Cooperative Interactive Genetic Algorithm Based on User's Preference

D. Gong [1], Y. Zhou [1,2], and T. Li [2]

[1] School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou 221008
[2] School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221008

dwgong@vip.163.com, and zhouyongchina@126.com

Abstract

Combined with the methodology of cooperative genetic algorithm, a cooperative interactive genetic algorithm based on a user's preference is proposed in this paper in allusion to solve a user's fatigue problem in interactive genetic algorithm. The method of picking up a user's preference based on fitness of a building block, the storage format of a user's preference based on a network database, the strategy of looking for users with same or similar preference based on deviation, and the steps of the algorithm are given. The algorithm can avoid a blind search of an initial population as in simple genetic algorithm, lead the search direction to the range that meets user's personalities by using immigrant individuals, and make a user concentrate his or her limited energy on a finer search and evaluation process, hence alleviating a user's fatigue. The efficiency of the algorithm proposed in this paper is verified through an instance.

Keyword: interactive genetic algorithm, preference, cooperative interactive genetic algorithm

I. Introduction

With the popularization of interactive genetic algorithm, the question of how to alleviate a user's fatigue in fitness evaluation becomes increasingly important [1]. In theory, parameters and genetic operators of genetic algorithm can be improved to accelerate the algorithm's convergence so as to shorten user's evaluation time and hence alleviating a user's fatigue. But these methods will lead to premature convergence of a population and the space search ability of genetic algorithm cannot be sufficiently brought into play. In practice, the fitness prediction is used to alleviate a user's burden. The authors gave a two-phased method for estimating individual's fitness based on neural network in 2005[2]; Lee et al gave a sparse fitness evaluation method in 1999[3]; Sugimoto et al gave a hybrid fitness assignment strategy in 2001 [4]. And recently, Kumara et al proposed a building block fitness estimation method [5]. All these fitness prediction methods are based on the current user's evaluation. But a user's evaluation in interactive genetic algorithm is completely subjective and continuously changing with the user's degree of cognition. Hence, the fitness evaluation in interactive genetic algorithm is fluctuant and its deviation will directly affect the reliability of convergent results and restrict the application of these methods.

A preference is mainly applied in multi-objective decision-making. It describes the important degree of different users to different objectives or factors [6-8]. Interactive genetic algorithm is mainly applied to optimize problems that performance index is difficult to be described with explicit functions, such as image generation, music composition, industry design and data mining [1]. These problems often have several objectives. A user's evaluation of different objectives or factors is based on his or her knowledge background, culture tendency, personal enjoyment and other factors, which will turn to be his or her preference for different objectives or factors. If users with same preference for same design problem can be found, the outcome designed by the users can be directly used for reference so that a blind search will be avoided and the current user's design process will also be accelerated.

The application of user's information can be considered as a kind of cooperative evolutions. But it is different from a traditional cooperative genetic algorithm. The methodology of a traditional cooperative genetic algorithm is to divide a variable space into several subspaces according to a certain rule. Each population evolves in one variable subspace along with other populations, and uses other populations' information during evolution. But the cooperation here is to look for the users with same or similar preference and uses the outcome of other user to accelerate the current user's evolutionary process. According to this methodology, a cooperative interactive genetic algorithm based on a user's preference is proposed in this paper. However, only when using cooperative information of the users who have same or similar preference, can the evaluation process of the current user be accelerated; otherwise, it will be disturbed and results in retrogression. Hence, how to pick up and make use of users' preference becomes a key point in applying this method successfully.

In allusion to the problems mentioned above, a fitness estimation method based on a building block is introduced to pick up a user's preference; a cooperative interactive genetic algorithm based on a user's preference is then proposed. Finally, the efficiency of the algorithm proposed in this paper is verified through an instance.

II. Pick-up and Application of User's Preference

A user's preference is included in fitness evaluated by the user. So it can be picked up from fitness evaluated by the user. The similarity of the preferences of two users is calculated by comparing their deviation. The method of picking up a user's preference based on fitness of a building block, the storage format of the user's preference based on a network database, and the strategy of looking for the users with same or similar preference based on deviation are expounded in detail as follows.

A. *Pick-up of User's Preference*

Combined with the fitness estimation method based on a building block proposed by Kumara [5], a method of picking up a user's preference based on a scheme is proposed in this paper in allusion to subjectivity and fluctuation of fitness evaluated by a user in interactive genetic algorithm.

In interactive genetic algorithm, the code of an individual generally has a real meaning. For example, some gene bits represent a kind of styles of the individual, such as the genotype in interactive music composition, face design and fashion design. Therefore, these gene bits with fixed meaning can be considered as a building block with one or many schemes, and each building block corresponds to a kind of factors and the scheme of a factor corresponds to its several preferences. Hence preferences of each factor are turned into schemes of a building block of an individual, and fitness of a scheme reflects a user's preference to an objective. The relationship among factors, building blocks and schemes is shown in Fig. 1, where H_i is a building block corresponding to the *i*-th factor, $i = 1,2\cdots N$, N is the number of factors of an individual; h_i^j is the *j*-th scheme corresponding to the *i*-th building block, $j = 1,2\cdots L_i$; L_i is the number of schemes included in the *i*-th building block. Each building block corresponds to an objective, and each scheme of a building block corresponds to a preference to the objective.

Fitness of a scheme is calculated via: [5]

$$f(h) = \frac{1}{m_h}\sum_{\{i|x_i \supset h\}} f(x_i) - \frac{1}{m'}\sum_{i=1}^{m'} f(x_i) \qquad (1)$$

where m' is the number of individuals being evaluated, m_h is the number of individuals including *h*, $f(x_i)$ is x_i's fitness, and $f(h)$ is *h*'s fitness.

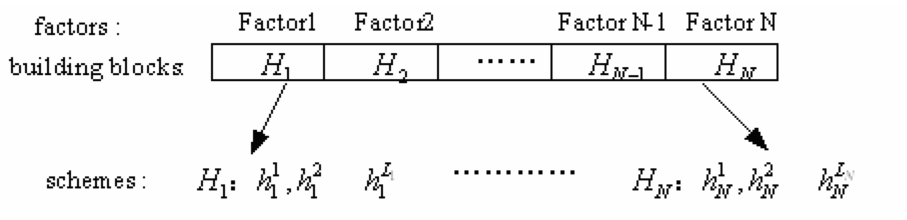

Fig. 1. Relationship among factors, building blocks and schemes

In order to reduce the effect of individual's fitness evaluated by a user on the evolution process, fitness of an individual is accepted with a certain confidence level. The function of confidence level is:

$$R(t) = \begin{cases} 1 + e^{-\alpha T} - e^{-\alpha t} & t < T \\ 1 & t \geq T \end{cases} \qquad (2)$$

where *t* is an evolutionary generation, T is a confidence level threshold, α is a confidence coefficient. In the initial stage of the evolution, user's cognition for an individual is insufficient, and the confidence level of fitness evaluated by him or her is low, namely, fitness of an individual is accepted with a discount. If the design object is so complex that it takes long time for a user to be familiar with it, α is set to a small value, and T is set to a big value. Otherwise, if the design object is so simple that it takes little time for a user to be familiar with it, α is set to a big value, and T is set to a small value. Along with the evolution, user's cognition for an individual is more and more sufficient and the confidence level gradually increases. When *t* reaches T, it can be assumed that a user has completely understood these individuals. In this case, the confidence level is 1.

The function of confidence level reflects the degree of user's cognition for an individual in design process, and the preference of an individual's factor based on schemes is obtained via:

$$P(h) = R(t) \cdot f(h) = R(t) \cdot [\frac{1}{n_h} \sum_{\{i|x_i \supset h\}} f(x_i) - \frac{1}{n'} \sum_{i=1}^{n'} f(x_i)] \quad \textbf{(3)}$$

where $P(h)$ is the preference value of h.

B. Storage of User's Preference

In order to sufficiently utilize the existing optimum individuals in an evolution, an optimum individual archive is designed to conserve information of the optimum individuals in this paper. A user's preference is picked up via formula (3) and stored in a user's preference table. The format of the table is shown in table 1. In table 1, $P_k(h_i^j)$ is the value of the user k 's preference for the *j*-th level of the *i*-th factor, $k = 1,2,\cdots,K$, K is the number of users. Before *t* reaches T, fitness of the optimum individual of each generation is $f(x_i) = R(t) f'(x_i)$, where $f'(x_i)$ is x_i 's fitness evaluated by a user, and $f(x_i)$ is its modified fitness. After *t* has reached T, fitness of an individual needn't be modified, namely, $f(x_i) = f'(x_i)$. The code of the optimum individual in each generation and its corresponding fitness are stored in an optimum individual archive and the details are shown in table 2, where $P_{k,t}$ is the number of genes of the optimum individual, and $f(x_{k,t})$ is its fitness.

Table 1. User's preference table

genotype	*OB1*：factor 1(building block 1)			…	*OBN*：factor *N*(building block *N*)		
preference of user *k*	$P_k(h_1^1)$	…	$P_k(h_1^{L_1})$	…	$P_k(h_N^1)$	…	$P_k(h_N^{L_N})$

Table 2. Optimum individual archive table

code of optimum individual in generation *t* and its fitness of user *k*	No. of genes	genotype	fitness
	$P_{k,t}$	0101…0101	$f(x_{k,t})$

C. Looking for Users with Same Preference

After the user's preference table has been established, one need look for the users with same or similar preference to perform a cooperative evolution. Supposing the factors considered in the design are $OB_1, OB_2, \cdots, OB_N$, for simplicity, the user's degree of cognition to different factors is supposed to be same. To same factor, the similarity of the preferences of two users is calculated by the deviation of their preferences. If the two users' deviation for same factor is equal to zero, it can be thought that the two users have same preference for this factor. The *i*-th factor is OB_i , the similarity on this factor between the users k_1, k_2 is defined as:

$$\sigma_i(k_1,k_2) = \sqrt{\frac{1}{L_i} \sum_{j=1}^{L_i} (P_{k_1}(h_i^j) - P_{k_2}(h_i^j))^2} \quad \textbf{(4)}$$

where $P_{k_1}(h_i^j)$ and $P_{k_2}(h_i^j)$ are the values of k_1's and k_2's preference for the j-th scheme of OB_i respectively.

As the user's degrees of cognition for all the factors are same, $\sigma(z,k)$ is defined as the similarity between the current user z and the user k, whose expression is:

$$\sigma(z,k)=\sum_{i=1}^{n}\sigma_i(z,k) \tag{5}$$

The smaller the $\sigma(z,k)$, the greater the similarity of the preference between z and k for different factors is. And the optimum individuals evaluated by k can be accepted as the cooperative individuals of z. The greater the $\sigma(z,k)$, the greater the discrepancy of the preference between z and k for different factors is, and then the optimum individuals evaluated by k can't be accepted as the cooperative individuals of z. So, the user with same preference as the current user can be found via formula (4) and (5).

III. Cooperative Interactive Genetic Algorithm Based on User's Preference

A. *Methodology of Algorithm*

The cooperative interactive genetic algorithm based on a user's preference presented in this paper uses the existing individuals from the populations that have same or similar evolutionary objectives with the current population to accelerate its evolution. This avoids a blind search of an initial population as in simple genetic algorithm, leads the search direction to the range that meets user's personalities by using immigrant individuals and makes a user concentrate his or her limited energy on a finer search and evaluation process; hence the user can easily find the best results.

The detailed process is as follows. Different users evolve their own populations respectively and submit the optimum individuals of each generation reflecting their own preferences to a database. Each user can decide whether to perform a cooperative evolution according to his or her need. If a user decides to perform a cooperative evolution, the best individual of the user who has the most similar preference with the current user will be selected automatically from the database and be submitted it to the current user. After that, the current user will decide whether to accept the individual. If yes, the worst individual of the current population will be replaced with it and the current user will evaluate it again.

B. *Makeup and Structure of System*

It is possible to exchange and transfer information efficiently because of high speed of Internet and globalization of network distribution. They provide the foundation for Internet-based design. This kind of design will not be restricted by space and time. It makes full use of information resource on Internet, provides an imaginable space for a designer, hence accelerating the process of design. So an Internet-based B/S structure is adopted in the cooperative interactive genetic algorithm based on a user's preference presented in this paper, in which a SQL server database is used to store information of the best individuals, and a web server is used to store information of the design and to generate an interactive webpage. A user visits the server though an IE browser, performs his or her design and makes use of useful information in the server database according to his or her requirement. At the same time, a user submits his or her best individuals in the process of design to the server database for later use by other users.

C. *Steps of Algorithm*

Step1 Generate an initial population and display it to a user for evaluation.

Step2 Submit the optimum individual of the evolution to the database.

Step3 Pick up a user's preference according to formula (3).

Step4 Calculate the similarity for different objectives between the current user and other users according to formula (4).

Step5 Judge whether the current user need a cooperative evolution. If yes, look for the user who has the most similar preference as the current user according to II.C.

Step6 Accept the optimum individual of the user with the most similar preference to the current user.

Step7 Judge whether the current user is satisfied with the cooperative result. If yes, end the evolution; otherwise, evaluate the cooperative result again and replace the worst individual with it.

Step8 perform selection, crossover and mutation operation.

Step9 Evaluate the generated offspring.

Step10 Judge whether the evolution termination condition is met. If yes, go to step2; otherwise, end the evolution and submit the optimum individual.

Among the steps mentioned above, the evaluation process is performed at the browser end, whereas the optimum individuals and a user's preference are stored at the server end. The user's preference table is updated along with new individuals in each generation. If a user need cooperative information, he clicks the button "cooperation" at the browser end to activate the mechanism of cooperation. Then $\sigma(z,k)$ between the current user and other users according to the user's preference table is calculated and the user with the largest similarity is looked for at the server end. After the user has been found, his or her best individual from the optimum individual archive is selected and submitted to the browser end of the current user to be evaluated according to Step7.

IV. Application in Fashion Design Conclusion

The efficiency of the algorithm is validated by a fashion design system based on interactive genetic algorithm in this section. For simplicity, the system divides a fashion into 3 parts and 6 factors. The 3 parts are collar, skirt and sleeve respectively, and each part has 2 factors, namely, the style and the color. Each factor is described with 2 consecutive gene bits, namely, a building block. So an individual's code has 12 bits in total, denoting collar color, collar style, skirt color, skirt style, sleeve color and sleeve style in turn. Each factor has 4 schemes corresponding to 4 projects for a user to select, consequently, each factor has 4 preference levels. Genotypes and preference levels of each factor are shown in table 3. For example, a scheme H＝**00******** denotes a fashion with a big collar. If scheme's fitness obtained from formula (3) is $f(H) = 20$, it denotes that the user's preference for this style of fashion is 20.

A. Pick-up and Application of User's Preference

The key problem of the algorithm proposed in this paper is to look for users with same or similar preference. If he or she is found, the current user can use existing information of the user for reference to accelerate his or her evolution. So the task of the subsection is mainly to validate whether the method of picking up a user's preference is proper and whether the strategy of looking for users with same or similar preference is correct.

Table 3. Genotypes and preference levels

fashion parts	factors	genotypes and preference levels			
		00	01	10	11
collar	collar style	big collar	small collar	flowering collar	tiny collar
	collar color	green	red	yellow	blue
skirt	skirt style	flowering fold long skirt	no fold long skirt	short skirt	canister long skirt
	skirt color	green	red	yellow	blue
sleeve	sleeve style	long sleeve	middle sleeve	flowering short sleeve	small short sleeve
	sleeve color	green	red	yellow	blue

From table 3, the search space is 4096 and there are only 6 factors, of which 3 factors are color. So the design object is not complex and a user can become familiar with it in short time. Hence, the confidence level threshold is set to T＝3 and the confidence coefficient is set to $\alpha = 0.9$. The system runs at a browser end using the parameters mentioned above and user's preferences after each generation are listed in table 4. To simplify the expression, the scheme of a middle collar is denoted as H＝00**********. It can be found from table 4 that the user's preferences become clear gradually along with the evolution. In the end, the user's preferences can be fixed so that the preferences of color, collar style, skirt style and sleeve style are "red", "big collar", "Canister long skirt" and "Flowering shorter sleeve" respectively. So a user with similar preferences can be found according to the current user's preferences and the optimum individual is submitted to the current user for reevaluation. There is a user k with similar preferences as those of the current user in this preference table and k's preferences are also listed in table 4. It can be calculated via formula (5) that the preferences for different factors between the current user z and k are the most similar. Therefore, the system will select an individual of the user k from the optimum individual table and submit it to the current user. The submitted individual is shown as Fig. 2. The individual is just the one satisfying the current user's preferences, namely, "red big collar, red canister long skirt and red flowering shorter sleeve". The current user has found the fashion he or she likes most in the 3rd generation and the algorithm ends.

Fig. 2. Cooperative individual submitted by system

Table 4. Preferences of user *z* and user *k* for fashion design

fashion parts	factors	genotypes	preferences of user *z*			preferences of user *k*
		scheme	t=1	t=2	t=3	
collar	collar style	big collar	20	25	30	31
		small collar	10	10	-10	10
		flowering collar	8	5	10	-10
		tine collar	10	10	10	8
	collar color	green	10	10	10	10
		red	25	35	40	40
		yellow	10	8	-10	-10
		blue	-10	9	10	5
skirt	skirt style	flowering fold long skirt	20	10	-10	-10
		no fold long skirt	10	10	11	10
		Short skirt	-10	10	10	9
		canister long skirt	31	30	40	43
	skirt color	green	10	8	10	10
		red	35	38	40	42
		yellow	-15	8	10	10
		blue	-10	-10	-10	-9
sleeve	sleeve style	long sleeve	10	5	-10	-10
		middle sleeve	10	10	-10	-10
		flowering short sleeve	30	30	40	42
		small short sleeve	-10	-10	-10	-10
	sleeve color	green	24	10	10	9
		red	35	30	40	40
		yellow	10	10	-10	-10
		blue	10	8	-10	-8

B. Complexity Analysis of Algorithm

In order to analyze the complexity of the algorithm, the design result of this paper is compared with that of Zhou et al [2] who adopted a two-phased method for estimating individual fitness based on neural network. It took 25 minutes 31 seconds for them to get the optimum individual, namely, the individual satisfying the user is found after 6 generations. But in this algorithm, only 3 generations are needed to find the fashion satisfying the user, which spends 12 minutes 18 seconds. Hence, by using the existing resources of the system sufficiently, the algorithm proposed in this paper efficiently accelerates the evolution. As a result, a user finds a satisfactory design result in short time.

V. Conclusions

To solve the problem of a user's fatigue, a fitness estimation method based on a building block is introduced to pick up a user's preference and a cooperative interactive genetic algorithm based on a user's preference is proposed in this paper. The method sufficiently utilizes the existing individuals of populations which have similar evolutionary objectives as the current population to accelerate the evolution, avoids a blind search of an initial population as in simple genetic algorithm, leads the search direction to the range that meets user's personalities by using immigrant individuals and makes a user concentrate his or her limited energy on a finer search and evaluation process, hence alleviating a user's fatigue. The method of picking up a user's preference based on fitness of a building block, the storage format of a user's preference based on a network database, the strategy of looking for users with same or similar preference based on deviation and the steps of the algorithm are given in this paper. The efficiency of the algorithm proposed in this paper is verified through an instance of fashion design. When the design factors of the current user are in order, how to look for the users with same or similar preference and the theoretical analysis of the influence of different preferences on the evolution are problems to be further studied.

Acknowledgments

Project Research on Key Issues of Co-interactive Evolutionary Computation and Their Implementation based on Internet (60304016) supported by National Natural Science Foundation of China.

References

[1] H. Takagi, "Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation," in *Proceedings of the IEEE*, 2001, 89(9), pp.1275-1296

[2] Y. Zhou, D. Gong, "Neural network based phase estimation of individual fitness in interactive genetic algorithm," *Control and Decision*, 2005, 20(2), pp.234-236.

[3] J. Lee, S. Cho, "Sparse fitness evaluation for reducing user burden in interactive algorithm," in *Proceedings of 1999 IEEE International Fuzzy Systems Conference*, 1999, pp. II998-II1003.

[4] F. Sugimoto, M. Yoneyama, "An evaluation of hybrid fitness assignment strategy in interactive genetic algorithm," in *Proceeding of the 5th Australasia-Japan Joint Workshop on Intelligent & Evolutionary Systems*, 2001, pp.62-69.

[5] S. Kumara, P. Martin, E. David, "Efficiency enhancement of genetic algorithms via building-block-wise fitness estimation," in *Proceedings of the 2004 Congress on Evolutionary Computation*, 2004, pp.720-727.

[6] C. Bana, *Readings in Multiple Criteria Decision Aid*, Berlin, Germany: Springer-Verlag, 1990.

[7] C. Coello, "Handling preferences in evolutionary multiobjective optimization: A survey," *IEEE Transactions on Evolutionary Computation*, 2000,1(7), pp.30–37.

[8] C. Dragan, C. Parmee, "Preferences and their application in evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, 2002, 6(1), pp.42-57.

D. Gong received the B.S. in applied mathematics from China University of Mining and Technology, Xuzhou, P.R.China, in 1992, M.S. in control theory and its applications from Beijing University of Aeronautics and Astronautics, Beijing, P.R.China, in 1995, and Ph.D. in control theory and control engineering from China University of Mining and Technology, Xuzhou, P.R.China, in 1999.

Since 1995, he has been a research associate in School of Information and Electrical Engineering, China University of Mining and Technology. Since 2003, he takes up the director of the Center for Intelligent System and Intelligent Control, China University of Mining and Technology. His main research interests are evolutionary computation and intelligent control. He has authored and co-authored more than 50 papers.

Y. Zhou received the B.S. in industrial automatization from Hehai University, Nanjing, P.R.China, in 1997, M.S. in control theory and control engineering from China University of Mining and Technology, Xuzhou, P.R.China, in 2003. And now he is a PH.D. student in control theory and control engineering in School of Information and Electrical Engineering, China University of Mining and Technology.

Since 2003, he has been a research associate in School of Computer Science and Technology, China University of Mining and Technology. His main research interests are evolutionary computation and intelligent Control. He has authored and co-authored more than 10 papers.

T. Li has been an undergraduate and specializes in electronic information science and technology in School of Computer Science and Technology, China University of Mining and Technology since 2003. His main research interests are computer application and programming.