This article can be cited as Z. S. Darani and M. Kaedi, Reducing the User Fatigue in Interactive Design: Utilizing Interactive Genetic Algorithm and the Desired Designs of Former Users, International Journal of Artificial Intelligence, vol. 18, no. 2, pp. 113-131, 2020.

Reducing the User Fatigue in Interactive Design: Utilizing Interactive Genetic Algorithm and the Desired Designs of Former Users

Zahra Sheikhi Darani[1] and **Marjan Kaedi**[2]

[1]Faculty of Computer Engineering, University of Isfahan,
Azadi Sq., Hezarjarib St., Isfahan, Iran
Email: sheikhy.zahra@gmail.com

[2]Faculty of Computer Engineering, University of Isfahan,
Azadi Sq., Hezarjarib St., Isfahan, Iran
Email: kaedi@eng.ui.ac.ir

ABSTRACT

Interactive evolutionary algorithms are a class of evolutionary algorithms adopted for customer centric product design. During the run of such algorithms, the customer (user) acts as a fitness function to evaluate the candidate designs based on his/her interests and preferences. These algorithms are usually iterated frequently to find the desirable design of customer; hence, the user fatigue problem during interaction with these algorithms is a major challenge. The present study develops a method to tackle this problem. In this method, the desired designs of former users are considered as valuable knowledge to support the algorithm execution in the future. This knowledge is applied to enrich the populations of interactive genetic algorithm to speed up finding the desired designs of users. The proposed method has been used for customer centric design of book covers. The results show that the proposed method improves the speed of algorithm and increase the user satisfaction.

Keywords: Customer centric product design, customer preferences modeling, interactive genetic algorithm, user fatigue.

Computing Classification System: Human centered computing, Interaction design process and methods

1. INTRODUCTION

A key factor for customer attraction in competitive markets is to design the attractive appearance for the products (Bloch 1995). The consistency of products with the customers' preferences increases the product sales (Crilly et al. 2004). However, different customers have different preference and interests. To increase the consistency, customers should be engaged in the process of product design. The methods which incorporate the customers in product design process are called customer centric product design. Some optimization algorithms can take a role in customer centric product design. Optimization problems face the challenge of finding the best solution among a big set of feasible solutions. There are many famous optimization issues in artificial intelligence such as knapsack problem, traveling salesman problem, path planning and so on (Osaba et al. 2018; Purcaru et al. 2013). These issues are used to model and solve the real optimization problems (Alvarez Gil et al. 2018; Purcaru and David 2019). Many optimization algorithms like as genetic algorithms (GAs) are used to solve these problems by maximizing or minimizing a mathematical function which models the problem (Rahmi et al. 2020; Duong et al. 2017).

Interactive genetic algorithms (IGAs) are a class of GAs adopted for customer centric product design. In IGA, a population of candidate designs are evolved iteratively until reaching an acceptable design. In IGA, the goal of optimization is to find the design that is compatible to the user's preferences. Therefore, no mathematical function is defined to model the problem in IGA. Instead, the optimization problem is modelled gradually based on the user's scores to candidate designs. During the evolution process of IGA, the customer evaluates and scores the candidate designs, in contrast to traditional GA where the candidate solutions are scored automatically using a fitness function (Ono et al. 2014; Brintrup et al. 2007). Both the traditional and interactive GAs are explained in Section 2. Despite the advantages of IGA, the interaction of humans with the algorithm faces some challenges (Ono et al. 2014; Brintrup et al. 2007). One of the important challenges is user fatigue which is caused by the evaluation of candidate designs during the execution of algorithm (Ono et al. 2014; Brintrup et al. 2007).

The IGA works in an iterative manner; it usually requires several number of iteration to reach the desired design of user, and several candidate designs should be scored by the user in every iteration. These frequent evaluations lead to user fatigue, and user fatigue, itself, may lead to other problems such as decreasing the evaluation accuracy and reducing the user satisfaction. This challenge has been investigated in several studies. A review of these studies are presented in Section 3.2.In the present study a new approach is presented to tackle the user fatigue in IGA. In the approach adopted in the present study, the desired designs of former users are stored in a database. These designs are used for tune up the algorithm's initial population. In addition, during the execution of algorithm, the users whose evaluation manner are similar to that of the current user are selected and their desired designs are injected to the populations of IGA. In this way, the evolutionary process of finding the user's desired design is accelerated and the user fatigue is reduced.

In what follows, first, in Section 2, the GA and the IGA are briefly discussed. In Section 3, the related literature is introduced. In Sections 4 and 5, the proposed method is introduced and evaluated in comparison with two former methods. Finally, in Section 6, some conclusions are drawn.

2. GENETIC ALGORITHM (GA) AND INTERACTIVE GENETIC ALGORITHM (IGA)

The GA has been inspired by Darwin's theory of natural selection (Mitchell 1997). In this algorithm, a population of candidate solutions are gradually evolved to obtain a high quality solution for a problem. The initial population consists of the candidate solutions of problem. Each of the candidate solutions is called a 'chromosome' which consists of a group of features called genes. The population members are evaluated by fitness function and are scored based on their degree of fitness. In a process called selection, a number of chromosomes (probably the chromosomes with high scores) are selected as parents to generate the next population, and, through crossover operation between these chromosomes, a new population is generated. Crossover operation is a process whereby two parent chromosomes are combined to generate offspring solutions each of which inherits some features from the first parent and some from the second. Also two operators called mutation and elitism are involved in generating the next population. In mutation, akin to what happens in nature, it is probable that a gene belonging to an offspring chromosome undergoes random changes. This alteration increases population variety which is a requirement for achieving a high quality solution. In the elitism operation, based on elitism rate, some chromosomes which have received the highest scores from the fitness function are conveyed directly to the next generation. The algorithm iterates until the termination condition is reached (Mitchell 1997; GUPTA 2015).

In the IGA, the chromosomes correspond to the candidate designs and each gene corresponds to a design feature. In the evaluation stage, the genes of each chromosome are decoded into the (graphical) features and the design relating to that chromosome is shown to the user through a graphical interface. Afterwards, instead of using a fitness function for automatic evaluation of chromosomes, the user, acting as the fitness function, scores the candidate solutions (Ono et al. 2014; Brintrup et al. 2007). Interactive evolutionary algorithms like IGA are used for issues which depend on the human feeling and interest. In these problems, fitness function can not be modelled by mathematical formulas, but the user acts as fitness function. In these problems, the optimization means creating the solution that satisfies the user preferences. Figure 1 represents the flowchart of IGA.

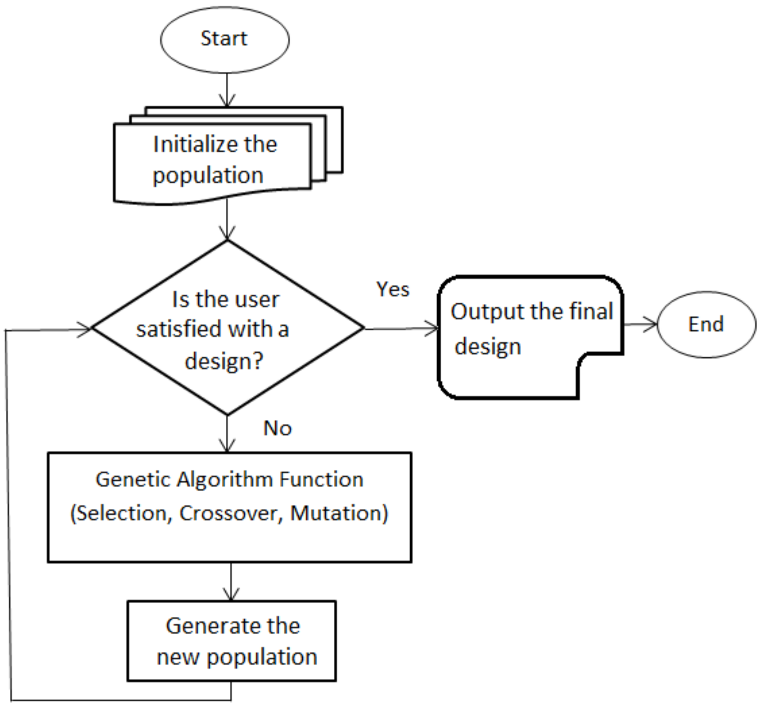

Figure 1. Flowchart of IGA. The flowchart stages have been inspired by former literature (Ono et al. 2014; Brintrup et al. 2007).

3. REVIEW OF LITERATURE

The issue of interactive evolutionary computation started approximately from 1980. It was first applied to art, graphics and animation and then to various areas such as industrial design, information retrieval, games, robotics, etc. (Takagi 2001). In Section 3.1, some of the applications of the interactive evolutionary algorisms for design in different areas are addressed, and, in Section 3.2, recent studies on the reduction of user fatigue in the interactive evolutionary algorithm are reviewed.

3.1. A review of the applications of the interactive evolutionary algorithm for design

Different studies have used IGA for design in different areas. One of the initial applications of this algorithm was in the field of fashion design (Kim and Cho 2000) in which different parts of a piece of clothing were encoded as the characteristics of a chromosome and users could create their design by evaluating the designs of each generation. In a different study, to design the cell phones, seven parts of a cell phone appearance were taken into consideration. These parts formed the structure of chromosomes, and the hierarchical IGA was used to create the desired design for the designer person (Lee and Chang 2010).

Affordance based design (ABD) is another recent applications of IGA (Mata et al. 2018). This design method focuses on the perceived interactions between users and products. Applying the integration of IGA and ABD to design the steering wheel shows that using IGA can improve the usability of products. In this application each chromosome represents a steering wheel design, and the user evaluates designs based on the affordance criteria such as turn ability and hand rest ability.

Another interesting area is the use of this algorithm in designing the virtual environment of games. In this application, characteristics such as water level, sunlight direction, and cloud motion are encoded in chromosomes. Then, the designs are graphically shown to the user and the user choses the three most desired images among them (Walsh and Gade 2010). In another study, in an image search and retrieval system, the IGA was used to improve the results obtained from the search. The initial results obtained from searching for an image were considered the initial population, and the three features of background, color, and image margin were incorporated into each chromosome. Afterwards, the population was gradually improved (Lai and Chen 2011).

In other studies, the interactive evolutionary algorithm was applied in website appearance design (Oliver et al. 2002), sign sound design (Miki et al. 2006), book cover design (Yu et al. 2014), car console design (Dou et al. 2016), unequal area facility layout design (Garcia-Hernandez et al. 2013), portfolio design considering investor's preferences (Sasaki et al. 2018), and recommender system improvement (Wang et al. 2019).

3.2. 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 a study, the interactive evolutionary algorithm was used to design book covers (Yu et al. 2014), and the k-means clustering method was used to reduce user fatigue. In this method, 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 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 area facility layout problem (Garcia-Hernandez et al. 2013). In this study, similar chromosomes are placed in a cluster, and, after evaluation of cluster centroids by the user, other cluster members are scored based on the cluster centroids scores. To evaluate the algorithm, two 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 (García-Hernández et al. 2015). For this purpose, niching methods (Mahfoud 1995) were used in the IGA. Comparison of this study to previous studies showed a decrease in standard deviation and in the mean number of times required to reach the desired design. In a recent study, IGA was applied to design facial animations on a 3D face model (Hailemariam et al. 2019) and a special kind of elitism was used in this algorithm to control the user fatigue challenge. In this method, each design is scored by the user and then, the sub-parts of the high scored designs (e.g., eyes and head) were transferred to the next generation. The results indicate the acceptable credibility and peculiarity of the produced facial animations.

In 2016, the multistage IGA was developed to tackle user fatigue. The aim was to bring the process of IGA 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 part of the design (e.g. the background design) is shown to the user and it is improved upon through IGA operations so as to achieve user satisfaction of that part. In the next step, different values for 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 a desired design for this part 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. Researchers claimed that this trend gradually increases users' knowledge about the design and diminishes the problem of low user knowledge for evaluating the whole design in the early stages (which lengthens the algorithm execution and increases user fatigue) (Dou et al. 2016). Considering that the multistage IGA is a recent and efficient method, the method proposed in the present paper is compared with the results of the multistage IGA.

In our former study, we developed a method to reduce the user fatigue in interactive design (Sheikhi Darani and Kaedi 2017). In that method, the user preferences are trained using the candidate elimination algorithm during the early stages of user interaction with the IGA. Afterward, in the next stages, the solutions which are incompatible with the user preferences are automatically recognized and a predefined low score is given to them without being scored by the user. In this way, the design process is accelerated and the user fatigue is reduced. In the current study, we present another method to increase the speed of design process. In our recent method, instead of learning the user preferences, we exploit the favourite designs of former users to enrich the population of IGA in order to accelerate the design process for a new user.

4. THE PROPOSED METHOD

As users employ the IGA to achieve their intended design, the system draws valuable knowledge from their interaction and the evolution of designs. This knowledge can be used to speed up the execution of the algorithm by other users. This idea has been adopted in the present study to enable the IGA to achieve the desired design more quickly, and, as a result, to reduce user fatigue, which is a result of the frequent design evaluations.

This study recommends that the designs evolved in the execution of the IGA (for previous users) be stored in a database and then incorporated in an intelligent manner into the process of evolution of GA for future users so as to accelerate algorithm evolution. Here, accelerating the algorithm means helping the algorithm to get the user's favorite design faster and with less user fatigue. Our proposed method helps the algorithm to find the user's desired design more quickly by reducing the number of designs evaluated by the users and decreasing the number of algorithm iterations.

In the IGA used in the present study, first, the initial populations of the algorithm are generated and enriched based on the method explained in Section 4.1. The designs existing in the population are graphically presented to the user so that the user scores them based on his/her preferences. These scores are regarded as the fitness of chromosomes. It is worth mentioning that in our study the optimization objective is to create the design that satisfies the user preferences using IGA. This objective depends on the user interests which are unknown at the beginning of the algorithm execution. Therefore, there is no mathematical equation to be optimized by the IGA. Instead, the optimization objective is formed and modelled gradually based on the user preferences which are inferred from his/her scores to the candidate designs during the execution of algorithm.

Afterwards, by applying elitism, selection, crossover, and mutation operators to the population, a new population of designs is developed. According to the method explained in Section 4.2, this population is enriched on the basis of the designs of other users. This process proceeds and the algorithm moves from one generation to the next generation until the user chooses one of the existing designs in the population as the final design. The pseudocode of proposed algorithm is presented in Figure 2. As it is shown in the pseudocode, the algorithm stops when it reaches a design that satisfies the user enough. Figure 3 presents the flowchart of the proposed method. In what follows, the way to enrich the initial population and other populations during the execution of the algorithm is explained.

```
Procedure Proposed_IGA
Inputs: Database of desired design of farmer users (designs_DB)
Outputs: desired design of current user

Begin
    Generating initial population
    Evaluating the population by user
    While user is not satisfied with the designs do
        Begin
          Generating new population using genetic operators (Selection, Elitism, Crossover, Mutation)
          Finding the most similar user to the current user from designs_DB
          Injecting the desired design of the most similar user to the current population
          Evaluating the population by user
        End
    Inserting user and its desired design to designs_DB
   Return the best design
End
```

Figure 2. The pseudocode of proposed algorithm.

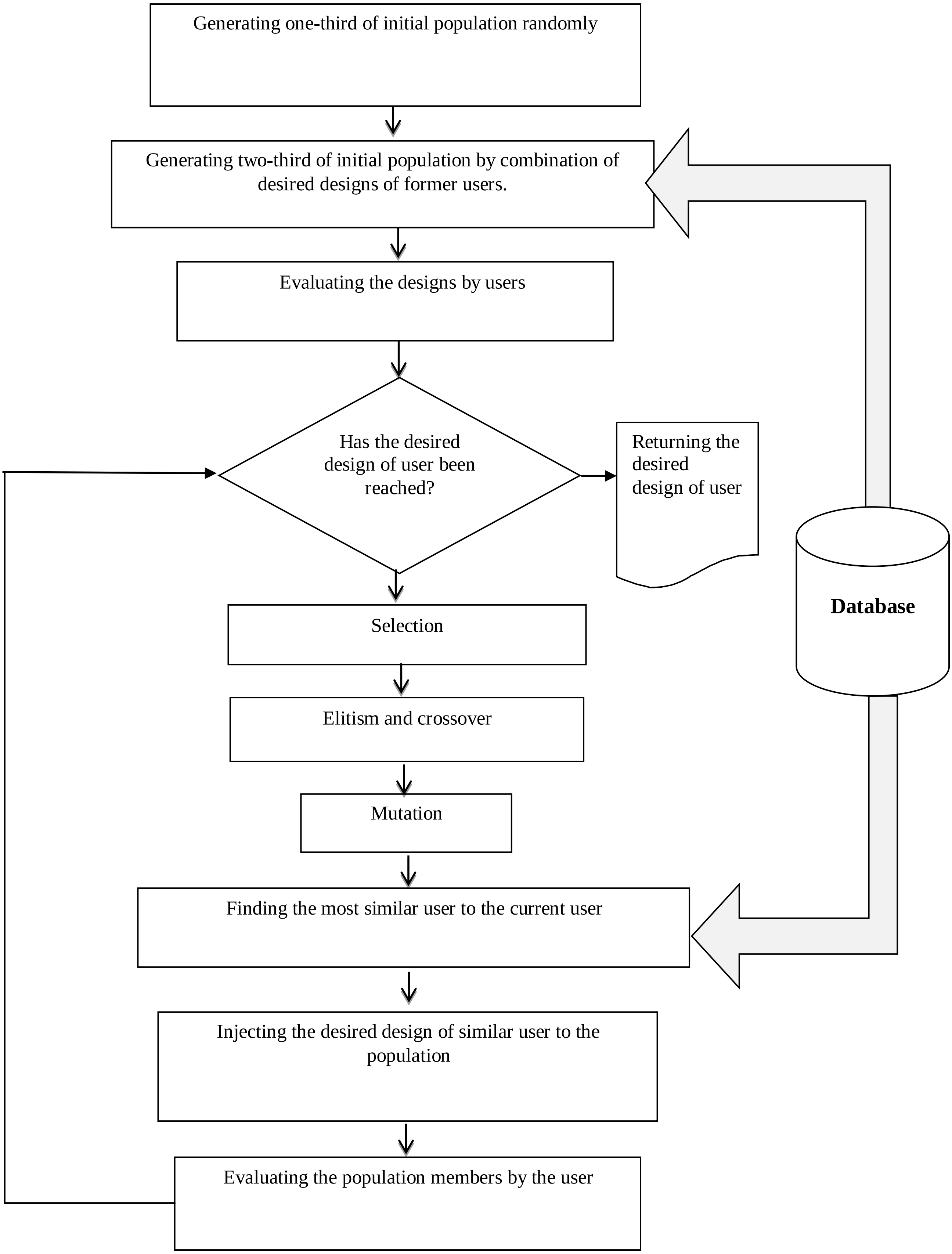

Figure 3. Flowchart of the proposed algorithm.

4.1. Enriching the initial population of the algorithm

In the GA, beginning from an appropriate initial population can speed up the process of reaching the final solution. The initial population is especially important in the IGA, in which the number of population members is limited due to the engagement of the user in chromosome evaluation. Therefore, beginning with an inappropriate initial population can exert a strong negative effect on the process of reaching the user's desired design.

For this reason, in the present study, in order to expedite the algorithm and reduce the number of user evaluations, the entire initial population is not generated randomly; rather, two-thirds of the initial population members are selected randomly from among the desired designs of former users so that by applying crossover operator on them, the features desired by other users reveal in the initial population. The remaining one-third is generated randomly so as to preserve the random nature and variety of the initial population.

4.2. Enriching the GA population during its execution

In the proposed method, besides the initial population, other populations are also enriched during the execution of the algorithm so that the algorithm is directed with higher speed toward achieving the user's desired design, hence reducing the number of necessary evaluations and user fatigue. For this purpose, in each generation, after the yielding of a new population of designs, among the previous system users, the user whose preferences are the most similar to that of the current user is selected and his/her final desired design is added to the current population of the algorithm (Section 4.2.1 demonstrates how to calculate users' similarity of preferences). As a result, when the current user scores the population members, this design probably receives a high score.

If this design is completely desired for the user and the user chooses it as the final desired design, the algorithm is terminated, hence preventing further iteration of the algorithm.

If the design is not selected as the final design, due to the high score it has received, it participates in the yielding of the next generation and passes its features (which are probably preferred by the user) to that generation. This accelerates the algorithm in achieving the user's desired design.

4.2.1. Calculating similarity among users

In the proposed method, the similarity of preferences of two users is computed on the basis of the similarity between their selected designs. For this purpose, for each user such as user u $(1 \leq u \leq U$ in which U is the number of previous users of system) who has used the IGA to reach his/her design, the user's highly scored design in the first generation of the algorithm (called $D_u^{Initial}$) and his/her selected design in the last generation of the algorithm (called D_u^{Final}) are stored in the database. During the execution of the algorithm for the current user (user c), after evaluation of the current designs in the i^{th} generation by the user, the design receiving the highest score in this generation is selected (this design is called the selected design of the current user in the i^{th} generation or D_c^i). Therefore, the similarity of this design to the $D_u^{Initial}$ and D_u^{Final} for each user in the database (i.e.,$1 \le u \le U$) is evaluated. Weighted Euclidean distance is used to compute this similarity. The weighted Euclidean distance between D_c^i and $D_u^{Initial}$ is measured through Eq. (1):

$$dis(D_c^i, D_u^{initial}) = \sqrt{w_1(f_{u,1}^{initial} - f_{c,1}^{i}) + ... + w_j(f_{u,j}^{initial} - f_{c,j}^{i}) + ... + w_{NK}(f_{u,k}^{initial} - f_{c,k}^{i})} \quad (1)$$

where $f_{u,j}^{initial}$ is the normalized value of the j_{th} feature in $D_u^{Initial}$, and $f_{c,j}^{i}$ is the normalized value of the j_{th} feature in D_c^i. K indicates the number of features in each design, and the w_j weight reflects the degree of importance of the j^{th} feature in the computation of Euclidean distance, because not all features of a design are equal in importance. These weights can be regulated on the basis of experts' judgment. In so doing, each expert assigns a rank between 1 and K to each design feature. Rank 1 demonstrates the most important feature from the viewpoint of that expert. The mean of the ranks assigned by the experts to each feature is considered the weight of that feature. In Section 5, some weights given by some experts are presented. In the same way, the weighted Euclidean distance between D_c^i and D_u^{final} is determined through Eq. (2):

$$dis(\mathrm{D}_c^i, D_u^{final}) = \sqrt{w_1(f_{u,1}^{final} - f_{c,1}^{i}) + ... + w_j(f_{u,j}^{final} - f_{c,j}^{i}) + ... + w_K(f_{u,k}^{final} - f_{c,k}^{i})} \quad (2)$$

where $f_{u,j}^{final}$ is the normalized value of the j^{th} feature in D_u^{final}, and the other notations of this equation are defined like the notations of Eq. (1). The similarity between user *c* and user *u* has an inverse relationship with the aforementioned Euclidean distance, and is computed through Eq. (3).

$$(u,c) = \frac{1}{average(dis(D_c^i, D_u^{initial}), dis(D_c^i, D_u^{final}))} = \frac{2}{dis(D_c^i, D_u^{initial}) + dis(D_c^i, D_u^{final})} \quad (3)$$

Equation (3) is computed for all the users (for $1 \le u \le U$), the user with the highest similarity to user *c* in terms of his/her preferences is selected, and the selected final design of this user is added to the i^{th} population.

The reason behind the use of initial and final designs for computing similarity is the likelihood that the user performs the evaluation with sufficient accuracy in these two stages. That is because, in the first stage, the user is at the onset of the design process, and his/her evaluation has not been influenced by factors such as fatigue from interaction with the algorithm. Besides, in the final stage, the user selects his/her final design after observing a number of designs, and he/she is expected to reflect his/her genuine preferences in his/her selection of the final desired design.

5. EVALUATION

The proposed method discussed in Section 4 is evaluated here. For this purpose, the proposed method is applied to design book cover and is compared with IGA and Multi-Stage IGA. Here, the process of converting solutions (i.e., chromosomes) to graphical designs is elaborated, and, afterwards, the parameters setting, evaluation process, and the results are discussed.

5.1. Graphic 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, and text, each of which has their specific characteristics. To take into consideration these components and their characteristics, the study by Yu et al. (Yu et al. 2014) was used. Table 1 presents the components and the number of bits required to demonstrate different values for their characteristics. The *RGB* system is used to describe the color characteristics and 8 bits are considered for each of the main colors (i.e., red, green, and blue). 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 has 4 different values for both of image and text components; hence, 2 bits are considered for this characteristic. The cover design sheet is considered as a grid with two columns and four rows; therefore, the position characteristics has eight values and is demonstrated by 3 bits. Considering the characteristics of each component, the size of each chromosome which describes a cover design is totally equal to 60 bits.

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	-

As mentioned in Section 4.2.1, experts' judgements are used to determine the weight of each feature. For this purpose, three graphic design experts were asked for their opinions, and the mean of the views of these experts was considered as the weight of the features. Table 2 demonstrates the final weight of each feature. As mentioned in Section 4, the features are multiplied to their relevant weights in computing the Euclidean distance.

5.2. Parameters setting

In the IGA, the population size is usually considered a number less than 10. In the multistage IGA (Dou, et al. 2016), which has been compared with the methods proposed in this paper, the population size is considered to be 6. In order to make a fair comparison, the population size in the present paper is set to be 6. In addition, in choosing the rate of genetic operators and their details, the parameters setting of the multistage IGA (Dou, et al. 2016) is followed. Therefore, the single point crossover is used with the crossover rate of 0.8. In addition, the mutation rate is considered to be 0.09. In conducting the crossover operation, the candidate crossover points are predefined so as to prevent crossover along a gene. In mutation, a bit is randomly selected and undergoes a random change. The elitism rate is assumed to be 0.1 to transfer the best chromosomes of one generation to the next generation.

Table 2: Attributes weights.

Attributes	The attributes weights obtained by averaging the experts weightings
Image position on the book cover	14.5
Size of book cover image	14
Background color	13
Size of text #1	11.75
Font of text #1	10.5
Color of text #1	10
Position of text #1	9.25
Size of text #2	7.25
Font of text #2	7
Color on text #2	6.5
Position of text #2	5.75
Size of text #2	3.25
Font of text #2	3
Color on text #2	2
Position of text #2	1.75

5.3. Evaluation procedure

To evaluate the proposed method, 15 users were asked to perform the cover design process separately using the simple IGA, multistage IGA (Dou et al. 2016), and the proposed algorithm. The algorithms have been implemented and applied to solve the cover design problem.

Each user used the algorithms one by one to obtain his/her desired cover designs. All the three algorithms stared from same initial populations. The random parameters in the algorithm cause variety in results. To reduce the effect of randomness, this process repeated three times for different initial populations and the averaged results are reported.

Figures 4, 5, and 6 illustrate a sample of populations in three consecutive generations for the issue of book cover design.

Figure 4. A demonstration of the first generation of chromosomes generated in the proposed algorithm.

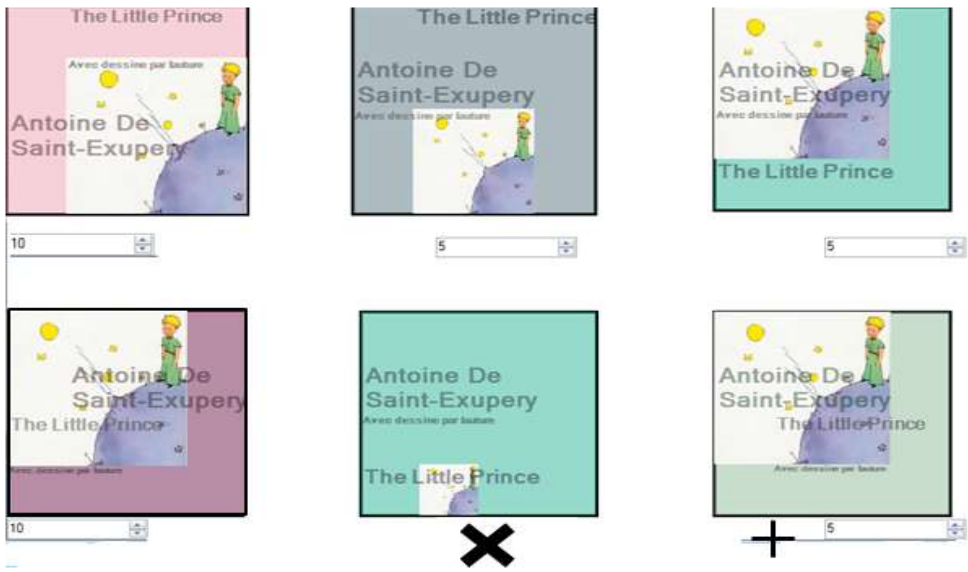

Figure 5. A demonstration of the second generation of chromosomes generated in the proposed algorithm.

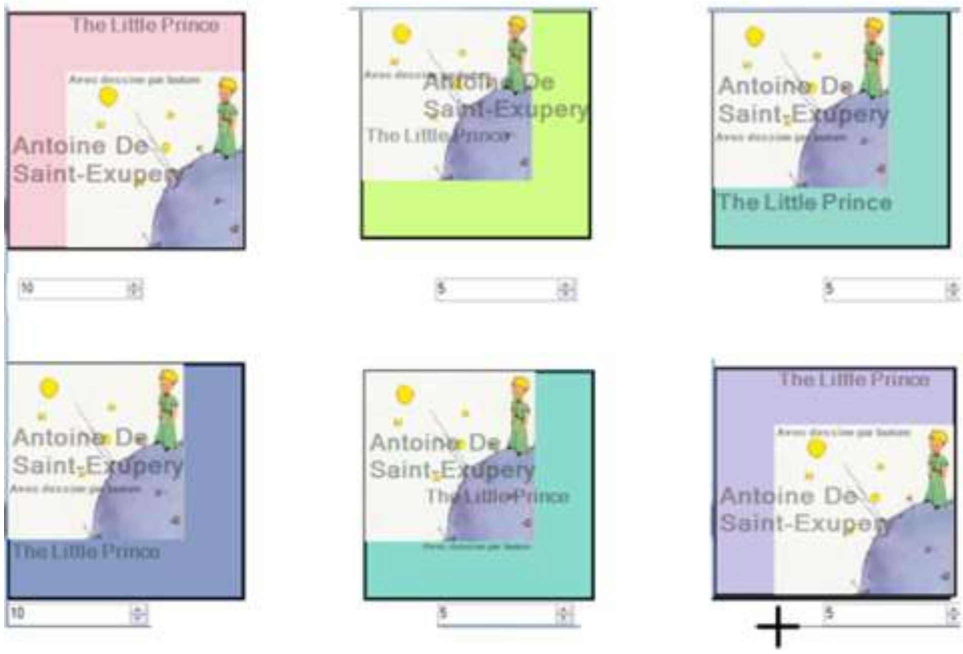

Figure 6. A demonstration of the third generation of chromosomes generated in the proposed algorithm.

The two designs existing in the first generation (Figure 4), which are marked by circle signs, have been generated at random. In each generation, the design marked by plus sign is the desired design of the most similar user to the current user. This design has been retrieved and added to that population.

Figure 7 represents the average design scores for three consecutive generations presented in Figures 4, 5, and 6. Although due to the low number of generations, it is not possible to discuss the convergence of the algorithm, but it is clear in the figure that the average of design scores moves towards convergence.

The mean number of generations, the mean number of individuals 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 Figures 8, 9, and 10.

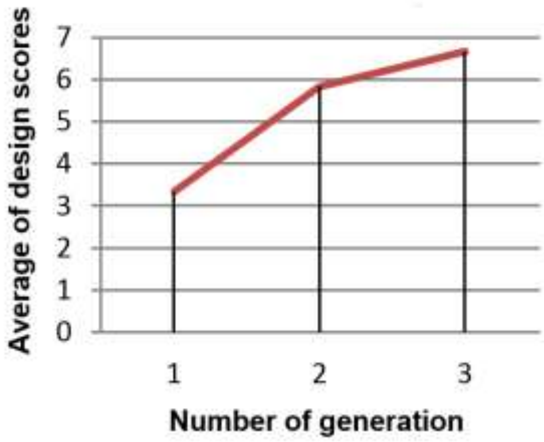

Figure 7. The average design scores for consecutive generations presented in Figures 4, 5, and 6.

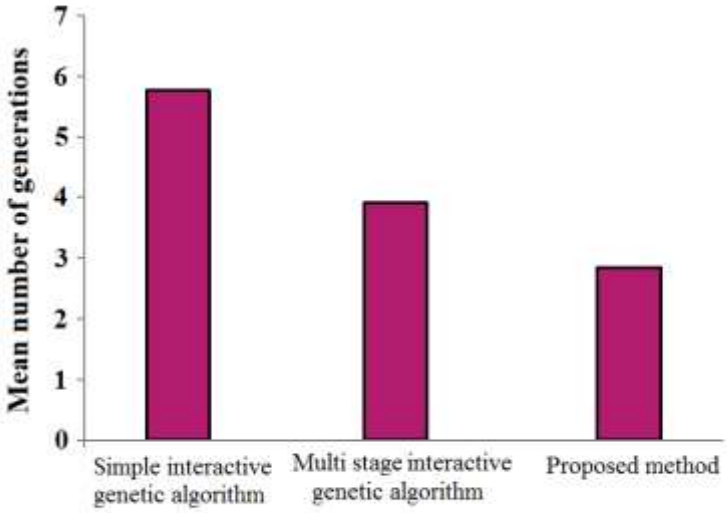

Figure 8. The “mean number of generations” in the proposed algorithm compared to simple IGA and multi stage IGA for book cover design problem.

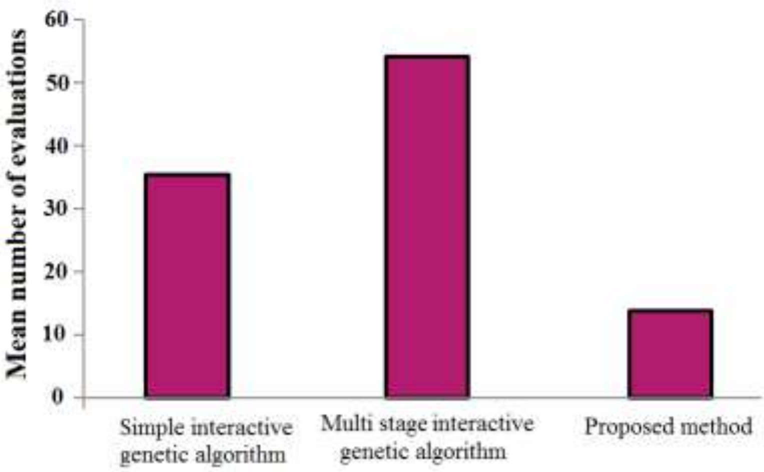

Figure 9. The “mean number of evaluations” in the proposed algorithm compared to simple IGA and multi stage IGA for book cover design problem.

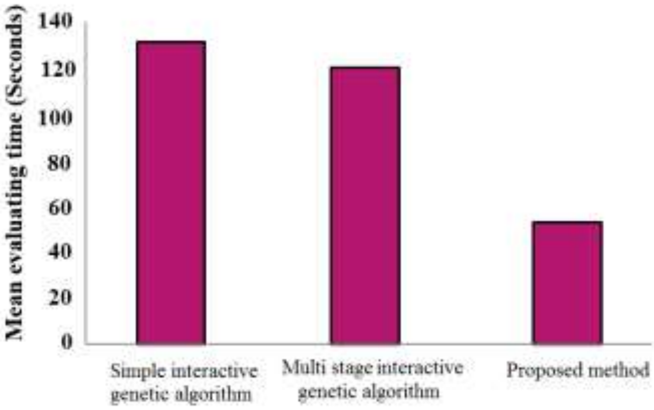

Figure 10. The “mean evaluating time” of the proposed algorithm compared to simple IGA and multi stage IGA for book cover design problem.

As it is shown in the charts, the proposed method has reduced both the number of generations and the number of design evaluations made by users. As a result, the user reaches his/her favorite design in a shorter time in comparison to other two methods. So, the design process is accelerated and the user fatigue is reduced by our proposed method.

In order to measure the degree of satisfaction in addition to these criteria, the After Scenario Questionnaire (ASQ) (Lewis 1995) 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 Figures 11 and 12) 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 inversed 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.

5.4. Evaluation analysis

As shown in the figures, the results are indicative of the 58% decrease in the evaluating time, the 47% decrease in the number of generations, and 52% decrease in the number of design evaluations, compared with the simple IGA. According to this results, it can be concluded that our proposed method achieves the user’s desired design in a shorter time and the user fatigue has been reduced through utilizing the preferred designs of other users in our proposed method.

Besides, in comparison with the multistage IGA, the proposed method has lead a 58% improvement in the evaluating time, a 22% decrease in the number of generation, and a 70% reduction in the number of design evaluations.

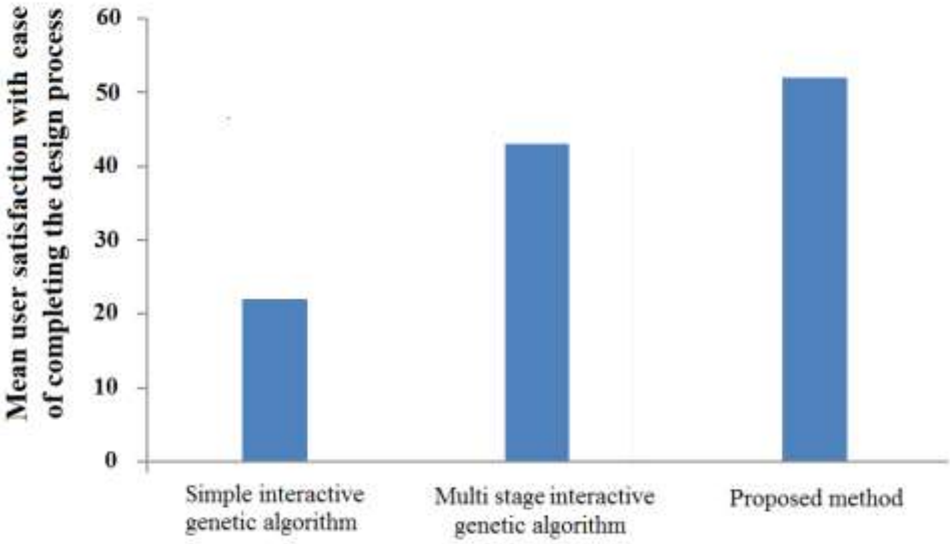

Figure 11. Mean user satisfaction with ease of completing the design process using either algorithm.

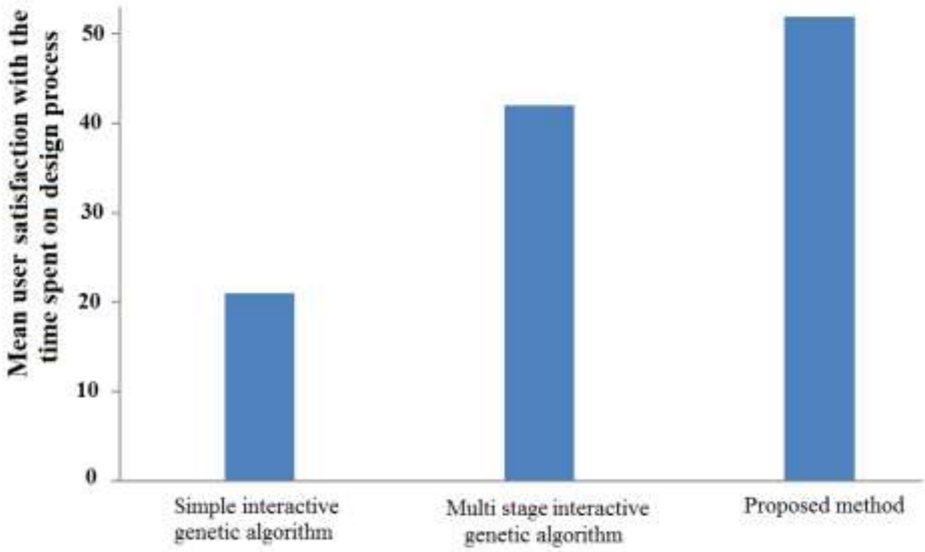

Figure 12. Mean user satisfaction with the time spent on design process using either algorithm.

The great improvement in the number of evaluations obtained by our proposed method compared to the multistage interactive genetic method is due to the fact that in some cases, in the proposed method the user has found his/her desired design in the initial population. In contrast, in the multistage interactive genetic method, the number of required evaluations for achieving the desired design is at least equal to the multiplication of the number of design components by the population size. That is, because each design includes 5 components, even if the user finds his/her desired component in the first iteration of every stage and no further iteration is needed, 5 stages are required to complete the design, each of which includes evaluation of all population chromosomes. Hence, this great difference in the number of evaluations is natural.

By a step-by-step completion of the design, the multistage IGA enables the user to proceeds with the process in a more disciplined fashion and to concentrate on a single component in each stage. On the other hand, however, this issue raises the possibility that the user fails to generate a desired design because of selecting an inappropriate component at the beginning stages of the design process without observing its combination with other components.

Figures 10 and 11 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 they spent on designing are the highest in the proposed algorithm and the lowest in the simple IGA. In the proposed algorithm, the initial population is enriched using the desired designs of similar users. For this reason, the current user faces more appropriate designs and 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.

6. CONCLUSION

In this paper, a method was proposed to reduce user fatigue in IGA. In this method, the algorithm populations are enriched by identifying similar users to the current user and applying their desired designs, enabling the user to reach his/her desired design with fewer iterations of the algorithm. The results obtained by testing this method for book cover design indicated a decrease in evaluating time, a decrease in the number of algorithm generations, and a decrease in the number of evaluations in comparison with the simple IGA and the multistage IGA.

In the proposed algorithm, the most similar user to the current user is selected among all the users who have previously worked with the system. As the number of users increases, finding similar users can increase the algorithm execution time. It is recommended that, for accelerating this process, the users be clustered in advance and the search for the similar user be carried out only in the clusters to which the current user belongs. Clustering can be carried out on the basis of user characteristics (Yusefi Hafshejani et al. 2018; Kazeminia et al. 2019) or design theme. The recently developed optimization algorithms presented in the literature (Goli et al. 2018; Shams et al. 2017; Joelianto and Prakoso 2017; Kaedi 2017) can also be examined to optimize the designs. 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 on a larger number of users.

7. REFERENCES

Alvarez Gil, R., Johanyák, Z., and Kovács, T., 2018, *Surrogate Model Based Optimization of Traffic Lights Cycles and Green Period Ratios Using Microscopic Simulation and Fuzzy Rule Interpolation*, International Journal of Artificial Intelligence, 16, 20-40.

Bloch, P.H., 1995, *Seeking the Ideal Form: Product Design and Consumer Response*, Journal of Marketing, 59, 16-29.

Brintrup, A.M., Ramsden, J., and Tiwari, A., 2007, *An Interactive Genetic Algorithm-Based Framework for Handling Qualitative Criteria in Design Optimization*, Computers in Industry, 58, 279-91.

Crilly, N., Moultrie, J., and Clarkson, P.j., 2004, *Seeing Things: Consumer Response to the Visual Domain in Product Design*, Design Studies, 25, 547-77.

Dou, R., Zong, C., and Li, M., 2016, *An Interactive Genetic Algorithm with the Interval Arithmetic Based on Hesitation and its Application to Achieve Customer Collaborative Product Configuration Design*, Applied Soft Computing, 38, 384-94.

Dou, R., Zong, C., and Nan, G., 2016, *Multi-Stage Interactive Genetic Algorithm for Collaborative Product Customization*, Knowledge-Based Systems, 92, 43-54.

Duong, Q.K., Chovanec, L., Hubinský, P., Vozák, D., Matuga, M., and Varga, P., 2017, *Comparison of Input Shaper Based on Genetic Algorithms with Analytical Approach*, International journal of artificial intelligence, 15, 21-32.

Garcia-Hernandez, L., Pierreval, H., Salas-Morera, L., and Arauzo-Azofra., A., 2013, *Handling Qualitative Aspects in Unequal Area Facility Layout Problem: An Interactive Genetic Algorithm*, Applied Soft Computing, 13, 1718-27.

García-Hernández, L., Palomo-Romero, J.M., Salas-Morera, L., Arauzo-Azofra, A., and Pierreval, H., 2015, *A Novel Hybrid Evolutionary Approach for Capturing Decision Maker Knowledge into the Unequal Area Facility Layout Problem* , Expert Systems with Applications, 42, 4697-708.

Goli, A., Aazami, A., and Jabbarzadeh, A., 2018, *Accelerated Cuckoo Optimization Algorithm for Capacitated Vehicle Routing Problem in Competitive Conditions*, International Journal of Artificial Intelligence, 16, 88-112.

Gupta, T., 2015, *Implementation of an Optimization Technique: Genetic Algorithm*, Advanced Research in Computer Engineering & Technology (IJARCET), 4, 4359-64.

Hailemariam, M., Goertzel, B., and Yohannes, T., 2019, *Evolving 3D Facial Expressions Using Interactive Genetic Algorithms,* In International Conference on Advances of Science and Technology, 492-502, Springer, Cham.

Joelianto, E., and Prakoso, B., 2017, *Stingless Bee Foraging Behaviour Algorithm for Optimisation*, International Journal of Artificial Intelligence, 15, 1-20.

Kaedi, M., 2017, *Fractal-based Algorithm: A New Metaheuristic Method for Continuous Optimization*, International Journal of Artificial Intelligence, 15, 76-92.

Kazeminia, A., Kaedi, M., and Ganji, B., 2019, *Personality-Based Personalization of Online Store Features Using Genetic Programming: Analysis and Experiment*, Journal of Theoretical and Applied Electronic Commerce Research, 14, 16-29.

Kim, H., and Cho, S., 2000, *Application of Interactive Genetic Algorithm to Fashion Design*, Engineering Applications of Artificial Intelligence, 13, 635-44.

Lai, C.C., and Chen, Y.C., 2011, *A User-Oriented Image Retrieval System Based on Interactive Genetic Algorithm*, IEEE Transactions on Instrumentation and Measurement, 60, 3318-25.

Lee, J.H., and Chang, M.L., 2010, *Stimulating Designers' Creativity Based on a Creative Evolutionary System and Collective Intelligence in Product Design*, International Journal of Industrial Ergonomics, 40, 295-305.

Lewis, J.R., 1995, *IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use*, International Journal of Human-Computer Interaction, 7, 57-78.

Mahfoud, S.W., 1995, *Niching Methods for Genetic Algorithms*, 21-60, University of Illinois at Urbana-Champaign.

Mata, I., Fadel, G., Garland, A., and Zanker, W., 2018, *Affordance Based Interactive Genetic Algorithm (ABIGA)*, Design Science, 4, e5.

Miki, M., Orita, H., Wake, S.H., and Hiroyasu, T., 2006, *Design of Sign Sounds using an Interactive Genetic Algorithm*, In 2006 IEEE International Conference on Systems, Man and Cybernetics, 3486-90.

Mitchell, T.M., 1997, *Machine Learning*, 249-255, McGraw-Hill Science, Boston.

Oliver, A., Monmarché, N., and Venturini, G., 2002, *Interactive Design of Web Sites with a Genetic Algorithm*, In ICWI2002, , 355-362, Lisbon, Portugal.

Ono, S., Maeda, H., Sakimoto, K., and Nakayama, S., 2014, *User-System Cooperative Evolutionary Computation for Both Quantitative and Qualitative Objective Optimization in Image Processing Filter Design*, Applied Soft Computing, 15, 203-18.

Osaba, E., Ser, J. D., Sadollah, A., Bilbao, M. N., and Camacho, D., 2018, *A Discrete Water Cycle Algorithm for Solving the Symmetric and Asymmetric Traveling Salesman Problem*, *Applied Soft Computing*, *71*, 277-290.

Precup, R., David, R., 2019, *Nature-Inspired Optimization Algorithms for Fuzzy Controlled Servo Systems*, Butterworth-Heinemann, Oxford, United Kingdom.

Purcaru, C., Precup, R., Iercan, D., Fedorovici, L., David, R., Dragan, F., 2013, *Optimal Robot Path Planning Using Gravitational Search Algorithm*, International Journal of Artificial Intelligence, 10, 1-20.

Quiroz, J., Dascalu, S., 2016, *Design and Implementation of a Procedural Content Generation Web Application for Vertex Shaders*. arXiv: 1608.05231[cs.GR], https://arxiv.org/abs/1608.05231.

Rahmi, A., Mahmudy, W.F., and Sarwani, M.Z., 2020, *Genetic Algorithms for Optimization of Multi-Level Product Distribution*, International Journal of Artificial Intelligence, 18, 135-147.

Sasaki, M., Laamrani, A., and Yamashiro, M., 2018, *An Interactive Genetic Algorithm for Portfolio Optimization Considering the Decision Maker's Preference*, Journal of Information and Optimization Sciences, 39, 989-1008.

Shams, M., Rashedi, E., Dashti, S.M., and Hakimi, A., 2017, *Ideal Gas Optimization Algorithm*, International Journal of Artificial Intelligence, 15(2), 116-130.

Sheikhi Darani, Z., and Kaedi, M., 2017, *Improving the Interactive Genetic Algorithm for Customer-Centric Product Design by Automatically Scoring the Unfavorable Designs*, Human-centric Computing and Information Sciences, 7, 38.

Takagi, H., 2001, *Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation*, Proceedings of the IEEE, 89, 1275-96.

Walsh, P., and Gade, P., 2010, *Terrain Generation Using An Interactive Genetic Algorithm*, IEEE Congress on Evolutionary Computation, Barcelona, 1-7.

Wang, P., Hong, C., and Lin, M., 2019, *Interactive Genetic Algorithm Joining Recommender System*, Intelligent Information and Database Systems, ACIIDS 2019, Lecture Notes in Computer Science, vol 11432. Springer, Cham, 40-48.

Yu, F., Li, Y., Kuang, L., and Li, Z., 2014, *Developing Support System for Cover Design with Interactive Evolutionary Computation*, The 26th Chinese Control and Decision Conference (2014 CCDC) , 408-13, Changsha.

Yusefi Hafshejani, Z., Kaedi, M., and Fatemi, A., 2018, *Improving Sparsity and New User Problems in Collaborative Filtering by Clustering the Personality Factors*, Electronic Commerce Research, 18, 813–836.