*Journal of Universal Computer Science, vol. 15, no. 13 (2009), 2446-2462
submitted: 31/10/08, accepted: 13/6/09, appeared: 1/7/09 © J.UCS*

Interactive Genetic Algorithms with Individual Fitness not Assigned by Human

Dunwei Gong
(China University of Mining and Technology, Xuzhou, P.R.China
dwgong@vip.163.com)

Xin Yao
(University of Birmingham, Birmingham, United Kingdom
X.Yao@cs.bham.ac.uk)

Jie Yuan
(China University of Mining and Technology, Xuzhou, P.R.China
sweet_joy@163.com)

Abstract: Interactive genetic algorithms (IGAs) are effective methods to solve optimization problems with implicit or fuzzy indices. But human fatigue problem, resulting from evaluation on individuals and assignment of their fitness, is very important and hard to solve in IGAs. Aiming at solving the above problem, an interactive genetic algorithm with an individual fitness not assigned by human is proposed in this paper. Instead of assigning an individual fitness directly, we record time to choose an individual from a population as a satisfactory or unsatisfactory one according to sensitiveness to it, and its fitness is automatically calculated by a transformation from time space to fitness space. Then subsequent genetic operation is performed based on this fitness, and offspring is generated. We apply this algorithm to fashion design, and the experimental results validate its efficiency.

Key Words: Optimization, genetic algorithm, interactive genetic algorithm, human fatigue, individual fitness
Category: I.2.8, G.1.6, H.1.2

1 Introduction

Optimization problems are very common in real-world applications, such as traveling salesman problem (TSP) [Lin and Kernighan 73], job-shop scheduling [Adams et al. 88], product design [Li and Azarm 00], and so on. For an optimization problem whose objective functions are continuously differentiable and whose scale is small or medium, some traditional optimization methods, such as Newton method [Xie et al. 03], are suitable to solve it. Whereas for an optimization problem whose objective functions are not differentiable, or even not continuous, or for one whose objective functions are differentiable but whose scale is very large, many traditional optimization methods are no longer applicable.

Genetic algorithms (GAs), proposed in early 1970s, are a kind of globally stochastic optimization methods inspired from nature evolution [Holland 75].

Since GAs do not require continuous and differentiable objective functions of an optimization problem, and can effectively find satisfactory solutions for a large scale problem, they have gained broad attention in the optimization community and fruitful achievements have been obtained [Gong and Pan 03, Wiegand 04, Deb et al. 02].

Although GAs do not require continuous and differentiable objective functions, they do require well-defined objective functions in order to calculate an individual fitness. But it is difficult for many complicated optimization problems to have one or several well-defined objective functions because of their implicit or fuzzy indices. Therefore GAs are not applicable to such optimization problems.

Interactive genetic algorithms (IGAs), proposed in middle 1980s, are effective methods to solve optimization problems with implicit or fuzzy indices [Dawkins 86]. They combine traditional evolution mechanism with human's intelligent evaluation, and human assigns an individual fitness rather than a function that is difficult or even impossible to express explicitly. Up to now, they have been successfully applied in many fields, such as fashion design [Kim et al. 00], face identification [Caldwell et al. 91], music composition [Tokui and Iba 00], hearing aid fitting [Takagi and Ohsaki 07], and so on.

The obvious characteristic of IGAs, compared with GAs, is that human assigns individual fitness. Human compares among individuals in the same generation and assigns fitness based on their phenotypes through human-computer interface. Frequent interaction of human-computer results in human fatigue. Therefore IGAs often have small population size and small generations [Takagi 01], which influences the algorithms' performances to some degree and restricts their applications in complicated optimization problems. Accordingly, how to alleviate human fatigue becomes one of important problems in IGAs.

Since human fatigue results from human's evaluation on individuals and assignment of their fitness, in order to alleviate human fatigue, a possible alternative is to change the approach to evaluate an individual and assign its fitness. The goal of this paper is to alleviate human fatigue by adopting an appropriate approach to evaluate an individual and assign its fitness. As we all know, human has different sensitiveness to different individuals in the same generation. Based on this, if we record time spent by human in choosing an individual from a population as a satisfactory or unsatisfactory one through the evolutionary system, and adopt a transformation to establish the relationship between time and the individual fitness, we can obtain an individual fitness without direct assignment by human. Therefore we can reduce time to human-computer interaction, and omit time spent in assigning an individual fitness by human, resulting in alleviating human fatigue greatly. For human does not directly assign an individual fitness, we call the algorithm an interactive genetic algorithm with an individual fitness not assigned by human (IGA-IFNAH).

In the next section, we will review some methods to alleviate human fatigue. The emphasis of this paper is in section 3, in which we will propose ideas of IGA-IFNAH, present strategies to obtain an individual fitness, describe steps of IGA-IFNAH, and give some further explanations. We will provide its applications in fashion design and some experimental results in section 4. Finally, we will draw some conclusions and point out our future work in section 5.

2 Related Works

Since human fatigue problem is important in IGAs, numerous researches have focused on how to alleviate human fatigue. Up to now, there have been many approaches to deal with it.

The first one is to adopt an appropriate value to express an individual fitness. For example, Takagi *et al.* proposed a fitness assignment method which combines a continuous fitness with a discrete one [Takagi and Ohya 96]. Based on uncertain or fuzzy cognition of human on an individual, Gong *et al.* adopted an interval number to express an individual fitness [Gong and Guo 07], hence alleviating the load resulting from evaluating an individual. It is easy to understand that different expressions of an individual fitness require different interfaces through which an individual is evaluated. A friendly interface is attractive and human is willing to use.

The second one is to use some surrogate-assisted models to evaluate a part of or even all individuals in some generations, hence the number of individuals evaluated by human decreases. For example, Sugimoto *et al.* estimated an individual fitness using fuzzy logic based on the distance and the angle between the evaluated individual and the optima being found [Sugimoto and Yoneyama 01]. Biles and Zhou *et al.* adopted neural networks (NNs) to learn human's intelligent evaluation on an individual, and the number of individuals evaluated by human decreases by use of neural networks evaluating individuals in an appropriate time [Biles et al. 96, Zhou et al. 05]. In order to improve learning precision and reduce network complexity, Gong *et al.* adopted multiple surrogate-assisted models [Gong et al. 07], in which a simple surrogate-assisted model only learned human's evaluation on a part of the search space. Wang *et al.* transformed the evaluation on an individual assigned by human into an absolute rating fitness and adopted it to train a support vector machine (SVM) to evaluate individuals [Wang et al. 06]. In any case, looking for effective models which can learn human's preference is very important. Otherwise, these models may mislead to the evolution of a population, and unsatisfactory solutions may be mistaken as satisfactory ones unavoidably.

The third one is to accelerate population convergence [Hayashida et al. 02]. As we all know, on condition of constant population size, the less the number of evolutionary generations of a population is, the less the number of individuals evaluated by human. Gong *et al*. reduced the valid search space by using knowledge acquired during the evolution of a population, and speeded up population convergence [Gong et al. 05a]. Another way is to make use of the evolution results of other populations, and the evolution of a population continues based on them, hence the number of evolutionary generation decreases to some degree [Gong et al. 05b].

The fourth one is to indirectly obtain an individual fitness through some devices. For example, Pallez *et al*. recently applied an eye-tracking device to measure human preference, and then obtained an individual fitness by a transformation from some parameters [Pallez et al. 07]. Some simulation results show that it is efficient in alleviating human fatigue.

The common character of the above methods is that an individual fitness is required. If we do not know it, the evolution of a population will not continue. But there are other methods that do not require it. For example, Llorà *et al*. directly chose the superior individual in tournament selection with size two based on his preference, and did not care their fitness [Llorà et al. 05]. Lewis et al. also directly chose good individuals in the current generation as parents in the next generation though interface, and did not care their fitness [Lewis and Ruston 05]. Although the evolution of a population goes on, we cannot express the dominance relationship among individuals since we do not know their fitness.

In fact, some efficient genetic operators, such as adaptive crossover and mutation operator [Srinivas and Patnaik 94], niche selection [Deb and Goldberg 89], and so on, often require the dominance relationship among individuals, and then adopt appropriate genetic operators or (and) genetic control parameters based on it in the evolution of a population. If we do not know the relationship, we cannot use these operators or (and) genetic control parameters.

In short, it is necessary for IGAs to obtain an individual fitness. A good interactive genetic algorithm should acquire and make full use of an individual fitness on condition of alleviating human fatigue.

3 Interactive Genetic Algorithm with Individual Fitness not Assigned by Human

3.1 Ideas of the Algorithm

IGAs produce satisfactory solutions through human-computer interface and evolve a population from generation to generation. We consider a population in some generation here. In general, human is very sensitive to the most satisfactory individual[Anderson 05], and will only spend a very short time in choosing it from the population. Similarly, human is also very sensitive to the most unsatisfactory individual, and will only spend a very short time in choosing it from the population, too. But for other individuals, human is not sensitive to them, and will spend more time in choosing them.

It is easy to record the time when the evolutionary system displays a population in some generation to human. In order to calculate time to choose an individual, we set up two sets (or folders), namely a satisfactory set and an unsatisfactory set. The satisfactory set only stores satisfactory individuals, and the unsatisfactory set only stores unsatisfactory individuals. The evolutionary system also automatically records the time when these individuals are stored in the two sets. For an individual, the difference between the time when it is stored in a set and the one when the system displays the population is time to choose it.

It is obvious that for an individual in the satisfactory set, the more human prefers it, the less time spent by human in choosing, hence the greater its fitness should be. Similarly, for an individual in the unsatisfactory set, the more human dislikes it, the less time spent by human in choosing, hence the smaller its fitness should be. Based on these, we can obtain an individual fitness through a map from time space to fitness space.

3.2 Strategies to Obtain Individual Fitness

Let $x(t)$ be a population in the t-th generation, $x_i(t)$ be an individual of it, and $T(x(t))$ be the time when the evolutionary system displays $x(t)$ to human. Let $S_{\mathrm{s}}(t)$ and $S_{\mathrm{u}}(t)$ be two sets that consist of satisfactory individuals and unsatisfactory individuals in $x(t)$ respectively. Denote the time when $x_i(t)$ is stored in $S_{\mathrm{s}}(t)$ or $S_{\mathrm{u}}(t)$ as $T(x_i(t))$. It is easy to obtain that human spends $T(x_i(t)) - T(x(t))$ in choosing $x_i(t)$ as a satisfactory or an unsatisfactory individual.

We need another scalar related with time in order to automatically calculate $x_i(t)$'s fitness, and denote it as $\alpha(x_i(t))$, which satisfies that the better $x_i(t)$ is, the greater the scalar. Therefore, for $x_i(t)$ in $S_{\mathrm{u}}(t)$, a candidate of $\alpha(x_i(t))$ is:

$$\alpha(x_i(t)) = T(x_i(t)) - \min_{x_j(t)\in S_{\mathrm{u}}(t)} T(x_j(t)) \tag{1}$$

And for $x_i(t)$ in $S_{\mathrm{s}}(t)$, a candidate of $\alpha(x_i(t))$ is:

$$\alpha(x_i(t)) = \max_{x_k(t)\in S_{\mathrm{s}}(t)} T(x_k(t)) - T(x_i(t)) + \max_{x_j(t)\in S_{\mathrm{u}}(t)} T(x_j(t)) - \min_{x_j(t)\in S_{\mathrm{u}}(t)} T(x_j(t)) \tag{2}$$

If we want to scale an individual fitness in range of $[f_{\min}, f_{\max}]$, then we present the following $x_i(t)$'s fitness:

$$f(x_i(t)) = f_{\min} + (f_{\max} - f_{\min}) \cdot \frac{\alpha(x_i(t))}{\beta(t)} \tag{3}$$

where

$$\beta(t) = \max_{x_k(t)\in S_{\mathrm{s}}(t)} T(x_k(t)) - \min_{x_k(t)\in S_{\mathrm{s}}(t)} T(x_k(t)) + \max_{x_j(t)\in S_{\mathrm{u}}(t)} T(x_j(t)) - \min_{x_j(t)\in S_{\mathrm{u}}(t)} T(x_j(t)) \quad (4)$$

It can be seen from (1) and (3) that for the most unsatisfactory individual $x_i(t)$ in $x(t)$, $\alpha(x_i(t))$ is zero, hence its fitness $f(x_i(t))$ is the smallest, namely $f_{\min}$. Similarly, it can also be seen from (2), (3) and (4) that for the most satisfactory individual $x_i(t)$ in $x(t)$,$\frac{\alpha(x_i(t))}{\beta(t)} = 1$, therefore its fitness $f(x_i(t))$ is the greatest, namely $f_{\max}$.

It is easy to deduce from (1) and (2) that the best individual in $S_{\mathrm{u}}(t)$ has the same value of $\alpha(\cdot)$ as the worst one in $S_{\mathrm{s}}(t)$. Therefore they have equal fitness, which implies that the best individual in the unsatisfactory set is as good as the worst individual in the satisfactory set, that is to say, human is difficult to make a clear decision. But in the two cases, the best individual in the unsatisfactory set and the worst individual in the satisfactory set should not have the same fitness, when there is only one kind of satisfactory individuals or only one kind of unsatisfactory individuals. In these two cases, human is able to make a clear decision. Therefore we should modify some formulas to make sure that the fitness of satisfactory individual(s) is bigger than that of unsatisfactory one(s).

If there is only one kind of individuals in $S_{\mathrm{s}}(t)$ or only one kind of individuals in $S_{\mathrm{u}}(t)$, for $x_i(t)$ in $S_{\mathrm{s}}(t)$, a candidate of $\alpha(x_i(t))$ is:

$$\alpha'(x_i(t)) = \max_{x_k(t)\in S_{\mathrm{s}}(t)} T(x_k(t)) - T(x_i(t)) + \max_{x_j(t)\in S_{\mathrm{u}}(t)} T(x_j(t)) - \min_{x_j(t)\in S_{\mathrm{u}}(t)} T(x_j(t)) + \varepsilon \quad (5)$$

Similarly, a candidate of $\beta(t)$ is:

$$\beta'(t) = \max_{x_k(t)\in S_{\mathrm{s}}(t)} T(x_k(t)) - \min_{x_k(t)\in S_{\mathrm{s}}(t)} T(x_k(t)) + \max_{x_j(t)\in S_{\mathrm{u}}(t)} T(x_j(t)) - \min_{x_j(t)\in S_{\mathrm{u}}(t)} T(x_j(t)) + \varepsilon \quad (6)$$

Where ε is a small constant which is set in advance.

Another interesting phenomenon is that $T(x(t))$ is absent in formula (1) to (4), that is to say, in order to calculate an individual fitness, we do not require the time when the evolutionary system displays a population to human. We only require the time when an individual is stored in the satisfactory or the unsatisfactory set, which can be automatically done by the evolutionary system.

Sometimes there are more than one individual with the same phenotype in the same generation, and they should have the same time to be stored in the satisfactory or the unsatisfactory set. To do this, we add an operation that when one of them is stored in a set, the others are also simultaneously stored in the same set by the evolutionary system.

3.3 Steps of IGA-IFNAH

The steps of the proposed algorithm are described as follows.

1. Set the values of evolutionary control parameters in the algorithm. Let $t = 0$, and initialize a population $x(t)$.
2. Decode and display $x(t)$ to human, let $S_{\mathrm{s}}(t) = S_{\mathrm{u}}(t) = \emptyset, i = 1$.
3. Check whether i is greater than $|x(t)|$ or not, if yes, go to step 5.
4. Investigate whether $x_i(t)$ is a satisfactory individual or not, if yes, then let $S_{\mathrm{s}}(t) \leftarrow S_{\mathrm{s}}(t) \cup \{x_i(t)\}$, otherwise let $S_{\mathrm{u}}(t) \leftarrow S_{\mathrm{u}}(t) \cup \{x_i(t)\}$. Record $T(x_i(t))$, let $i = i + 1$, and go to step 3.
5. Calculate $\max\limits_{x_k(t) \in S_{\mathrm{s}}(t)} T(x_k(t))$, $\min\limits_{x_k(t) \in S_{\mathrm{s}}(t)} T(x_k(t))$, $\max\limits_{x_j(t) \in S_{\mathrm{u}}(t)} T(x_j(t))$, $\min\limits_{x_j(t) \in S_{\mathrm{u}}(t)} T(x_j(t))$, $\alpha(x_i(t))$, $\beta(t)$ and $f(x_i(t))$, $i = 1, 2, \ldots, |x(t)|$.
6. Check whether the algorithm stops or not, if yes, then go to step 8.
7. Perform genetic operators and generate offspring. Let $t = t + 1$, and go to step 2.
8. Output the most satisfactory solution and stop the algorithm.

In the above steps, $\emptyset$ is a null set, and $|x(t)|$ indicates the population size of $x(t)$.

3.4 Further Explanations

An obvious character of the proposed algorithm, compared with early IGAs, is that human **does not** assign an individual fitness. What human does is to choose an individual from the population and store it **in an appropriate set in appropriate order** according to his/her preference. Then the evolutionary system automatically calculates its fitness. Therefore human fatigue resulting from evaluating an individual is greatly alleviated.

In addition, the evolutionary system automatically calculates an individual fitness, not just a dominance relationship among different individuals. Therefore, not only traditional genetic operators, such as tournament selection, one-point crossover and one-point mutation, but also many efficient genetic operations proposed by many researchers in recent years, such as niche selection, crossover and mutation with adaptive rates, can be applied to the proposed algorithm, which implies the efficient performance of IGA-IFNAH.

The key of the proposed algorithm is to determine appropriate order of a chosen individual, which is not difficult on condition that we obey the cognitive law. If we violate the general cognitive law and choose an individual in stochastic order, the algorithm will not work. But it does not mean that we cannot do any other things during running the algorithm. In fact, we can have a rest or answer a phone, or have a cup of coffee, and so on, which does not affect the performance of the algorithm on condition that we choose an individual in appropriate order before and after the interrupt. After all, the algorithm does not require the absolute time but the relative one, and we only compare the individuals in the same generation.

If the individuals in a generation are all very good, then the time to make a decision will be long, these individuals fitness will be small although they may be stored in the satisfactory set. On the contrary, if the individuals in a generation are all very bad, then the time to make a decision will be also long, some individuals fitness will be great although they are stored in the unsatisfactory set. The above two cases indicate an individual fitness is not consistent but changeable with generations, which will not affect the performance of the algorithm for the same reason as the above.

4 Applications in Fashion Design

4.1 Backgrounds

Fashion design is a very popular vocation for everyone likes to wear satisfactory fashion but few can design a satisfactory one. In fact, fashion design is a very complicated process and often completed by designers who have been trained systematically. Although there are some softwares available for fashion design, they are often too special for an ordinary person to use. With the development of society pursuing personalities becomes a fad. That is to say, human often likes to wear fashion with some personalities. It is very useful if there is a fashion design system for an ordinary person to design his or her satisfactory fashion.

We hope to establish a fashion design system for an ordinary person to generate a suit by combining all parts from different databases. That is to say, parts of suit are stored in databases in advance. What human does is to combine different parts into his or her most satisfactory suit by using the system. In fact, the above is a typical combination optimization problem and can be solved by evolutionary optimization methods.

But what is “the most satisfactory suit”? Different persons have different opinions on it because of different personalities and these opinions are often fuzzy and implicit. Therefore, it is impossible to get a uniform and explicit index to

Table 1: *Colors and their codes*

Color	Code	Color	Code
black	0000	gray	1000
blue	0001	bright blue	1001
green	0010	bright green	1010
cyan	0011	bright cyan	1011
red	0100	bright red	1100
carmine	0101	bright carmine	1101
yellow	0110	bright yellow	1110
white	0111	bright white	1111

be optimized. It is infeasible for GAs to deal with it, whereas it is suitable for IGAs to do.

Therefore, we developed a fashion evolutionary design system based on IGA-IFNAH by using Visual Basic 6.0. We also developed corresponding fashion evolutionary design systems based on an IGA with continuous fitness, called traditional IGA (TIGA) [Gong et al. 07], and an IGA with interval individual fitness (IGA-IIF) [Gong and Guo 07] respectively by using the same development tool, and did some experiments to compare their performances.

4.2 Individual Codes

The same individual code is adopted in these systems. For simplification, the phenotype of an individual is a suit composed of coat and skirt, and its genotype is a binary string of 18 bits, where the first 5 bits expresses the style of coat, the 6th to 10th bits expresses the style of skirt, the 11th to 14th bits expresses the color of coat, and the last 4 bits expresses the color of skirt. There are 32 styles for coat and skirt respectively, and their names correspond to the integers from 0 to 31, which are also their decimals of these binary codes. The colors and their codes are shown as Table 1. They are all stored in different databases. According to human's preference, these systems look for "the most satisfactory suit" in the design space with $2^5 \times 2^5 \times 2^4 \times 2^4 = 262144$ suits during evolutionary optimization.

4.3 Parameters Setting

In order to compare the performance of the three algorithms, the same genetic operators and parameters but different approaches to evaluate an individual during evolution are adopted. The population size $|x(t)|$ is equal to 8. $f_{\min}$ and

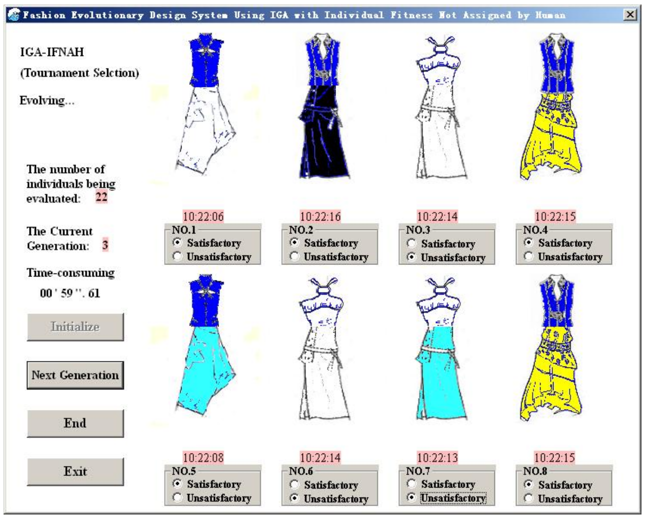

Figure 1: *Interface of human-computer interaction in IGA-IFNAH*

$f_{\max}$ are 0 and 1000 respectively. Tournament selection with size 2, one-point crossover and one-point mutation operators are adopted, and their probabilities p_{c} and p_{m} are 0.6 and 0.02 respectively. The allowable maximum evolutionary generations T is equal to 16. That is to say, if the evolution does not converge after 16 generations, the system will automatically stop it. When the evolution converges, namely there are at least 6 individuals with the same phenotype in some generation the system will also automatically stop it. Also, when human is satisfied with the optimal results, one can stop the evolution manually.

4.4 Evolutionary Interface and Individual Evaluation

The interface of human-computer interaction in IGA-IFNAH, shown as Fig. 1, includes 3 parts. The first one is individual phenotype and their evaluations. Human evaluates the suits through selecting such radio buttons as "satisfactory" or "unsatisfactory" in an appropriate order. The second part is command buttons for a population evolving, e.g., "Initialize", "Next Generation", "End" and "Exit". And the third one is some statistic information of the evolution, including the number of individuals being evaluated, the current generation, and time-consuming.

Once the evolutionary system displays the population in some generation through the interface, human will look for the most sensitive individuals. If an individual is identified as a satisfactory one, human will click the "satisfactory" radio button under it; otherwise, human will click the "unsatisfactory" radio

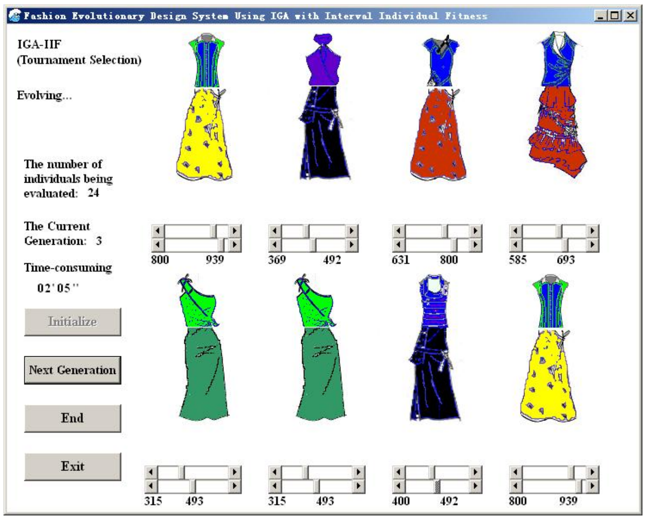

Figure 2: *Interface of human-computer interaction in IGA-IIF*

button. At the same time, the system will automatically record and display time to click these buttons. Then human will look for the second sensitive individuals and perform the same operation until all individuals in this generation are identified. For example, as shown in Fig. 1, human thought the 1st individual in the 3rd generation was the most satisfactory. Therefore, he/she clicked the "satisfactory" radio button under it, and the system automatically recorded and displayed time to click, shown as "10:22:06" with pink background color upon the radio buttons. Another example, human thought the 7th individual in the 3rd generation was the most unsatisfactory. Hence, he/she clicked the "unsatisfactory" radio button under it, and the system also automatically recorded and displayed time to click, shown as "10:22:13" with pink background color upon the radio buttons. As mentioned in subsection 3.2, the 3rd and the 6th individuals with the same phenotype have the same time to be selected as shown in Fig. 1, so do the 4th and the 8th individuals.

After human has identified all individuals, the system will automatically calculate their fitness according to (3). If human clicks "Next Generation", the system will perform genetic operators described as subsection 4.3 to generate offspring, and then display them to human. The system will cycle the above procedure until the evolution automatically or manually stops.

The interface of human-computer interaction in IGA-IIF, shown as Fig. 2, also includes 3 parts. The first one is individual phenotype and their evaluations. In order to assign the fitness of a suit, human drags the two scroll bars under it. Of the two scroll bars, the upper one stands for the lower limit of the fitness,

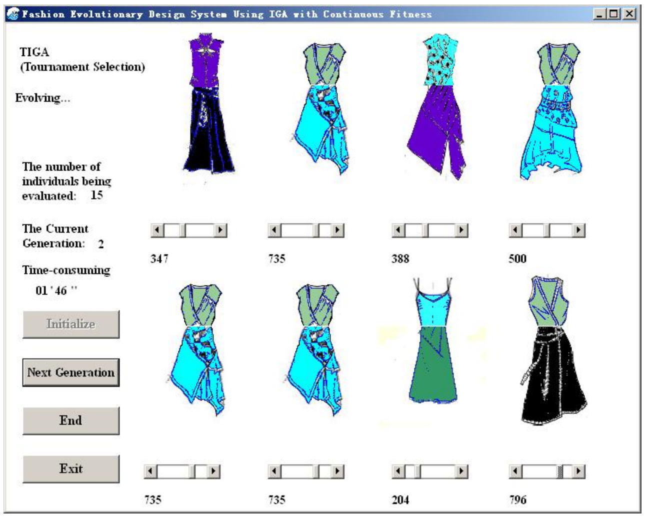

Figure 3: *Interface of human-computer interaction in TIGA*

and the lower one stands for the upper limit of the fitness. The lower limit and the upper limit are also displayed under these scroll bars. The second and the third parts are the same as those in IGA-IFNAH. Having evaluated all suits, if human clicks "Next Generation", the system will perform genetic operators described as subsection 4.3 to generate offspring, and then display them to human. The system will cycle the above procedure until the evolution automatically or manually stops. The interested reader can refer [Gong and Guo 07] for detail.

Similarly, the interface of human-computer interaction in TIGA, shown as Fig. 3, also includes 3 parts. The first one is individual phenotype and their evaluations. In order to assign the fitness of a suit, human drags the scroll bar under it only once. The second and the third parts are the same as those in IGA-IFNAH. Having evaluated all suits, if human clicks "Next Generation", the system will perform genetic operators described as subsection 4.3 to generate offspring, and then display them to human. The system will cycle the above procedure until the evolution automatically or manually stops. The interested reader can refer [Gong et al. 07] for detail.

4.5 Results

We ran the three evolutionary systems based on IGA-IFNAH, IGA-IIF and TIGA respectively 8 times independently, recorded the time-consuming for evaluating individuals and the number of individuals being evaluated in each run, and calculated their sums, shown as Table 2 and Table 3.

Table 2: *Time-consuming for evaluating individuals (m's")*

	IGA-IFNAH	IGA-IIF	TIGA
1	01'22"	07'48"	05'40"
2	00'56"	03'00"	04'02"
3	00'57"	06'10"	06'58"
4	01'16"	**08'33"**	**07'44"**
5	01'15"	03'44"	03'10"
6	00'50"	03'41"	05'02"
7	**01'18"**	03'53"	05'49"
8	01'03"	05'17"	06'15"
Sum	8'57"	42'06"	44'40"

Table 3: *Number of individuals being evaluated*

	IGA-IFNAH	IGA-IIF	TIGA
1	39	81	59
2	45	28	42
3	38	65	63
4	40	**96**	**86**
5	46	35	39
6	35	38	45
7	**52**	39	56
8	37	62	69
sum	332	444	459

It can be seen from Table 2 that for IGA-IFNAH, IGA-IIF and TIGA, the longest time-consuming for evaluating individuals in each run is 01'18", 08'33" and 07'44" respectively. They are all less than 10 minutes, which is acceptable because human often does not feel fatigue within 10 minutes. This means that it often takes human much less time to design fashion by using these systems.

It is easy to see from Table 3 that for IGA-IFNAH, the largest number of individuals being evaluated is 52, which is equivalent to the population evolving about 7 generations, far less than T. That is to say, human found "the most satisfactory suit" in small generations by using IGA-IFNAH. For IGA-IIF, except one run without finding "the most satisfactory suit", the other runs found "the most satisfactory suit" in at most 12 generations. For TIGA, all runs found "the most satisfactory suit" in at most 11 generations. This indicates the three

Table 4: *Average time-consuming for evaluating individuals in each run and for evaluating an individual*

Algorithms	Average time-consuming for evaluating individuals in each run (m's")	Average time-consuming for evaluating an individual (s")
IGA-IFNAH	**1'07"**	**1.6"**
IGA-IIF	5'16"	5.7"
TIGA	5'35"	5.8"

algorithms are feasible to deal with fashion design.

In order to compare the performance of different algorithms in alleviating human fatigue, we calculated the average time-consuming for evaluating individuals in each run and the average time-consuming for evaluating an individual, shown as Table 4. The items in Table 4 are calculated from the data in Table 2 and Table 3. We obtained the 2nd column of Table 4 through dividing the last row of Table 2 by 8, and the 3rd column of Table 4 through dividing the last row of Table 2 by that of Table 3.

It is obvious from Table 4 that the average time-consuming for evaluating individuals in each run of IGA-IFNAH is 1'07", which is about one-fifth of that of IGA-IIF (5'16") and TIGA (5'35"). In addition, the average time-consuming for evaluating an individual of IGA-IFNAH is 1.6", which is less than a third of that of IGA-IIF (5.7") and TIGA (5.8"). Different time-consuming for evaluating an individual is due to different approaches of evaluation. For TIGA, human needs to assign an accurate fitness to an individual, therefore it takes him/her much time to consider what the fitness should be. For IGA-IIF, human does not need to assign an accurate fitness to an individual. In order to obtain an individual fitness, human needs to assign its upper limit and lower limit. The above approaches need to assign an individual fitness by human. In contrast, for IGA-IFNAH, an individual fitness is not assigned by human directly but automatically calculated by the evolutionary system. What human does is to identify an individual satisfactory or unsatisfactory in an appropriate order according to him/her preference, which alleviates human fatigue greatly.

The success rate to find "the most satisfactory suits" within limited time is another index to compare the performance of these algorithms. We calculated the success rate to find "the most satisfactory suits" within 3 minutes, 4 minutes and 5 minutes respectively. Considering the 8 independent runs, we recorded the times to find "the most satisfactory suits" within 3 minutes, 4 minutes and 5 minutes, and then divided these numbers by 8. For example, there are 4 times

Table 5: *Success rates*

Algorithms	Within 3 minutes	Within 4 minutes	Within 5 minutes
IGA-IFNAH	100%	100%	100%
IGA-IIF	12.5%	50%	50%
TIGA	0	12.5%	25%

for IGA-IIF to find "the most satisfactory suits" within 5 minutes, therefore the success rate of IGA-IIF within 5 minutes is $\frac{4}{8} \times 100\% = 50\%$. The success rates of different algorithms within different time is shown in Table 5.

It is easy to see from Table 5 that when human spent 3 minutes in evaluating individuals, all runs of IGA-IFNAH found "the most satisfactory suits", only one run of IGA-IIF found it, while TIGA did not find it. When time increases to 5 minutes, 4 runs of IGA-IIF found "the most satisfactory suits", while only 2 runs of TIGA found it. This indicates that IGA-IFNAH has more opportunities to find "the most satisfactory suits" in short time than the other two algorithms.

To sum up, the proposed algorithm in this paper has good performance in alleviating human fatigue and looking for "the most satisfactory suits".

It is worth noting that the system given in this section is only an experimental platform. The real-world fashion design process is very complicated. For example, a suit may be divided into many parts, and each part may have many styles and colors. Therefore, the whole design space may be considerable large. In any case, the approach of applying evolutionary optimization in fashion design proposed in this paper is novel and feasible, and it establishes a foundation for real-world application. Therefore, it is significant in theory and practice.

5 Conclusions

Human fatigue problem, resulting from evaluation on individuals and assignment of their fitness, is very important and hard to solve in IGAs. How to solve human fatigue problem effectively becomes key to improve performance of IGAs.

It is easy to understand that human fatigue will be alleviated to some degree if he/she does not directly assign an individual fitness. Based on this, a novel interactive genetic algorithm, namely IGA-IFNAH, is proposed in this paper in which human does not directly assign an individual fitness. According to different sensitiveness of human to different individuals, we record time to choose an individual from a population as a satisfactory or unsatisfactory one, and automatically calculate an individual fitness by a transformation from time space to fitness space, then perform subsequent genetic operation based on this fitness, and generate offspring. Application in fashion design validates its efficiency.

An uncertain individual fitness, such as interval fitness, can reflect human's fuzzy and gradual cognition to an individual [Gong and Guo 07]. We obtain a certain individual fitness through a transformation from time space to fitness space in this paper. It is hard for this fitness expression to reflect human's real cognition. Therefore, we will further study how to obtain an uncertain individual fitness through other transformation in the future.

Acknowledgement

This paper was completed when Dunwei Gong was visiting CERCIA, School of Computer Science, the University of Birmingham. Project "Theory and application of evolutionary optimization for optimized problems with uncertain hybrid indices" was supported by National Natural Science Foundation of China with granted No. 60775044 and Program for New Century Excellent Talents in University with granted No. NCET-07-0802.

References

[Adams et al. 88] Adams, J., Balas, E., Zawack, D.: "The Shifting Bottleneck Procedure for Job-shop Scheduling"; Management Science, 34, 3 (1988), 391–401.

[Anderson 05] Anderson, J. R.: "Cognitive Psychology and Its Implications"; Worth Publishers, New York / U.K. (2005).

[Biles et al. 96] Biles, J. A., Anderson, P. G., Loggi, L. W.: "Neural Network Fitness Functions for a Musical IGA"; Proc. International Symposium on Intelligent Industrial Automation and Soft Computing, (1996), 39–44.

[Caldwell et al. 91] Caldwell, C., Johnston, V. S.: "Tracking a Criminal Suspect through 'Face-space' with a Genetic Algorithm"; Proc. 4^{th} International Conference on Genetic Algorithms, (1991), 416–421.

[Dawkins 86] Dawkins, R.: "The Blind Watchmaker"; Longman, Essex / U.K. (1986).

[Deb and Goldberg 89] Deb, K., Goldberg, D. E.: "An Investigation of Niche and Species Formation in Genetic Function Optimization"; Proc. 3^{rd} International Conference on Genetic algorithms, (1989), 42–50.

[Deb et al. 02] Deb, K., Pratap, A., Agarwal, S., Meyarivan T.: "A Fast and Elitist Multiobjective Genetic Algorithm: NSGACII"; IEEE Transactions on Evolutionary Computation, 6, 2 (2002), 182–197.

[Gong and Guo 07] Gong, D.W, Guo, G.S.: "Interactive Genetic Algorithms with Interval Fitness of Evolutionary Individuals"; Dynamics of Continuous, Discrete and Impulsive Systems, Series B: Complex Systems and Applications-modeling, Control and Simulations, 14(s2) (2007), 446–450.

[Gong et al. 05a] Gong, D., Hao, G., Shi, Y., Shi M.: "Interactive Genetic Algorithm with Holding down Survival of the Fittest Based on Extinction Mechanism"; International Journal of Information Technology, 11, 10 (2005), 11–20.

[Gong et al. 07] Gong, D. W., Hao, G. S., Zhou, Y., Guo Y. N.: "Theory and Applications of Interactive Genetic Algorithms" (in Chinese); Defense Industry, Beijing / China (2007).

[Gong and Pan 03] Gong, D. W., Pan, F. P.: "Theory and Applications of Adaptive Genetic Algorithms"; China University of Mining and Technology, Xuzhou / China (2003).

[Gong et al. 05b] Gong, D., Zhou, Y., Li, T.: "Cooperative Interactive Genetic Algorithm Based on User's Preference"; International Journal of Information Technology, 11, 10 (2005), 1–10.

[Hayashida et al. 02] Hayashida, N., Takagi, H.: "Acceleration of EC Convergence with Landscape Visualization and Human Intervention"; Applied Soft Computing, 1, 4 (2002), 245–256.

[Holland 75] Holland, J. H.: "Adaptation in Natural and Artificial Systems"; University of Michigan, Michigan / U.S.A. (1975).

[Kim et al. 00] Kim, H. S., Cho, S. B.: "Application of Interactive Genetic Algorithm to Fashion Design"; Engineering Applications of Artificial Intelligence, 13, 6 (2000), 635–644.

[Lewis and Ruston 05] Lewis, M., Ruston, K.: "Aesthetic Geometry Evolution in a Generic Interactive Evolutionary Design Framework"; New Generation Computing, 23, 2 (2005), 171–179.

[Li and Azarm 00] Li, H., Azarm, S.: "Product Design Selection under Uncertainty and with Competitive Advantage"; Mechanical Design, 122, 4 (2000), 411–418.

[Lin and Kernighan 73] Lin, S., Kernighan, B. W.: "An Effective Heuristic Algorithm for the Traveling Salesman Problem"; Operations Research, 21, 2 (1973), 498–516.

[Llorà et al. 05] Llorà, X., Sastry, K., Goldberg, D. E., Gupta, A., Lakshmi, L.: "Combating User Fatigue in iGAs: Partial Ordering, Support Vector Machines, and Synthetic Fitness"; Proc. Genetic and Evolutionary Computation Conference, (2005), 1363-1370.

[Pallez et al. 07] Pallez, D., Collard, P., Baccino, T., Dumercy, L. : "Eye-tracking Evolutionary Algorithm to Minimize User Fatigue in IEC Applied to Interactive Onemax Problem"; Proc. Genetic and Evolutionary Computation Conference, (2007), 2883–2886.

[Srinivas and Patnaik 94] Srinivas, M., Patnaik, L. M.: "Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms"; IEEE Transactions on Systems, Man and Cybernetics, 24, 4 (1994), 656–667.

[Sugimoto and Yoneyama 01] Sugimoto, F., Yoneyama, M.: "An Evaluation of Hybrid Fitness Assignment Strategy in Interactive Genetic Algorithm"; Proc. 5^{th} Australasia-Japan Joint Workshop on Intelligent and Evolutionary Systems, (2001), 62–69.

[Takagi 01] Takagi, H.: "Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation"; Proc. IEEE, 89, 9 (2001), 1275–1296.

[Takagi and Ohsaki 07] Takagi, H., Ohsaki, M.: "Interactive Evolutionary Computation-based Hearing Aid Fitting"; IEEE Transactions on Evolutionary Computation, 11, 3 (2007), 414–427.

[Takagi and Ohya 96] Takagi, H., Ohya, K: "Discrete Fitness Values for Improving the Human Interface in an Interactive GA"; Proc. IEEE Conference on Evolutionary Computation, (1996), 109–112.

[Tokui and Iba 00] Tokui, N., Iba, H.: "Music Composition with Interactive Evolutionary Computation"; Proc. 3^{rd} International Conference on Generative Art, (2000), 215–226.

[Wang et al. 06] Wang, S. F., Wang. X. F., Takagi, H.: "User Fatigue Reduction by an Absolute Rating Data-trained Predictor in IEC"; Proc. IEEE Congress on Evolutionary Computation, (2006), 2195–2200.

[Wiegand 04] Wiegand, R. P.: "An Analysis of Cooperative Coevolutionary Algorithms"; Dissertation of George Mason University, Fairfax, VA, U.S.A. (2004).

[Xie et al. 03] Xie, Z., Li, J.P., Tang, Z.Y.: "Nonlinear Optimization"; University of Defense Science and Technology, Changsa / China (2003).

[Zhou et al. 05] Zhou, Y., Gong, D. W., Hao, G. S., Guo Y. N., Sun X. Y.: "Phase Estimations of Individual's Fitness Based on NN in Interactive Genetic Algorithms"; Control and Decision (in Chinese), 20, 2 (2005), 234–236.