The 8th International Conference on
Computer Science & Education (ICCSE 2013)
April 26-28, 2013. Colombo, Sri Lanka

MoB1.4

A Genetic Algorithm of Test Paper Generation

Song Yan, Yang Guoxing*
Guanzhuang Campus
University of Science & Technology Beijing
Beijing, China
gxyang@sina.com

Abstract—**The algorithm of test paper generation is an important research subject in testing. A genetic algorithm of test paper generation is given in the paper. Chromosome coding, fitness function, and genetic operator are discussed in details.**

Keywords—genetic algorithm; test paper generation; item bank system

I. INTRODUCTION

Along with the development of computer and database technology, item bank system has become an indispensable tool of the various educational institutions and examination departments. Efficiency of generating test paper is the one of the key functions of the item bank system. Therefore, professionals have done a lot of research on the method of generating test paper.

Test paper generation is to extraction a set of examination questions from the item bank according to the test paper parameters, such as the types of question, difficulty, knowledge points and exposure times. The most important condition is the question types which the test paper contains and the question amounts of each question types. The other two conditions are the question amounts of each difficult and the question amounts of each knowledge point in the test paper. The first condition must be met, and the other two conditions can be some discrepancy, that is to say the final test paper do not require strictly satisfy the last two conditions.

At present, the commonly used method of test paper generating has the following three kinds.

Random extraction method: In the process of the test paper generation, one question is selected randomly from the Item bank according to the current constraint, check the questions whether meet the constraints of the objective function, if not satisfied, then choose a new question for new trial, until the test paper that meet the users requirement was generated. If unable to produce a test paper to meet user demand, the test paper generation process failure. The research in this area can be found in the literature [1].

Retrospective testing method: The states of each step of the random selection method are recorded. When the search fails, release the state of the last record, then transform to a new state according to a certain rules. Through continuous retrospective testing until the test generation is finished or returned to the start point [2]. If the state and question quantity are less, the success rate of this conditional depth-first algorithm is better. But for the practical application, the randomness of questions selected is lack, and the test paper generation time is too long. The memory usage of this method is relatively large, and the program structure is relatively complex. However, retrospective testing can guarantee that a feasible solution can be found as long as the solutions to meet the conditions existing in the item bank.

Genetic algorithm: Genetic algorithm is a kind of easy to operate, parallel search algorithm for global optimization through the simulation of natural biological evolution and natural selection mechanism. The solving process will be expressed as the process of “chromosome” survival of the fittest. Through the evolution of generations of "chromosome" group (population), including the selection, crossover and mutation operations, eventually converge to the individual that is the most adapted to the environment, thereby obtaining the optimal solution or satisfactory solution. According to the basic idea of the genetic algorithm, the test paper generating algorithm which can solve the multiple constraints combination optimization problem in the test paper generating, and with a high success rate of test paper generating in a relative short time has been designed. There are a lot of researches in this area [3, 4, 5, 6].

This paper will give an affective genetic algorithm for the test paper generating.

II. THE OVERALL IDEA OF THE ALGORITHM

On the whole, the algorithm can be described as follows.

Step1. Set the parameters of the test paper. The parameters may include question types and the amount of questions of each type in the test paper, the point’s distribution among difficulties, and the point’s distribution among knowledge points.

Step2. Read the question messages from the item bank. The messages include the amounts of each type, the difficulties, the knowledge points and the exposure times of each question.

Step3. Initial population, many individual solutions are randomly generated to form an initial population. The individual is expressed as a gene and a two-dimensional array is used to express the gene. One row of the array represents a question type and one element of the row is the question number of that type. This is easy to realize in Java.

978-1-4673-4463-0/13/$31.00 ©2013 IEEE

Step4. Evaluate the fitness of each individual in that population. The factors that influence the fitness include the difficulties, the knowledge points and the exposure times.

Step5. Repeat this step until exceed the generation limit. The new generation was generated through the operation of selection, crossover and mutation. Then evaluate the individual fitness of the new individuals.

The flow chart of the algorithm is shown in figure 1.

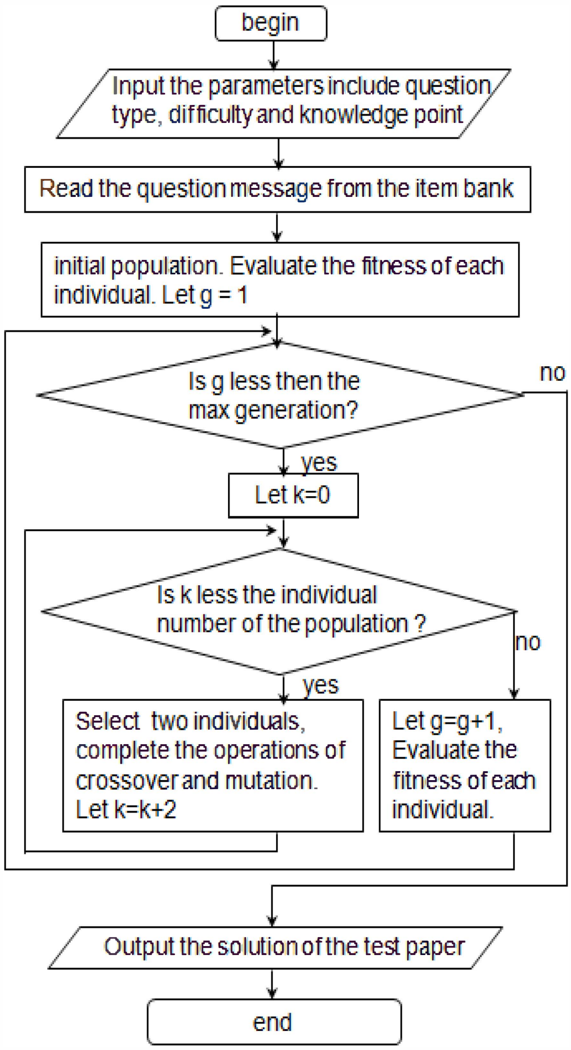

Figure 1. The flow chart of the algorithm

III. CHROMOSOME ENCODING AND FITNESS CALCULATION

Chromosome encoding and Fitness calculation are the foundations of the genetic algorithm.

A. Chromosome encoding

In order to realize the mapping of solution space to the genetic algorithm space, the solution space of the problem must be coded at first.

We use integer to encode the solution space in this paper. A two-dimensional array is used to present the code as in figure 2. Each element of the array presents a question number of the test paper. The question numbers of one question type were arranged in one row. Assume there are m question types in the test paper, so the code array has m rows. Each row represents a question type.

$$\begin{bmatrix} a_{11} & a_{12} & \cdots\cdots & a_{1k_1} \\ a_{21} & a_{22} & \cdots\cdots & a_{2k_2} \\ & \cdots\cdots\cdots\cdots & & \\ a_{m1} & a_{m2} & \cdots\cdots & a_{mk_m} \end{bmatrix}$$

Figure 2. Chromosome code

Suppose there are k_1 questions of type 1, k_2 questions of type 2, …, k_m questions of type m in the test paper, so there are k_1 elements in the first row, k_2 elements in the second row, …, and k_m elements in the last row. The element a_{ij} represents the number of the j^{th} question in the i^{th} question type. It is important to note, that the number of elements of each row is different.

B. Fitness calculation

The selection of fitness function is the key to the intelligent test paper generation algorithm. The fitness functions that we used to use measure the importance of various constraints in order to obtain maximum importance of comprehensive index or minimum error of comprehensive index.

In general, the fitness function is determined by and transformed from the objective function. In genetic algorithms, the advantages and disadvantages of individual are distinguished by the values of fitness.

The main constraints of generating the test paper include total scores, total times, question types, knowledge points, and difficulties and so on. The initial population that the genetic algorithm introduced in this paper is required to meet the parameters of total score, total time, and question types, and the populations of any generation in the process of evolution are always maintained to meet the requirements. Therefore the fitness function should not contain these factors.

We take three factors (knowledge point, difficulty and exposure time) as the example to introduce the fitness function.

First, we discuss the influence of knowledge points to the fitness function.

Suppose there are m knowledge points that we have set up in the test paper, and the scores of each knowledge point were z_1, z_2, z_3, …, z_m. If the actual scores corresponding to these knowledge points in an individual of the population were z_1', z_2', z_3', ..., z_m', then the total knowledge point score's difference between the test paper that we set up and the actual individual can be calculated as follows:

$$F_z = \sum_{i=1}^{m} |z_i - z_i'|$$

Obviously, the smaller the value of the F_z is, the closer the individual is to the solution.

Then, we discuss the influence of difficulties to the fitness function.

Similarly, suppose there are n difficulties we have set up in the test paper, and the scores of each difficulty were d_1, d_2, d_3, …, d_n. If the actual scores corresponding to these difficulties in an individual of the population were d_1', d_2', d_3', ..., d_n', then the total difficulty score's difference between the test paper that we set up and the actual individual can be calculated as follows:

$$F_d = \sum_{i=1}^{n} |d_i - d_i'|$$

At last, we discuss the influence of exposure times to the fitness function.

The total exposure time of an individual is the sum of each question's exposure time of the individual. The calculation of the total exposure time is relatively simple. Suppose the exposure time of the j^{th} question of the i^{th} question type is b_{ij}, then the total exposure time of the individual can be calculated as follows:

$$F_b = \sum_{i=1}^{m} \sum_{j=1}^{k} b_{ij}$$

In the above formula, m is the number of question types in the test paper, and k_i is the number of questions of the i^{th} question type.

Of course, we expect the small value of the total exposure time.

Traditionally, we should max the value of fitness function of genetic algorithm, so we will use the sum of the reciprocal of the three values that we obtained above as the fitness function.

The fitness function is defined as follows：

$F = w_1*1/(F_z+1) + w_2*1/(F_d + 1) + w_3*1/(F_b+1)$

In witch, $0 < w_i < 1$, (i=1, 2, or 3) and $w_1+w_2+w_3 = 1$.

w_1, w_2, w_3 are the weights of the three factors. They can be set according to the importance of each factor.

For example, if the user thinks the knowledge point distribution in the test paper is the important factor, the weight w_1 should be set a lager value, and the other two weights should be set smaller values.

In order to avoid dividing by 0, 1 is added to the denominators of every item in the above formula. Obviously, the greater the fitness is, the better the individual is.

IV. Initial Population

Individuals (each corresponding to a test paper) are obtained by select questions from item bank randomly. A certain number of individuals form the initial population. In the process of extracting questions, it must be guaranteed that the total scores, and the number of questions of each question type of each individual are meet the test paper requirements.

V. Genetic Operator

New generation is obtained from the last generation by the operations of selection, crossover and mutation.

A. Selection

Currently the selection operator has many algorithms of different type[7]. The algorithms include roulette wheel selection, stochastic universal sampling, stochastic tournament model, expected value model, deterministic sampling and remainder stochastic sampling with replacement etc. The following will briefly introduce some of them.

1) Roulette Wheel Selection

This is the first selection method presented by Holland in genetic algorithm. Because of the simple and practical, this selection method has been widely used.

It is a selection method based on percentage, the possibility of descendants retained is depends on the proportion of each individual fitness. If the population size is m and the fitness of individual i if F_i, the selected probability of individual k is expressed as

$$P_k = F_k / \sum_{i=1}^{m} F_i \quad (k=1,2,\ldots,m)$$

The roulette is used to select two individuals from the last generation. The greater the fitness is, the larger probability the individual can be selected.

Roulette is a commonly used random selection method, similar to the roulette in bet game. The individual fitness were conversed to selection probability proportionate. According to the ratio of the individual, a disk is proportionally divided into sectors. After the disc has been rotate, the individual that corresponding to the sector which the pointer stops was selected. Obviously, the greater the fitness of the individual is, the area of the corresponding sector in a disc is bigger, the chances of being selected is more.

2) Stochastic Universal Sampling

This method is similar to the roulette wheel selection method. It can be regard as an improved version of the roulette method. The characteristic of this method is that only need one spin of the roulette wheel.

In this method, the rotating pointer number is same to the population scale, and the distribution of the rotating pointers are uniform.

3) Stochastic Tournament Model

The individual that has the largest fitness among a few individuals has been selected to genetic to the next generation.

The number of the few individual that join compared each time is called the league size N. Generally N equals 2.

The process is as follows:

(1) The N individuals are selected randomly from the population, then compare the fitness values of the N individuals. The individual which has the highest fitness genetic to the next generation.

(2) After repeat the above process m times, m individuals of the new generation were obtained.

4) Expected Value Model

Expected value method is also called no playback randomly selected method. The individual was randmly selected according to the survival expectation of each individual in the next generation.

The process is as follows:

(1) Calculate the expected survival number of each individual in the next generation.

$$N_k = mF_k \Big/ \sum_{i=1}^{m} F_i \quad (k=1,2,\ldots,m)$$

(2) If an individual were selected to participate in the crossover operation, its expected survival number minus 0.5. If an individual was not selected to participate in the crossover operator, its expected survival number minus 1.0 in the next generation.

(3) With the selection process, if an individual's expected survival number less than 0, then the individual will not have the opportunity to be selected.

5) Roulette Wheel Selection in this paper

The Roulette wheel selection is used in our algorithm. In the following paragraph, the selection algorithm is described in details.

Suppose the population size is m and the fitness of the i^{th} individual is F_i. First of all the individuals are sorted according to their fitness values from big to small.

The lower 10% fitness individuals are replaced by the higher 10% fitness individuals and the higher 10% fitness individuals are direct selected to the next generation. And then the roulette wheel selection is used to finish the selection.

Set parent population $A=\{a_1, \ldots, a_m\}$, where each individual fitness is F_i. The initial state of the next generation population is $X=\{\ \}$. The process of selection operator is described as follows.

(1) All the individuals are sorted descending according to their fitness value. The population after sorted is $B = \{b_1, \ldots, b_m\}$, where $F_i > F_{i+1}$.

(2) The lower 10% fitness individuals are replaced by the higher 10% fitness individuals. Now the population is $C=\{c_1, \ldots, c_m\}$. And bring the higher 10% fitness individuals to the next generation. Add these individuals to the next generation population X.

(3) Calculate the tatol value of individuals in the population C.

(4) Calculate the selected probability of each individual.

$$P_k = F_k \Big/ \sum_{i=1}^{m} F_i \quad (k=1,2,\ldots,m)$$

(5) Use the roulette wheel selection to selecte the rest 90% individuals.

(6) Add the 90% individuals to X to form the new generation.

B. Crossover

According to the given cross probability p_c, the crossover operations were applied to the two selected individuals. One row in the two-dimensional array which expressed genes is corresponding one question type, so each line is crossed individually in order to ensure that the question number of each type is unchanged. Single point crossover is applied to each row of the two-dimensional array. If duplicate codes were appeared in an individual, the original code should be retained.

For example, suppose there are three question types, and 10,5,5 questions in each type. Crossover operator is applied to the following two individuals a and b. The underlined codes are the cross section.

$$a \begin{bmatrix} 5 & 2 & 23 & 8 & 18 & 54 & \underline{32} & \underline{25} & \underline{76} & \underline{15} \\ 2 & 23 & \underline{54} & \underline{38} & \underline{17} & & & & & \\ 25 & 16 & 18 & \underline{8} & \underline{38} & & & & & \end{bmatrix}$$

$$b \begin{bmatrix} 19 & 21 & 29 & 6 & 25 & 56 & \underline{38} & \underline{31} & \underline{75} & \underline{13} \\ 25 & 28 & \underline{14} & \underline{3} & \underline{27} & & & & & \\ 27 & 12 & 48 & \underline{35} & \underline{3} & & & & & \end{bmatrix}$$

After crossover operatition, two new individuals a′ and b′ were obtained as follows.

$$a' \begin{bmatrix} 5 & 2 & 23 & 8 & 18 & 54 & 38 & 31 & 75 & 13 \\ 2 & 23 & 14 & 3 & 27 & & & & & \\ 25 & 16 & 18 & 35 & 3 & & & & & \end{bmatrix}$$

$$b' \begin{bmatrix} 19 & 21 & 29 & 6 & 25 & 56 & 32 & \underline{25} & 76 & 15 \\ 25 & 28 & 54 & 38 & 17 & & & & & \\ 27 & 12 & 48 & 8 & 38 & & & & & \end{bmatrix}$$

As the first line of individual b' has a duplicate code 25, restore it to the original code 31.The final individual b″ is obtained.

$$b'' \begin{bmatrix} 19 & 21 & 29 & 6 & 25 & 56 & 32 & \underline{31} & 76 & 15 \\ 25 & 28 & 54 & 38 & 17 & & & & & \\ 27 & 12 & 48 & 8 & 38 & & & & & \end{bmatrix}$$

C. *Mutation*

According to the given mutation probability p_m, the mutation operations were applied to the two individuals respectively. If the mutation is happened, the element that will be mutated in the two-dimensional array is randomly selected. According to the row's number of the element, one question of the same type is randomly selected from the item bank. It is necessary to ensure that the question seleted from item bank does not repeat with the questions already in the individual. The two individuals of the new generation were produceed .

Repeat above process until the individual number of the new generation reaches the predetermined number.

VI. SUMMARY

We have presented a genetic algorithm of test paper generation in this paper. Chromosome coding, fitness function, and genetic operator are the main contents we discussed. This algorithm has a good performance for a medium scale item bank.

In practice, the values of some parameters such as weights of each factor in the fitness function, the cross probability p_c and the mutation probability p_m are very important. So we should take different parameter values in different item bank. The factors that influence the parameter values include the scale of the item bank and the distribution of question type, difficult and knowledge point.

REFERENCES

[1] Chen Zelin, Zang Qingbiao, Test bank design and algorithm of test system based on java[J], Journal of Chongqing University of Technology (Natural Science), vol. 24(2010), No. 3, 48-55, (In Chinese).

[2] Wang Meng, Jin Hanjun, Wang Xiaorong, Research on set at random algorithm in intelligent generating test paper [J], Computer Engineering and Design, Vol.27(2006), No.19, 3583-3585 (In Chinese).

[3] Yang Luming, Chen Daxin, Application of an improved genetic algorithm in composing a test paper[J], Computer and Digital Engenering, Vol.32(2004), No.5, 76-79 (In Chinese).

[4] Zhang Duan, Su Jianmin, Research of the intelligent test paper composition based of the genetic algoritnm in the construction of the examination database [J], Techniques of Automation & Application, Vol.29(2010), No.1, 41-43(In Chinese).

[5] Yin Lin, Research on the application of genetic algorithm in auto composing test paper[J], Journal of Langfang Teachers College(Naturnal Science Edition), Vol.11(2011), No.3, 26-28 (In Chinese).

[6] Xiao Lian, Cui Duwu, Design and implemention of creation system for test paper based on genetic algorithm[J], Computer Applications, Vol.28(2008), No.5, 1362–1364 (In Chinese).

[7] Yang Ping, Zheng Jinhua, Comparison and research over genetic selection operators [J], Computer Engineering and Applications, Vol43(2007), No.15, 59-65 (In Chinese).

[8] Huang Baoling, Application of Adaptive Genetic Algorithm in Intelligent Test Paper Composition[J], Computer Engineering, Vol.37(2011), No.14, 161–163 (In Chinese).