Available online at www.sciencedirect.com
ScienceDirect
Progress in Natural Science 18 (2008) 359–365
Progress in Natural Science

Special report

Adaptive interactive genetic algorithms with individual interval fitness

Dunwei Gong *, Guangsong Guo, Li Lu, Hongmei Ma

School of Information and Electrical Engineering, China University of Mining & Technology, Xuzhou 221008, China

Received 29 October 2007; received in revised form 15 November 2007; accepted 15 November 2007

Abstract

It is necessary to enhance the performance of interactive genetic algorithms in order to apply them to complicated optimization problems successfully. An adaptive interactive genetic algorithm with individual interval fitness is proposed in this paper in which an individual fitness is expressed by an interval. Through analyzing the fitness, information reflecting the distribution of an evolutionary population is picked up, namely, the difference of evaluating superior individuals and the difference of evaluating a population. Based on these, the adaptive probabilities of crossover and mutation operators of an individual are presented. The algorithm proposed in this paper is applied to a fashion evolutionary design system, and the results show that it can find many satisfactory solutions per generation. The achievement of the paper provides a new approach to enhance the performance of interactive genetic algorithms.

Keywords: Optimization; Genetic algorithm; Interaction; Interval; Probability of genetic operator

1. Introduction

In recent years, genetic algorithms (GAs), a kind of global and probabilistic optimization algorithms with high performance, have been paid broad attentions by researchers world wide, and plentiful achievements have been made. Traditional genetic algorithms do not require the optimization problems be continuous and differentiable, but often require that their indices are well defined in order to calculate an individual fitness. This is difficult for many complicated optimization problems, because the indices of these problems are usually fuzzy and described qualitatively, and one calls them implicit indices. Therefore, traditional genetic algorithms cannot be applied to these problems.

Interactive genetic algorithms (IGAs), proposed in middle 1980s, are effective methods to solve optimization problems with implicit indices [1]. They combine traditional evolutionary mechanism with human's intelligent evaluation, and an individual fitness is assigned by human rather than by a function that is difficult or even impossible to describe explicitly. IGAs are a new cooperation mode that combines human with computers, and this interaction embodies a human's preference fully. Up to now, they have been applied successfully to such fields as face identification, fashion design, music composition, linguistics processing and rhythm control, knowledge acquisition and data mining, and so on [2–6].

The obvious characteristic of IGAs, compared with traditional GAs, is the evaluation of an individual assigned by human. The problem of human fatigue results in a small population size and evolutionary generations of IGAs [7], which influences the performance of IGAs and restricts their applications to complicated optimization problems with implicit indices. Accordingly, how to evaluate an individual to improve the performance of these algorithms becomes one of the key issues.

Generally speaking, there are two methods to evaluate an individual as follows. One is that human evaluates an individual directly based on his or her preference, a fitness

* Corresponding author. Tel.: +86 516 83995199.
E-mail address: dwgong@vip.163.com (D. Gong).

1002-0071/$ - see front matter
doi:10.1016/j.pnsc.2007.11.010

assignment method combining continuous with discrete fitness [8] was proposed. The other is that the evaluation is based on either human or approximate models. Sugimoto et al. estimated an individual fitness using fuzzy logic based on the distance and angle between the evaluated individual and the optima [9]. Biles and Zhou et al. adopted neural networks to learn human's intelligent evaluation, and the number of individuals evaluated by human is reduced by use of neural networks in appropriate time [10,11]. Llora et al. estimated an individual fitness based on support vector machine [12], while Hao et al. did it based on the "fitness" of gene sense units [13]. Wang et al. transformed the evaluation on an individual assigned by human into an absolute rating fitness and applied it to train a surrogate to evaluate an individual [14].

But the fitness in all these methods above is precise. As we all know, the cognition process of human is fuzzy and gradual, therefore the evaluation on an individual assigned by human should also be fuzzy and gradual. It is difficult to reflect the above characteristics if a precise value is adopted to describe an individual fitness.

It is significant to improve the performance of IGAs by picking up the knowledge on a population, and applying it to the subsequent evolutions. In addition to produce approximate models and estimate an individual fitness mentioned before, there are many other ways. For example, Wang et al. constructed the parents of a population by support vector machine to enlarge the range of superior parents [15]. Hu et al. applied rough set theory to pick up the knowledge on an evolution process to speed up the convergence rate of an algorithm [16]. Jiang et al. gained a human's preference through summing up these results from genetic operation in the first several generations, which is applied to guide selection and mutation operators [17]. Gong et al. divided the whole search space into satisfactory subspace, taboo subspace, and unknown subspace, which were evolved by common operator and distinct operator, resulting in shrinking the satisfactory subspace and improving the ability in exploitation [18].

There is another effective method that improves the performance of IGAs by improving traditional genetic operators, in which a typical one is to change the probabilities of crossover and mutation operators adaptively according to a population evolution. There are a lot of achievements on adaptive probabilities of crossover and mutation operators in traditional GAs, one refers Ref. [19] to a comprehensive review given by Gong et al., while few about those in IGAs [20]. The reason is that it is difficult to establish a quantitative relation between these probabilities and an individual fitness.

In order to embody the cognition law of human on an evaluated object, and reduce human fatigue resulting from evaluating an individual, an interval is adopted to express an individual fitness in this paper. Based on the knowledge on a population, novel probabilities of crossover and mutation operators, which are changeable adaptively, and an adaptive interactive genetic algorithm with individual interval fitness (AIGA-IIF) will be proposed. The algorithm proposed can find many satisfactory solutions and reduce human fatigue at the same time. The feasibility and efficiency of the algorithm will be verified through its applications in a fashion evolutionary design system.

2. Individual interval fitness

In IGAs, the evaluation of an individual is an interactive process between human and computers. In traditional IGAs, the evaluation of an individual is assigned by human in the range of fitness according to his or her preference, and it is a precise value. It is easy to understand that the greater the value, the more human prefers the evaluated object. But this kind of evaluation method cannot reflect human's preferences exactly because of fuzzy and gradual human's cognition. In addition, in order to assign precise evaluation, many times of maps from sentient thought to rational thought is necessary, therefore increasing human fatigue inevitably.

In order to reflect the human's cognition on an evaluated object exactly, an interval is adopted to express its fitness in this paper, namely, the fitness of an individuals is an interval. When an interval is adopted to express an individual fitness, only the approximate upper and the lower limit of the fitness of the individual are required, which alleviates human's load during his or her evaluation process dramatically.

Let the ith individual of a population in the tth generation be $x_i(t)$, $i = 1,2,\cdots,N$, and the population size be N. Because of fuzzy human's cognition on $x_i(t)$, one can hardly assign $x_i(t)$'s fitness exactly, but does assign its range easily, which can be expressed with an interval. Therefore, $x_i(t)$'s fitness can be described as follows.

$$f(x_i(t)) = [\underline{f}(x_i(t)), \bar{f}(x_i(t))] \tag{1}$$

Its width $w(f(x_i(t)))$ can be defined as

$$w(f(x_i(t))) = \bar{f}(x_i(t)) - \underline{f}(x_i(t)) \tag{2}$$

and its midpoint $m(f(x_i(t)))$ can be defined as

$$m(f(x_i(t))) = \frac{\underline{f}(x_i(t)) + \bar{f}(x_i(t))}{2} \tag{3}$$

where $\underline{f}(x_i(t))$ and $\bar{f}(x_i(t))$ are the lower limit and the upper limit of human's evaluation on $x_i(t)$, respectively.

Generally speaking, human has different preferences to different individuals, resulting in different fitness intervals. It is easy to see that the greater the lower limit of $f(x_i(t))$ together with the smaller of its width, the higher and the more exact the evaluation on $x_i(t)$ is; otherwise, the smaller the upper limit of $f(x_i(t))$ together with the greater its width, the lower and the rougher the evaluation on $x_i(t)$ is. In general, the human's cognition on $x_i(t)$ is fuzzy at early stage of a population evolution, therefore $w(f(x_i(t)))$ is great. This cognition will become clearer and clearer along with the evolution, and therefore $w(f(x_i(t)))$ will become narrower

and narrower. Compared with a precise fitness, an interval fitness is more approximating the mode of human thought which embodies fuzzy and gradual cognition of human on evaluated objects validly.

3. Populations distribution

When an interval is adopted to express an individual fitness, all fitness intervals form the fitness interval of a population. It is easy to understand that the greater the lower limit of an individual fitness, the more human prefers it, and hence it is a superior one. Therefore, one can find such an individual by analyzing the distribution of intervals with great fitness. On the contrary, the smaller the upper limit of an individual fitness, the less human prefers it, and hence it is an inferior one. Therefore, one can find such an individual by analyzing the distribution of intervals with small fitness.

3.1. Superior fitness interval

Considering a population $x(t)$, an interval that the fitness of superior individuals belong to is called a superior fitness interval, and denoted as $f_s(t)$. Then one has

$$f_s(t) = \left[\max_{x_i(t)\in x(t)} \underline{f}(x_i(t)),\ \max_{x_i(t)\in x(t)} \bar{f}(x_i(t))\right] \tag{4}$$

which is shown as Fig. 1a. The individuals whose fitness fall into the superior fitness interval entirely or partly are satisfactory with the number of M. That is to say, an individual $x_j(t)$ is satisfactory if and only if $f_s(t) \cap f(x_j(t)) \neq \emptyset$.

3.2. Inferior fitness interval

Considering a population $x(t)$, an interval that the fitness of inferior individuals belong to is called an inferior fitness interval, and denoted as $f_i(t)$. Then one has

$$f_i(t) = \left[\min_{x_i(t)\in x(t)} \underline{f}(x_i(t)),\ \min_{x_i(t)\in x(t)} \bar{f}(x_i(t))\right] \tag{5}$$

which is shown as Fig. 1b. The individuals whose fitness fall into the inferior fitness interval entirely or partly are unsatisfactory. That is to say, an individual $x_k(t)$ is unsatisfactory if and only if $f_i(t) \cap f(x_k(t)) \neq \emptyset$.

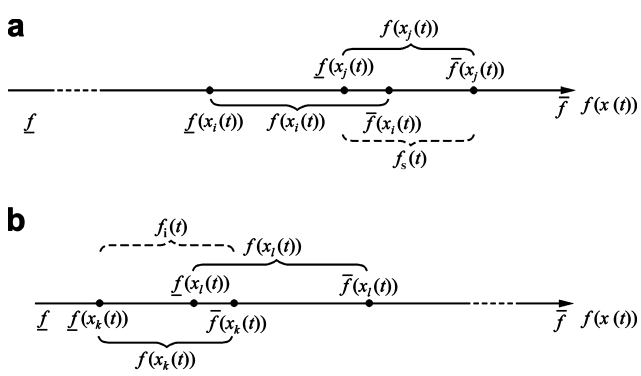

Fig. 1. Superior fitness interval (a) and inferior fitness interval (b).

3.3. Difference of evaluating superior individuals

Considering a superior individual $x_j(t)$, the part of its fitness falling into the superior fitness interval is $[\max_{x_i(t)\in x(t)} \underline{f}(x_i(t)), \bar{f}(x_j(t))]$, whose midpoint is $(\max_{x_i(t)\in x(t)} \underline{f}(x_i(t)) + \bar{f}(x_j(t)))/2$. For all superior individuals, the average midpoint is $\frac{1}{M}\sum_{j=1}^{M}(\max_{x_i(t)\in x(t)} \underline{f}(x_i(t)) + \bar{f}(x_j(t)))/2$, and the difference between the midpoint of the superior fitness interval and the average above is called the difference of evaluating superior individuals, and denoted as $\Delta_1(t)$. Then one has

$$\Delta_1(t) = m(f_s(t)) - \frac{1}{M}\sum_{j=1}^{M}(\max_{x_i(t)\in x(t)} \underline{f}(x_i(t)) + \bar{f}(x_j(t)))/2 \tag{6}$$

$\Delta_1(t)$ describes the distribution of individual fitness in the superior interval. In general, at early stage of a population evolution, the difference among individuals is great, so is the difference among the fitness of superior individuals. Therefore, the dispersion degree of the fitness interval of these superior individuals is great, resulting in a large $\Delta_1(t)$. Along with the evolution, the fitness of different individuals is closer and closer to each other, and the difference among the fitness of these superior individuals becomes smaller and smaller, resulting in a small $\Delta_1(t)$.

3.4. Difference of evaluating population

The difference between the midpoint of the superior fitness interval and that of the inferior fitness interval is called the difference of evaluating a population, and denoted as $\Delta_2(t)$. Then one has

$$\Delta_2(t) = m(f_s(t)) - m(f_i(t)) \tag{7}$$

$\Delta_2(t)$ is the distance between the midpoint of the superior fitness interval and that of the inferior fitness interval, and reflects the difference between the superior individuals and the inferior ones. At early stage of a population evolution, the distribution of evolutionary population is random, and the difference between the superior individuals and the inferior ones is great, resulting in a large $\Delta_2(t)$. Along with the evolution, the superior individuals are conserved, whereas the inferior ones are discarded, which leads to the difference between the superior individuals and the inferior ones decrease, and so does that of fitness, resulting in a small $\Delta_2(t)$.

$\Delta_1(t)$ and $\Delta_2(t)$ describe the change of individuals' fitness during a population evolution, and reflect the homogenization of these fitness from different aspects, which can be used to adjust the probabilities of crossover and mutation operators adaptively.

4. Probabilities of crossover and mutation operators

4.1. Probability of crossover operator

The idea of the probability of crossover operator presented in this paper is as follows.

Firstly, the probability of crossover of an individual will be great if the midpoint of its interval fitness is close to that of the superior one.

Secondly, the probability of crossover of an individual will be small if the difference of evaluating superior individuals and (or) the difference of evaluating a population are (is) great, implying large dispersion degree among the individuals.

Lastly, the probability of crossover of an individual will be small along with the population evolution. For the aim of a population evolution is to ensure the convergence of an algorithm at later stage of the evolution.

Based on those above, the probability of crossover $p_c(x_i(t))$ of an individual $x_i(t)$ is given as follows.

$$p_c(x_i(t)) = \frac{1}{1 + \exp\left(-k_1 \cdot \frac{T}{t} \cdot \frac{\Delta_1(t)+\Delta_2(t)}{m(f_s(t))-m(f(x_i(t)))}\right)} \tag{8}$$

where T is the termination generation, and k_1 is an adjustable coefficient.

Considering two parents $x_i(t)$ and $x_j(t)$, firstly, calculate $p_c(x_i(t))$ and $p_c(x_j(t))$ according to Formula (8), then choose the larger one as their probability of crossover, and perform crossover.

4.2. Probability of mutation operator

Similarly, the idea of the probability of mutation operator presented in this paper is as follows.

Firstly, the probability of mutation of an individual will be great if the midpoint of its interval fitness is close to that of the inferior one.

Secondly, the probability of mutation of an individual will be small if the difference of evaluating superior individuals and (or) the difference of evaluating a population are (is) great, implying large dispersion degree among the individuals.

Lastly, the probability of mutation of an individual will be small along with the population evolution, for the aim of a population evolution is to ensure the convergence of an algorithm at later stage of the evolution.

Based on those above, the probability of mutation $p_m(x_i(t))$ of an individual $x_i(t)$ is presented as follows.

$$p_m(x_i(t)) = \frac{1}{1 + \exp\left(-k_2 \cdot \frac{T}{t} \cdot \frac{\Delta_1(t)+\Delta_2(t)}{m(f(x_i(t)))-m(f_i(t))}\right)} \tag{9}$$

where k_2 is another adjustable coefficient.

Considering $x_i(t)$, firstly, calculate $p_m(x_i(t))$ according to Formula (9), and then perform mutation with this probability.

5. Steps of the algorithm

The steps of the algorithm proposed in this paper include:

Step 1: Set the values of control parameters in the algorithm. Let $t = 0$, and initialize a population denoted as $x(t)$.
Step 2: An interval fitness is assigned on an individual.
Step 3: Produce parents by tournament selection.
Step 4: Perform crossover and mutation operators according to Formula (8) and (9), and produce offspring. Let $t = t + 1$.
Step 5: Judge if the termination criterion is met, if yes, then go to step 6; otherwise go to step 2.
Step 7: Output the optima and stop the algorithm.

6. Applications in a fashion evolutionary design system

Fashion design is a very popular vocation for everyone who likes to wear fashion ably but few can design a satisfactory one. In fact, design of fashion is a very complicated process and often completed by designers who have been trained professionally. Although there are some softwares available for fashion design, they are often too special for an ordinary person to use. With the development of social life style pursuing personality 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 be optimized. It is infeasible for traditional GAs to deal with it, whereas it is suitable for IGAs to do.

Therefore, we developed a fashion evolutionary design system based on AIGA-IIF 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), and an IGA with interval individual fitness (IGA-IIF) by using the same development tool, and did some experiments to compare their performances.

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 express the style of coat, the 6th to 10th bits express the style of skirt, the

11th to 14th bits express the color of coat, and the last 4 bits express the color of skirt. There are 32 styles for coat and skirt, 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 in 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.

6.1. Parameter settings

The minimal and the maximal individual fitness are $\underline{f}(x_i(t)) = 0$, $\bar{f}(x_i(t)) = 1000$, respectively. The population size and the number of individuals in tournament selection are 8 and 2, respectively. In the experiment, the probabilities of crossover and mutation operators in IGA-IIF and TIGA are listed in Table 2. $k_1 = k_2 = 9$ in Formulas (8) and (9). Stop a population evolution manually when the population converges or human is satisfied with the evolutionary results.

6.2. Performance analysis

Firstly, the algorithm proposed in this paper is considered. After an evolution stops, the number of satisfactory solutions in this run is calculated, and the algorithm is run 20 times independently with the results listed in Table 3.

Then IGA-IIF and TIGA are considered. For the cases of different probabilities of crossover and mutation operators listed in Table 2, the evolutionary generations and the number of satisfactory solutions in each run are calculated, and the algorithm is run 20 times independently with the results listed in Tables 4 and 5.

It can be seen from Tables 4 and 5 that the evolutionary generations and the number of satisfactory solutions of IGA-IIF and TIGA will increase when the probability of crossover operator is constant while the probability of mutation operator increases, and so as to the case when the probability of mutation operator is constant while the probability of crossover operator increases, but the later case has great influence on them, implying that crossover is the main operator, while mutation is secondary.

In addition, it can be seen from Tables 4 and 5 that the evolutionary generations of IGA-IIF is fewer than that of TIGA, while the number of satisfactory solutions found by IGA-IIF is more than that of TIGA under the same conditions, reflecting that an interval adopted to express an individual fitness can enhance the performance of algorithm.

Finally, the three algorithms, namely AIGA-IIF, IGA-IIF and TIGA, are considered. The number of satisfactory solutions per generation with different probabilities of crossover and mutation operators is calculated, and the results of 20 independent runs are listed in Table 6.

It can be seen from Table 6 that for 9 different conditions above, the number of satisfactory solutions per generation found by AIGA-IIF is the most, IGA-IIF the second, and TIGA the least, which indicates that the performance of IGAs can be enhanced in different degrees by adopting an interval fitness and adaptive probabilities of crossover and mutation operators to them.

Table 1
Colors and their codes

Code	Color	Code	Color
0000	Black	1000	Gray
0001	Blue	1001	Bright blue
0010	Green	1010	Bright green
0011	Cyan	1011	Bright cyan
0100	Red	1100	Bright red
0101	Carmine	1101	Bright carmine
0110	Yellow	1110	Bright yellow
0111	White	1111	Bright white

Table 3
Evolutionary generations and number of satisfactory solutions

No. of experiment	Evolutionary generations	No. of satisfactory solutions
1	9	27
2	9	22
3	8	24
4	11	30
5	9	27
6	10	29
7	8	26
8	9	28
9	11	30
10	9	26
11	8	25
12	10	26
13	11	28
14	9	30
15	8	26
16	8	25
17	10	23
18	12	32
19	11	24
20	9	26
Average	9.4	26.7

Table 2
Probabilities of crossover and mutation operators in IGA-IIF and TIGA

	1	2	3	4	5	6	7	8	9
p_c	0.50	0.50	0.50	0.60	0.60	0.60	0.70	0.70	0.70
p_m	0.05	0.08	0.10	0.05	0.08	0.10	0.05	0.08	0.10

Table 4
Average evolutionary generations

No. of experiment	1	2	3	4	5	6	7	8	9
IGA-IIF	10.45	10.70	11.05	11.40	11.80	12.05	12.10	12.15	12.20
TIGA	10.55	10.60	10.70	10.75	10.85	11.25	11.50	12.00	12.55

Table 5
Average number of satisfactory solutions

No. of experiment	1	2	3	4	5	6	7	8	9
IGA-IIF	25.35	26.05	26.95	27.95	29.10	29.75	30.00	30.20	30.40
TIGA	24.75	24.95	25.25	25.45	25.80	26.75	27.40	28.70	30.10

Table 6
Number of satisfactory solutions per generation

	AIGA-IIF	IGA-IIF	TIGA
1		2.426	2.346
2		2.435	2.354
3		2.439	2.360
4		2.452	2.367
5	2.84	2.466	2.377
6		2.469	2.378
7		2.479	2.383
8		2.486	2.392
9		2.492	2.398

7. Conclusion

The key of applying IGAs broadly is to improve their performance. Compared with traditional IGAs, the algorithm proposed in this paper has two features: the first one is that an interval is adopted to express an individual fitness, reflecting human's cognition appropriately; the second one is that the probabilities of crossover and mutation operators change adaptively with a population evolution, maintaining the diversity of population availably and establishing a foundation for finding several satisfactory solutions. The proposed algorithm has been applied to a fashion evolutionary design system, and compared with other algorithms on the evolutionary generations and the number of satisfactory solutions. The experimental results show that the proposed algorithm has good performance.

There are two issues should be further studied: one is that other uncertain methods, such as fuzzy number and stochastic number, are adopted to express an individual fitness based on human's cognition process; the other is based on different expressions above, picking up the knowledge on a population and guiding subsequent evolutions to improve the performance of algorithms.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (Grant No. 60775044) and Program for New Century Excellent Talents in University.

References

[1] Dawkins R. The blind watchmaker. Essex: Longman; 1986.
[2] Caldwell C, Johnston VS. Tracking a criminal suspect through 'face-space' with a genetic algorithm. In: Proceedings of the 4th international conference on genetic algorithms. 1991, p. 416–421.
[3] Smith JR. Designing biomorphs with an interactive genetic algorithm. In: Proceedings of the 4th international conference on genetic algorithms. 1991, p. 535–538.
[4] Kim HS, Cho SB. Application of interactive genetic algorithm to fashion design. Eng Appl Artif Int 2000;13(6):635–44.
[5] Tokui N, Iba H. Music composition with interactive evolutionary computation. In: Proceedings of the 3rd international conference on generative art. 2000, p. 215–226.
[6] Morita T, Iba H, Ishizuka M. Generating emotional voice and behavior expression by interactive evolutionary computation. In: Proceedings of the 62nd annual meeting of Japan society for information processing. 2001, p. 45–46.
[7] Takagi H. Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. In: Proceedings of the IEEE. 2001, p. 1275–1296.
[8] Takagi H, Ohya K. Discrete fitness values for improving the human interface in an interactive GA. In: Proceedings of the IEEE conference on evolutionary computation. 1996, p. 109–112.
[9] Sugimoto F, Yoneyama M. An evaluation of hybrid fitness assignment strategy in interactive genetic algorithm. In: Proceedings of the 5th Australasia–Japan joint workshop on intelligent & evolutionary systems. 2001, p. 62–69.
[10] Biles JA, Anderson PG, Loggi LW. Neural network fitness functions for a musical IGA. In: Proceedings of the international symposium on intelligent industrial automation and soft computing. 1996, p. 39–44.
[11] Zhou Y, Gong DW, Hao GS, et al. Phase estimations of individual's fitness based on NN in interactive genetic algorithms. Control Decis (in Chinese) 2005;20(2):234–6.
[12] Llora X, Sastry K, Goldberg DE, et al. Combating user fatigue in iGAs: partial ordering, support vector machines, and synthetic fitness. Illinois Genetic Algorithms Lab, University of Illinois at Urbana-Champaign, 2005.
[13] Hao GS, Gong DW, Shi YQ, et al. Methods of replacing users with machines in interactive genetic algorithms. Pattern Recogn Artif Int (in Chinese) 2006;19(1):111–5.
[14] Wang SF, Wang XF, Takagi H. User fatigue reduction by an absolute rating data-trained predictor in IEC. In: Proceedings of IEEE congress on evolutionary computation. 2006, p. 16–21.
[15] Wang SF, Wang SF, Wang XF. Improved interactive genetic algorithm incorporating with SVM and its application. J Data Acquisition Process 2003;18(4):429–33.
[16] Hu J, Chen EH, Wang SF, et al. Improvement on convergence and quality of user evaluation in interactive genetic algorithms. J Univ Sci Technol China (in Chinese) 2002;32(2):210–6.

[17] Jiang SS, Cao XB, Wang XF. Model and design of user agent based on IGA. Pattern Recogn Artifi Int (in Chinese) 2004;17(2):244–8.
[18] Gong DW, Hao GS, Zhou Y, et al. Theory and applications of interactive genetic algorithms (in Chinese). Beijing: Press of Defense Industry; 2007.
[19] Gong DW, Pan FP. Theory and applications of adaptive genetic algorithms (in Chinese). Beijing: Press of Defense Industry; 2003.
[20] Gong DW, Hao GS, Zhou Y, et al. Interactive genetic algorithms with multi-population adaptive hierarchy and their application in fashion design. Appl Math Comput 2007;185:1098–108.