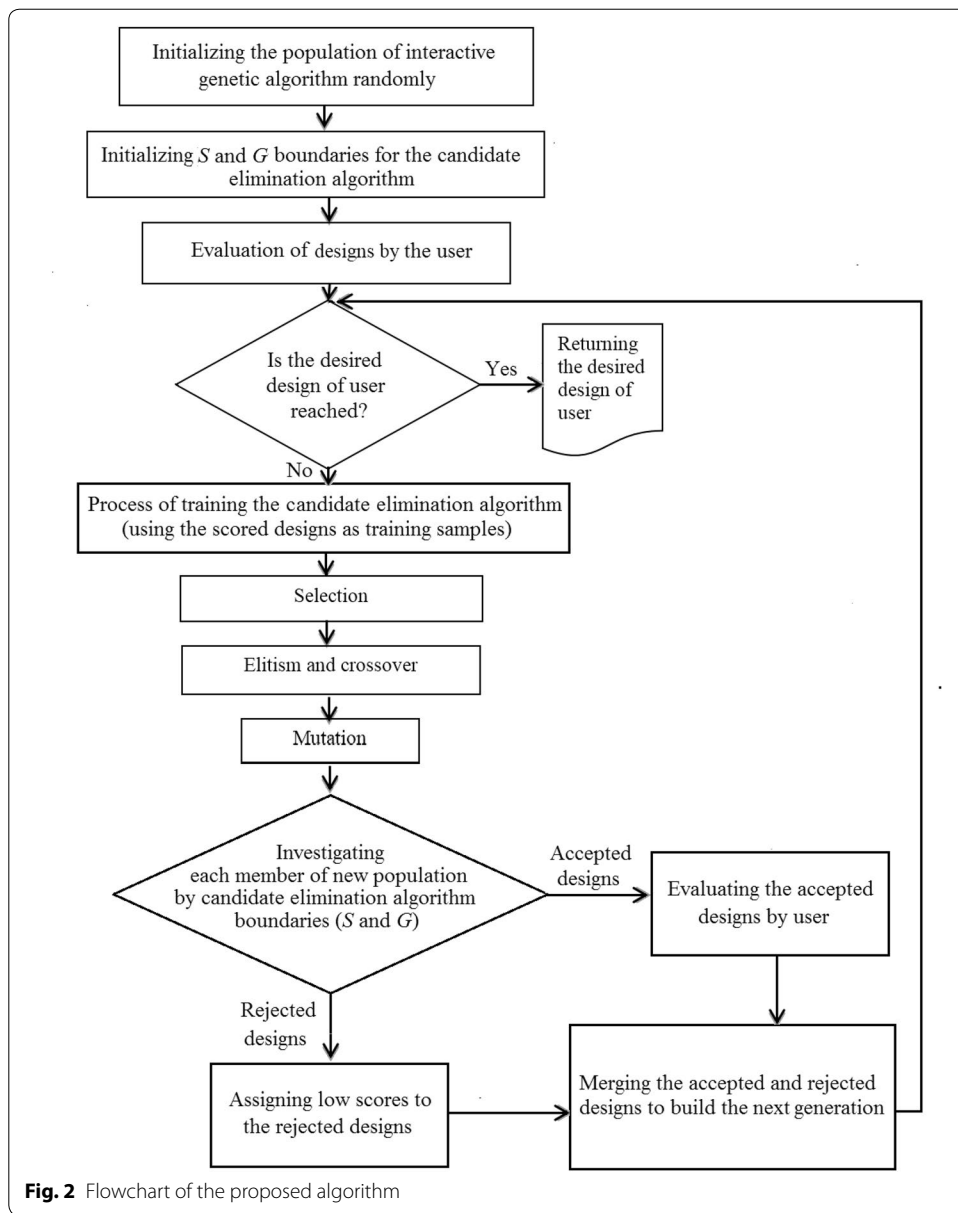
Afterwards, using elitism, selection, crossover, and mutation operators in the interactive genetic algorithm, a new population of candidates is generated. In this process, each generated offspring, prior to being added to the new population, is investigated using the candidate elimination algorithm and only the offspring accepted by this algorithm, i.e., those falling between the upper and lower boundaries of the candidate elimination algorithm, are shown to the user. Other offspring receive a predefined low score from the algorithm. This process is iterated until the end of the interactive genetic algorithm. Therefore, the two steps in our proposed algorithm that differentiate it from the previous interactive genetic algorithms are as follows:

- In each iteration of the algorithm, the population members scored by the user are presented to the candidate elimination algorithm as training samples so that the boundaries of the candidate elimination algorithm become increasingly precise.

For evaluating the members of each generation of interactive genetic algorithm, the offspring accepted by the candidate elimination algorithm are showed to the user to be scored. On the other hand, the designs rejected by the candidate elimination algorithm will probably be rejected by the user too; thus, they automatically receive a predefined low score from the algorithm and will not be shown to the user. Therefore, the number of evaluations to be done by the user is decreased and the user fatigue is reduced.

Actually, another approach to handle the designs that are rejected by the candidate elimination algorithm is to discard them and not to transfer them to the next generation; however, our experiments showed that discarding the rejected designs reduces the population diversity and decreases the algorithm efficiency.

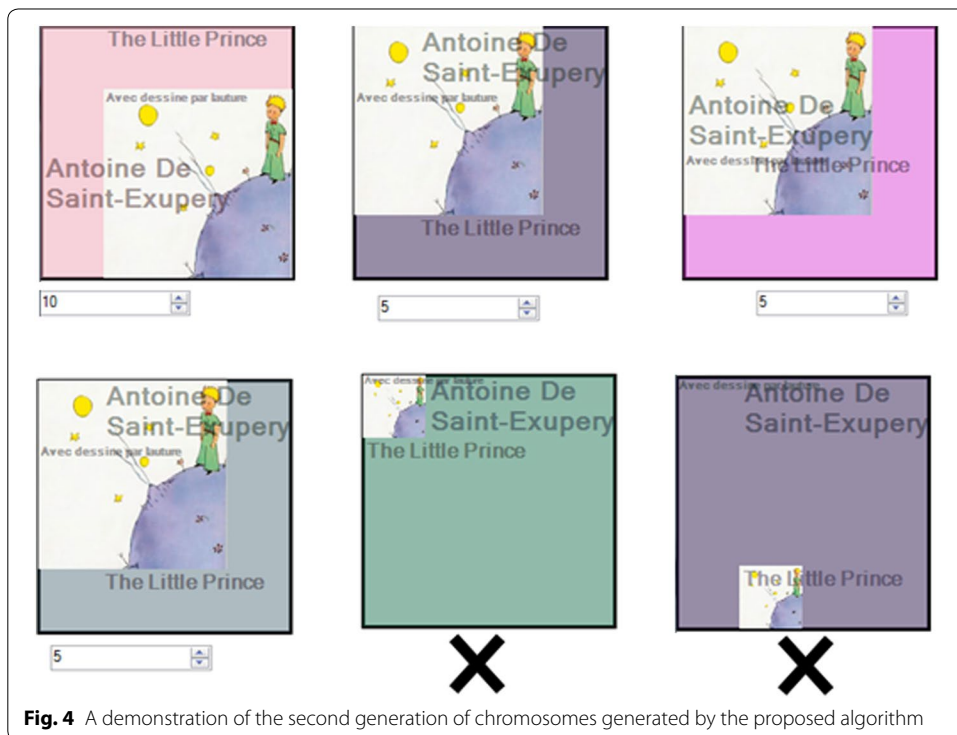
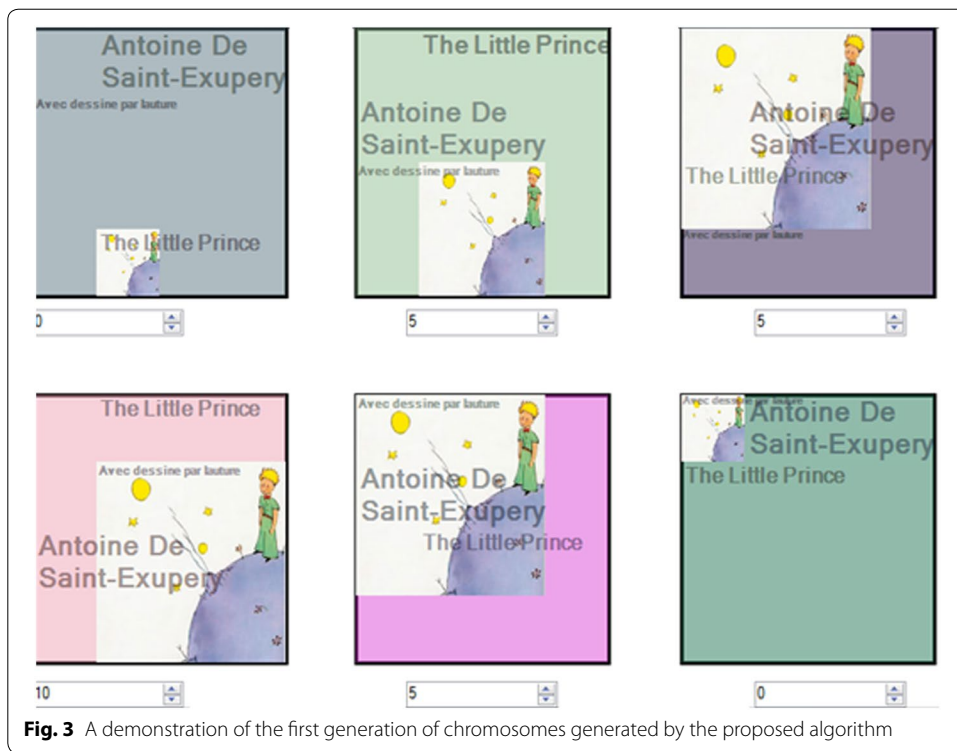
Figure 2 illustrates the flowchart of the proposed method.



Figures 3, 4, and 5 illustrate a sample of populations in three consecutive generations for the problem of book cover design. Designs which are marked by “×” signs are identified as inappropriate designs by the candidate elimination algorithm and get a low score from the algorithm.

Evaluation

In this section, the proposed method is evaluated. First, the design in question is elaborated on, and, afterwards, the process of evaluation as well as evaluation results is discussed. It should be noted that the proposed method is application independent and can be used for other design problems.



Graphical design

In the present study, the proposed method is evaluated on the problem of book cover design. Each book cover design includes the three components of background, image,

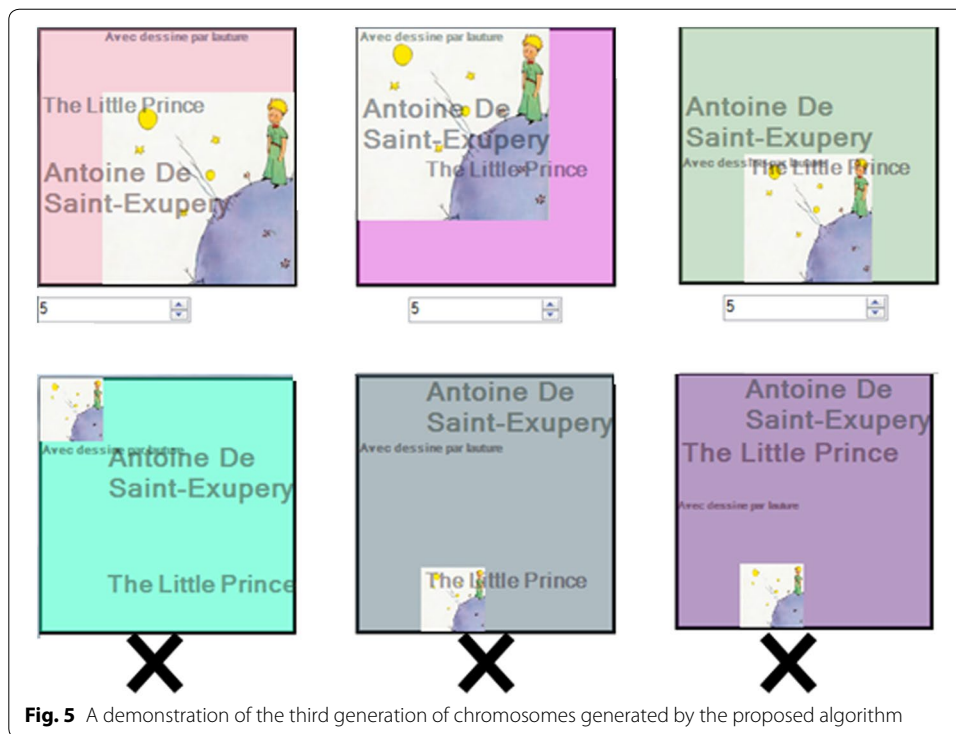


Fig. 5 A demonstration of the third generation of chromosomes generated by the proposed algorithm

and text, each of which has their specific characteristics. To describe these components and their characteristics, the study conducted by Yu et al. [18] was used. Table 1 presents the components and the number of bits relating to each component. The RGB system was used for the color characteristics and 8 bits were considered for each of the three main colors. Therefore, each color can receive a value between 0 and 255. The font characteristic can be one of the four popular fonts, i.e., Cambria, Calibri, Times New Roman, and Arial; therefore 2 bits are used to describe it. The size characteristic can have 4 different values for each image and text components; hence, 2 bits are considered for this characteristic. Considering that the cover design sheet is considered as a grid with two columns and four rows, the position characteristics has eight values. Therefore, this characteristic is demonstrated by 3 bits. Considering the characteristics of each component, in sum, the size of each chromosome which describes a cover design equals 60 bits.

Parameters setting

In the interactive genetic algorithm, the size of populations is usually considered a number below 10. In the multi-stage interactive genetic method [26], which has been

Table 1 Components of each chromosome and the characteristics of each component for book cover design

Components	Font	Position	Size	Color
Background	–	–	–	24 bits
Text	2 bits	3 bits	2 bits	24 bits
Image	–	3 bits	2 bits	–

compared with the methods proposed in this paper, the population size has been considered to be 6. In order to preserve the similarity, the size of all populations in the present paper has been decided to be 6. In addition, in choosing the rate of genetic operators and the details of conducting them, attempts have been made to follow the settings of the multi-stage interactive genetic algorithm. Therefore, the crossover rate has been considered 0.9, and single point crossover has been used. Crossover operation is implemented in such a way that it does not disturb the genes. To do this, the crossover point is selected from among the separation points of genes.

In mutation, which occurs with a probability of 0.02, a bit is randomly selected and undergoes a change. The elitism rate has been assumed to be 0.2 so that the best chromosomes of one generation proceed to the next generation. The value "1" is assigned to the designs that are rejected by the candidate elimination algorithm, as a predefined low score.

Evaluation procedure

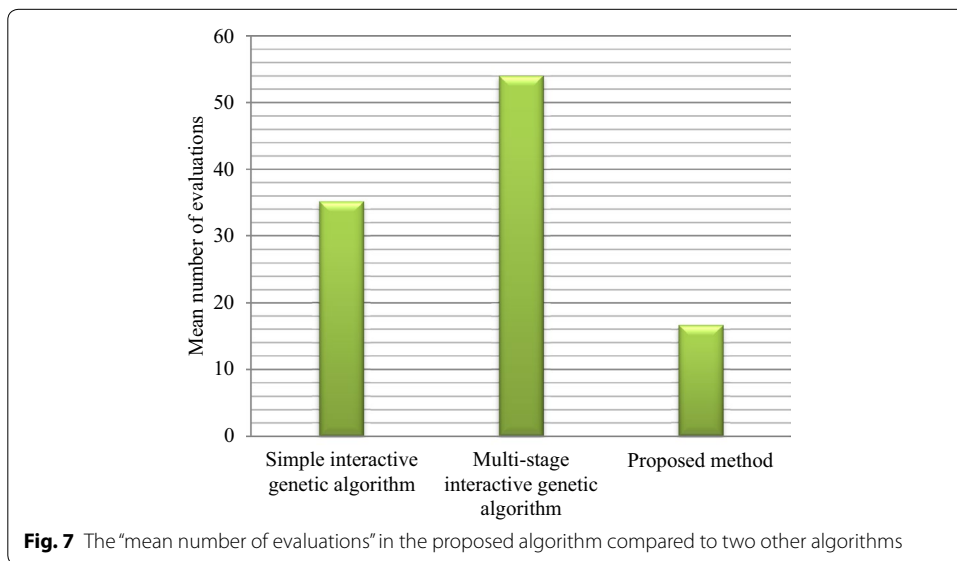
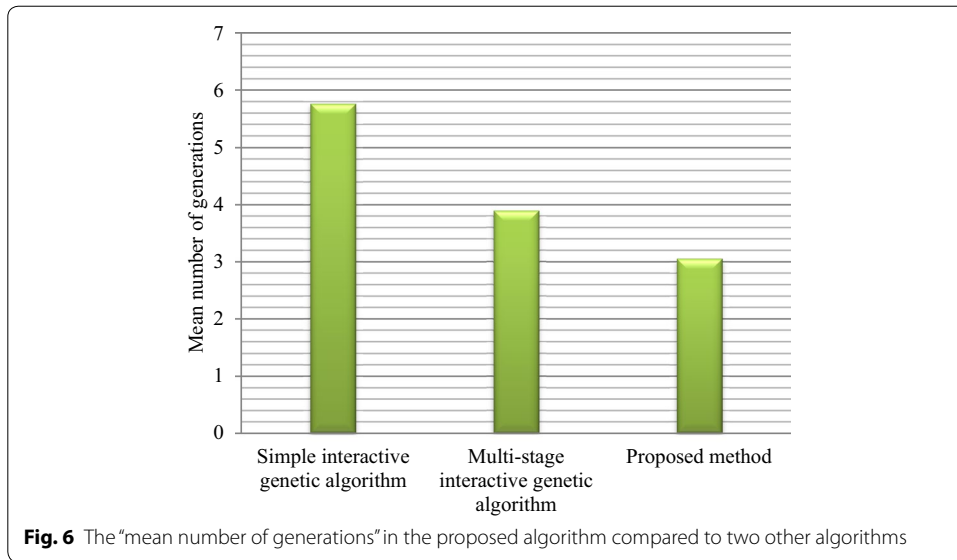
To evaluate the proposed method, 15 users were asked to perform the cover design process separately using the simple interactive genetic algorithm, multi-stage interactive genetic algorithm, and the proposed algorithm. As discussed in "[Review of literature on reducing user fatigue in interactive evolutionary algorithm](#)" section, the multi-stage interactive genetic algorithm is one of the recent methods for reducing user fatigue. In this algorithm, the design process is constituted by a few evolutionary stages each of which adds a new component to the design [26]. This method was implemented and compared with the proposed method for book cover design. Using each algorithm, each user performs the design process three times starting from three different initial populations. The three populations are identical for all three algorithms.

The mean number of generations, the mean number of chromosomes being evaluated by user, and the mean evaluating time are considered as the evaluation criteria. The values of these criteria averaged over the 15 users for the three algorithms are presented in Figs. 6, 7, and 8.

In order to measure the degree of satisfaction in addition to these criteria, the After Scenario Questionnaire (ASQ) [27] was employed. Using this questionnaire, the degree of user satisfaction with the ease of completing the design process and also the degree of user satisfaction with the time spent on designing were measured. The user's answer to each question in the questionnaire was a number between 1 (total agreement) and 7 (total disagreement). Each user filled out the questionnaire after the completion of the design process. Afterwards, the mean values of the indices (demonstrated in Figs. 9, 10) were determined on the basis of users' answers to the questions. It is worth noting that the mean values determined for each index have been inverted and multiplied by 100 so as to make them appropriate for display on the diagram. Therefore, in these diagrams, higher values demonstrate higher user satisfaction.

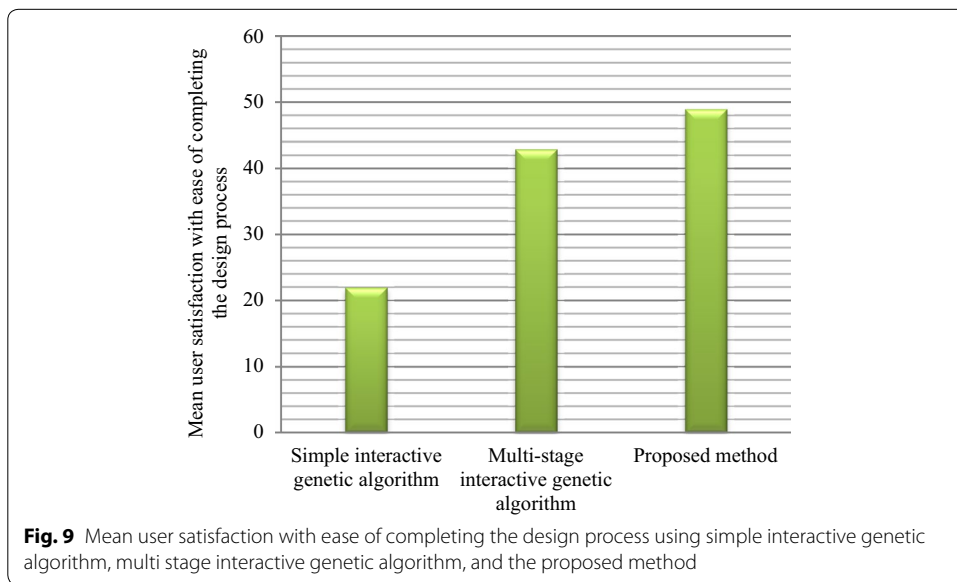
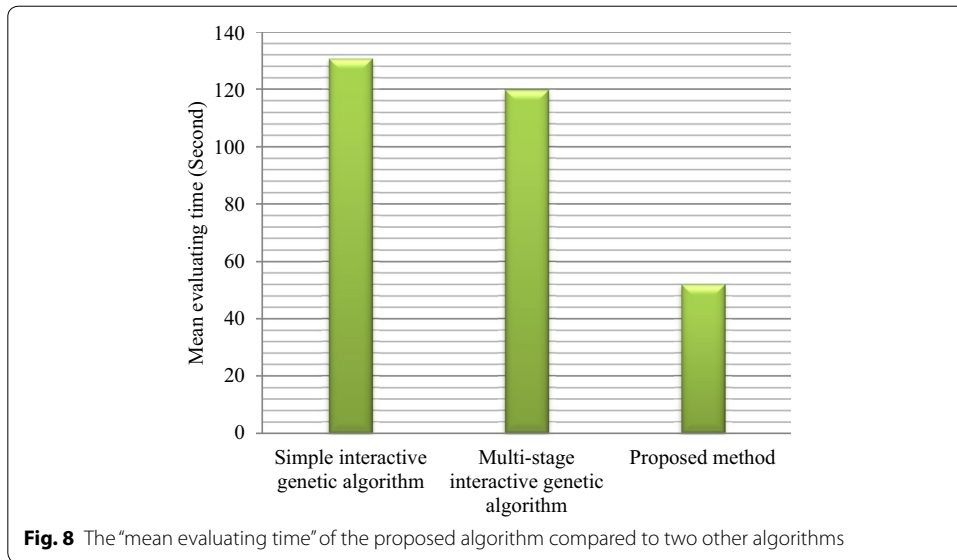
Results analysis

As indicated by the results, compared with the simple interactive genetic algorithm, the proposed algorithm demands fewer number of design evaluations made by user and achieves the user's desired design within a shorter time and after fewer algorithm



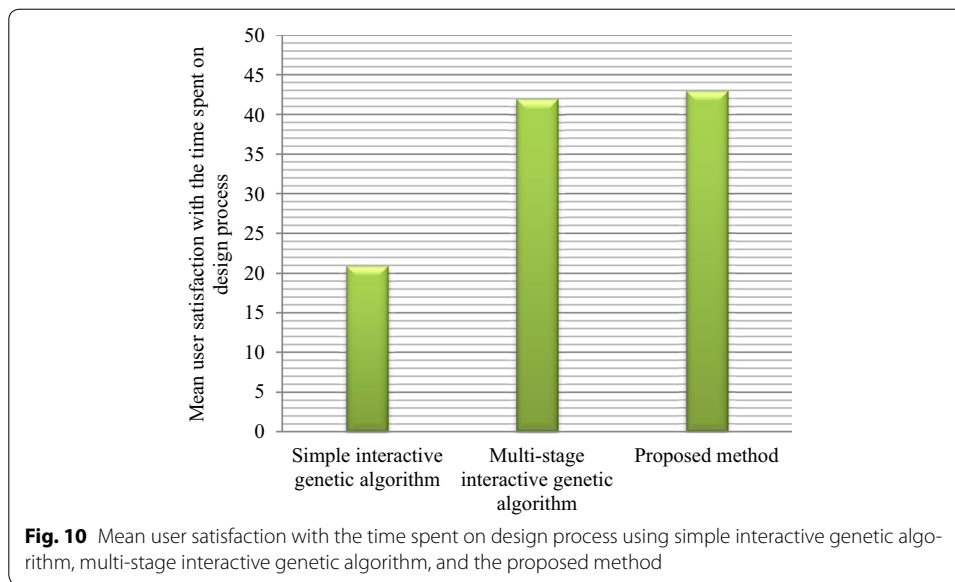
generations. This reduces user fatigue in the proposed method. The 60% decrease in the evaluation time, the 46% decrease in the number of generations, and the 53% decrease in the number of evaluation demonstrate that the proposed method can efficiently ascertain user preference and improve the design process by screening generated chromosomes and predicted the chromosomes which are not consistent with user preference. In the proposed method, the combination of candidate elimination algorithm and interactive genetic algorithm leads to the fast identification of user preferences. Therefore, the time to achieve the user’s desired design and the number of generations are reduced. In addition, scoring undesirable designs by the algorithm is another factor that reduces the number of user evaluations.

Besides, in comparison with the multi-stage interactive genetic method, the proposed method has created a 56% improvement in the evaluation time and a 21% improvement



in the number of generations and 69% improvement in number of evaluations made by users. By performing a step by step design process, the multi-stage interactive genetic algorithm increases the focus of user on a component at each stage of the design. For this reason, compared to the simple interactive genetic algorithm, less time and fewer number of generations are needed to reach the user’s target. However, the results indicate that the proposed method performs better than multi-stage interactive genetic algorithm. This is because the step by step design process increases the number of evaluations needed in the multi-stage algorithm.

The great difference between the number of evaluations obtained by our proposed method and that of the multi-stage interactive genetic algorithm is due to the fact that the number of required evaluations for achieving the desired design in the multi-stage interactive genetic algorithm is at least equal to the multiplication of the number of



design components by the population size. In our experiment, each book cover design includes five components. Therefore, five stages are required to complete the design, each of which includes evaluation of all population chromosomes. On the other hand, for some users, our proposed method may find the user's desired design in the initial population. Hence, this great difference in the number of evaluations in our proposed method and multi-stage interactive genetic algorithm is natural. For the same reason, the number of evaluations in multi-stage interactive genetic algorithm is greater than that of simple interactive genetic algorithm. However, as declared by the authors [26], in the multi-stage interactive genetic algorithm the user is able to judge the design features more effortlessly because of utilizing a step by step design process. For example, in our experiments, only the background color is evaluated by the user in the first stage of algorithm; in the next stage, the background color has been specified and just several locations for the image should be evaluated by the user, and so on. Therefore, even though the mean number of evaluations in simple interactive genetic algorithm is lower than that of multi-stage interactive genetic algorithm, the user can evaluate the designs more quickly in the multi-stage interactive genetic algorithm. Thus, the mean evaluating time of the multi-stage interactive genetic algorithm is lower than that of the simple genetic algorithm. The results demonstrated in Figs. 7 and 8 confirm this issue. To investigate the significance of the statistical results, the standard deviation metric is used [20]. Table 2 shows the average and standard deviation of evaluation criteria (i.e., the mean number of generations, the mean number of chromosomes being evaluated by user, and the mean evaluating time) for each algorithm. In addition, the average and standard deviation of chromosome's scores in each generation of algorithms are presented in Table 3. As shown in these tables the standard deviation obtained using the proposed method is smaller than those of other two algorithms which means that our proposed method is more consistent in this regard compared to other algorithms.

Figures 9 and 10 demonstrate that the degree of users' satisfaction with "the ease of completing the design process" and also the degree of users' satisfaction with "the time

Table 2 The standard deviation and the average of the evaluation criteria for the algorithms

	The number of generations		The number of evaluations		The mean of evaluating time (second)	
	Std. dev	Average	Std. dev	Average	Std. dev	Average
Simple interactive genetic algorithm	2.59	5.77	13.76	35.33	92.93	130.97
Multi-stage interactive genetic algorithm	2.19	3.91	29.64	54.13	57.12	119.95
Proposed method	1.29	3.06	5.16	16.71	21.50	52.18

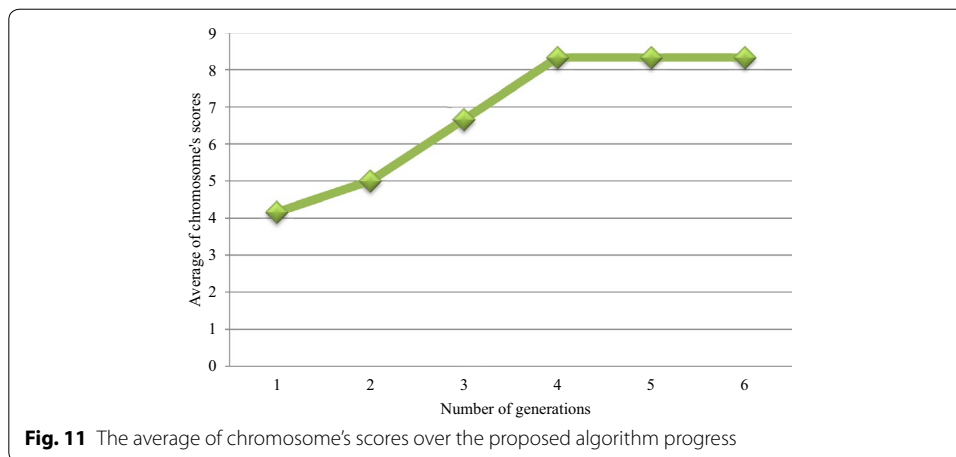
Table 3 The standard deviation and the average of the chromosome's scores in five generations of algorithms

Number of generations	Simple interactive genetic algorithm		Multi-stage interactive genetic algorithm (last stage)		Proposed method	
	Std. dev	Average	Std. dev	Average	Std. dev	Average
1	3.43	8.83	3.81	2.50	3.43	4.16
2	2.88	5	4.48	4.16	3.53	5
3	3.72	3.33	3.81	2.50	2.35	6.66
4	4.08	5	3.43	4.16	2.35	8.33
5	3.72	3.33	4.71	3.33	2.35	8.33

they spent on designing" are the highest in the proposed algorithm and the lowest in the simple interactive genetic algorithm. In the proposed algorithm, the designs that are predicted to be unfavorable for the user are scored automatically by the algorithm; hence, the user does not need to score them and he/she requires fewer evaluations. This makes the algorithm more attractive to the user and enhances the user's satisfaction with his/her interaction with the algorithm. On the other hand, the multi-stage interactive genetic algorithm enables the user to focus on a single component in each stage. However, if an inappropriate component is selected by the user at the beginning stages of the design process when he/she has not still observed its combination with other components, the user may fail to generate a desired design and hence, his/her satisfaction is decreased.

As it is shown in Fig. 8, the mean evaluating time in the proposed method is much lower than that of multi-stage interactive genetic algorithm. Furthermore, the user satisfaction about the time spending on the design process is higher than that of multi-stage interactive genetic algorithm (Fig. 10). Although the difference in the user satisfaction indicates the success of the proposed method in accelerating the design process, the improvement in the user satisfaction is marginal. It can be concluded that despite the long time users spend on using multi-stage interactive genetic algorithms, the step by step design process is interesting to the users.

Figure 11 demonstrates the average of chromosome's scores in each generation of the proposed method. As it is shown in this figure, the average score of chromosomes is gradually increased over the run of algorithm and finally it is converged. This means that the number of user desirable designs is improved during the progress of algorithm and finally, in sixth generation the user finds his/her favorite design and stops the process.



It should be notified that only the score of chromosomes that are not eliminated by the candidate elimination algorithm are considered in calculating the average scores.

Conclusion

In this paper, a method was proposed to reduce user fatigue in interactive genetic algorithm. In this method, using the candidate elimination algorithm, inappropriate chromosomes (i.e., designs) are predicted and scored by the algorithm with a predefined low value. Evaluation of the proposed method indicated a decrease in evaluation time and a decrease in the number of algorithm generations in comparison with the simple interactive genetic algorithm and the multi-stage interactive genetic algorithm. The proposed method can be utilized for customer-centric design of wide range of products such as cellphones, cars, and clothing. Furthermore, it can be employed in applications such as office interior layout design, webpage graphical design, and software user interface design. In the mentioned applications the user fatigue, which is one of the most important problems during the design process, can be alleviated by our proposed method.

In addition to reduction of user fatigue, it seems that in the proposed method the boundaries of the candidate elimination algorithm which are gradually made on the basis of user scoring of chromosomes can prevent paradoxical evaluations by the user as well as imprecise scoring which are the shortcomings of other interactive design algorithms. This may be studied and evaluated in future research.

In addition, in future research, the proposed method may be used in the design of other products than book covers, and the capability of this method may be assessed by more users. Another guideline to improve the current study is to protect the system against the malicious users who want to intentionally make the algorithm unusable. This issue may be resolved in the future studies using the authentication mechanisms [28, 29].

Another challenge in the proposed method is that if the user's preferences change widely during the design process, the algorithm may be misled in learning the preferences. One idea to address this problem is to apply a low training rate while the algorithm is in the beginning generations and the user is not still sure about his/her preferences. The algorithm training rate can be increased gradually while the algorithm proceeds and the user's preferences become stable. Besides, due to the dynamic nature of

customer-centric design process, the evolutionary methods presented for dynamic environments [30, 31] can be applied to improve the method.

Authors' contributions

Both the authors contributed significantly to the research and this paper. Both authors read and approved the final manuscript.

Authors' information

Zahra Sheikhi Darani holds an MSc degree in Information Technology from the Faculty of Computer Engineering, University of Isfahan, Iran. Her research interests include interactive evolutionary algorithms, customer-centric product design, and electronic commerce.

Marjan Kaedi is an Assistant Professor of Faculty of Computer Engineering, University of Isfahan, Iran. Her research interests include electronic commerce, artificial intelligence, and optimization.

Acknowledgements

The authors wish to express their gratitude to the Iranian E-Commerce Scientific Association and Mobin Iran Electronic Development Company under whose auspices the present study was conducted.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

Not applicable.

Consent for publication

Not applicable.

Ethics approval and consent to participate

Not applicable.

Funding

The authors wish to express their gratitude to the Iranian E-Commerce Scientific Association and Mobin Iran Electronic Development Company under whose auspices the present study was conducted.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 24 March 2017 Accepted: 3 October 2017

Published online: 11 October 2017

References

- Bloch P (1995) Seeking the ideal form: product design and consumer response. *Mark Sci* 59:16–29
- Truong Y, Klink RR, Fort-Rioche L, Athaide GA (2014) Consumer response to product form in technology-based industries. *J Prod Innov Manag* 31:867–876
- Wang Y, Tseng MM (2011) Integrating comprehensive customer requirements into product design. *CIRP Annals Manuf Technol* 60:175–178
- Ono S, Maeda H, Sakimoto K, Nakayama S (2014) User-system cooperative evolutionary computation for both quantitative and qualitative objective optimization in image processing filter design. *Appl Soft Comput* 15:203–218
- Brintrup AM, Ramsden J, Tiwari A (2007) An interactive genetic algorithm-based framework for handling qualitative criteria in design optimization. *Comput Ind* 58:279–291
- Russell S, Norvig P (1995) *Artificial intelligence a modern approach*. Prentice Hall, New Jersey
- Mitchell TM (1997) *Machine learning*. McGraw-Hill Science, Boston
- Takagi H (2001) Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. *Proc IEEE* 89:1275–1296
- Kim H-S, Cho S-B (2000) Application of interactive genetic algorithm to fashion design. *Eng Appl Artif Intell* 13:635–644
- Lee J-H, Chang M-L (2010) Stimulating designers' creativity based on a creative evolutionary system and collective intelligence in product design. *Int J Ind Ergon* 40:295–305
- Frade M, Fernandez de Vega F, Cotta C (2009) Breeding terrains with genetic terrain programming: the evolution of terrain generators. *Int J Comput Games Technol* 2009:13
- Chih-Chin L, Ying-Chuan C (2011) A user-oriented image retrieval system based on interactive genetic algorithm. *IEEE Trans Instrum Meas* 60:3318–3325
- Venturini G, Monmarché N, Oliver A (2002) "Interactive design of websites with a genetic algorithm", presented at the ICWI2002, Lisbon. pp 355–362
- Miki M, Orita H, Wake SH, Hiroyasu T (2006) "Design of sign sounds using an interactive genetic algorithm". In: 2006 IEEE international conference on systems, man and cybernetics (SMC). pp 3486–3490
- Yang H-F (2016) Development of a self-design system for greeting cards on the basis of interactive evolutionary computation. *Kybernetes* 45:521–535

16. Zamani F, Amani-Tehran M, Latifi M (2009) Interactive genetic algorithm-aided generation of carpet pattern. *J Text Inst* 100:556–564
17. Araki Y, Osana Y (2012) Office layout support system for polygonal space using interactive genetic algorithm; Generation of layout plans for workspace. In: 2012 IEEE international conference on systems, man, and cybernetics (SMC). pp 1039–1044
18. Yu F, Li Y, Kuang L, Li Z (2014) Developing support system for cover design with interactive evolutionary computation. In: The 26th Chinese control and decision conference (2014 CCDC). pp 408–413
19. Garcia-Hernandez L, Pierreval H, Salas-Morera L, Arauzo-Azofra A (2013) Handling qualitative aspects in unequal area facility layout problem: an interactive genetic algorithm. *Appl Soft Comput* 13:1718–1727
20. García-Hernández L, Palomo-Romero JM, Salas-Morera L, Arauzo-Azofra A, Pierreval H (2015) A novel hybrid evolutionary approach for capturing decision maker knowledge into the unequal area facility layout problem. *Expert Syst Appl* 42:4697–4708
21. Mahfoud SW (1995) Niching methods for genetic algorithms. University of Illinois, Urbana-Champaign, p 1995
22. Akase R, Okada Y (2014) WebGL-based 3D furniture layout system using interactive evolutionary computation and its user evaluations. *Int J Space Based Situat Comput* 4:143–164
23. Quiroz JC, Louis SJ, Shankar A, Dasalu SM (2007) Interactive genetic algorithms for user interface design. In: 2007 IEEE congress on evolutionary computation. pp 1366–1373
24. Quiroz JC, Louis SJ, Banerjee A, Dasalu SM (2009) Towards creative design using collaborative interactive genetic algorithms. In: 2009 IEEE congress on evolutionary computation. pp 1849–1856
25. Sorn D, Rimcharoen S (2013) Web page template design using interactive genetic algorithm. In: 2013 international computer science and engineering conference (ICSEC). pp 201–206
26. Dou R, Zong C, Nan G (2016) Multi-stage interactive genetic algorithm for collaborative product customization. *Knowl Based Syst* 92:43–54
27. Lewis JR (1995) IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *Int J Hum Comp Interact* 7:57–78
28. Wang D, Wang P (2017) Two birds with one stone: two-factor authentication with security beyond conventional bound. In: IEEE transactions on dependable and secure computing. pp 1
29. Wang D, Wang N, Wang P, Qing S (2015) Preserving privacy for free: efficient and provably secure two-factor authentication scheme with user anonymity. *Inf Sci* 321:162–178
30. Kaedi M, Ghasem-Aghaee N, Ahn CW (2013) Evolutionary optimization in dynamic environments: bringing the strengths of dynamic Bayesian networks into Bayesian optimization algorithm. *Int J Innov Comput Inf Control* 9:2485–2503
31. Kaedi M, Ghasem-Aghaee N, Ahn CW (2016) Biasing the transition of Bayesian optimization algorithm between Markov chain states in dynamic environments. *Inf Sci* 334:44–64

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ springeropen.com
