

Research On Intelligent Test Paper Generating Strategy Based On Genetic Algorithm And BP Neural Network

Chao Liu, Yan Hu

College of Computer Science and Technology, Wuhan University of Technology, Wuhan, China

Keywords: genetic algorithm; BP neural network; intelligent test paper generation

Abstract

The prevalence of MOOC (Massive Open Online Course), while providing us with a new efficient way of learning, demands an evolved online test system for certificating attended students properly. As examination of each course still being the simple and powerful way to check out, generating test paper as the core of exam system, is particularly important. Therefore, we present an enhanced method based on BP neural network optimizing the weight coefficient of objective function in paper generating model. After the Restructuring of mutation operator, and adjusting of crossover operator with the adaptive function, the performance of genetic algorithm modified for efficient test paper generating has been improved further. Experimental results show that the improved algorithm presented in this paper is efficient for the problem of genetic algorithm in premature and local optimization.

1 Introduction

MOOC^[1] is the abbreviations of Massive Open Online Course, which is a free web-based distance learning program aimed at unlimited participation and open access via the internet. As these courses are designed for the participation of large numbers of geographically dispersed students, students are free to decide their venues and ways to learn according to their own schedule and demands. A MOOC also removes other constrains and limitations in traditional face-to-face courses, such as age, diploma, or major, and focus on providing education for participants' good only, which may enable certification, employment or further studies. Currently, MOOC platforms have launched a certificate authentication features. Tsinghua University's online school, for instance, will send a course's certificate to the participant who complete it. After watching the lecture videos and finishing other learning materials, students should take the quizzes and pass final exam to earn the certificate. With the rapid development of information technology, and constantly improvement of the online exam, MOOC certificates will be widely recognized by society in the future. Intelligent Test Paper composition, as the core of online exams, demands increasingly better performances with high efficiency and quality of automatically test paper generating, which however is decided by the design of its algorithm. The so-called Intelligent Test

Paper composition refers to the test paper automatically generated by a computer as much as possible to meet the requirements of a certain test from the library. The general approaches^[2] are random algorithm, backtracking heuristics, genetic algorithm, etc. The random algorithm and backtracking heuristics do not have the intelligence and test paper generations often fail especially quantity of questions. Genetic algorithm is a computational model for simulating the biological evolution process, with simple, strong robustness, global optimization and other characteristics, and it is suitable for the intelligent test paper composition. But there are obvious deficiencies of Genetic algorithms, such as ahead convergence and short of climbing ability. Given this state of affairs, we present a more practical modified genetic algorithm with a reconstruction mutation operator, and experiments show that the modification can improve the performance of intelligent test paper composition.

2 Mathematical model of generating test paper

Intelligent test paper is composed automatic from the generated test paper database with several constrains for high test quality, according to the requirements of teachers and teaching. The constraint conditions are the types, scores and difficulty of questions, and also the discrimination, knowledge points, reference times and test duration, etc. These constraints are viewed as sets of attributes, and each test question would have a certain set with n attributes. With m questions selected according to test constraint conditions, a test paper would be successfully generated from the test database. And this paper can be presented as $m \times n$ matrix D , we have

$$D = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$$

Figure 1: Attribute matrix of test questions.

Here we use m to represent the number of questions in this test paper, and n as the number of attributes in the set of a certain question. Considering of the efficiency and the essence of this problem, we just take 8 parameters into consider, and denote them as follows: question types a_1 , question scores a_2 , difficulty a_3 , discrimination a_4 , knowledge points a_5 , time duration a_6 , last selected time a_7 and item exposure a_8 . Specific constraint conditions are as follows:

(1) question types Q :

$$Q_j = \sum_{i=1}^m d(a_{i1})a_{i2} \quad (1)$$

Q_j refers to the total score of question type j . When $a_{i1} = j$, $d(a_{i1}) = 1$; when $a_{i1} \neq j$, $d(a_{i1}) = 0$.

(2) total score S :

$$S = \sum_{i=1}^m a_{i2}$$

Given by users, default as 100.

(3) test paper difficulty P :

$$P = \sum_{i=1}^m \frac{a_{i2}a_{i3}}{S} \quad (3)$$

(4) test paper discrimination K :

$$K = \sum_{i=1}^m \frac{a_{i2}a_{i4}}{S} \quad (4)$$

(5) knowledge points Z :

Z_j refers to the total score of knowledge point j .

$$Z = \sum_{i=1}^s \left(\left| Z_j - \frac{S}{s} \right| \right) / S$$

$$Z_j = \sum_{i=1}^m d(a_{i5})a_{i2}$$

When $a_{i5} = j$, $d(a_{i5}) = 1$; when $a_{i5} \neq j$, $d(a_{i5}) = 0$. The number of knowledge points is s .

(6) time duration T :

$$T = \sum_{i=1}^m a_{i6} \quad (7)$$

(7) The last selected time is to ensure that the problem won't be selected repeatedly in a certain period of time. In order to improve the efficiency of pumping problems, you can filter out the unqualified questions based on this parameter.

(8) Item exposure means the total times of a question being pumped out. Exposure value defaults to 0, and it will automatically add 1 when the question is selected.

3 Objective function optimization

Objective function

$$E = \sum_{i=1}^n f_i w_i \quad (8)$$

f_i refers to the deviation between attribute value and the expected value. In order to prevent them from offsetting each other, f_i takes the absolute value. w_i is the weight coefficient of the i -th attribute, and can be given according to the

importance of different attributes. E is the sum of $f_i \times w_i$. The value of E should be as small as possible.

The importance of the attributes is different, so their weight coefficients are different. The weight coefficients used to be determined by experienced teachers according to their experience. Now we can use BP neural network^[3] for training to determine the weight coefficients. BP neural network training process is as follows:

- (1) Input signals $X = (x_1, x_2, \dots, x_m)$ and desired output signals $Y = (y_1, y_2, \dots, y_n)$.
- (2) Calculate the actual outputs $Z = (z_1, z_2, \dots, z_n)$.
- (3) The deviations between the target values and the actual outputs

$$|Z - Y| = \sqrt{\sum_{i=1}^n (z_i - y_i)^2} \quad (9)$$

If $|Z - Y| > \varepsilon$, go to the next step, otherwise return (1).

(4) Adjust the weight coefficients:

$$w_i(t+1) = w_i(t) + \sum_{j=1}^n \frac{\partial E}{\partial z_j} \frac{\partial z_j}{\partial w_i} \quad (10)$$

$$= w_i(t) + \sum_{j=1}^n (y_j - z_j) r_{ij}$$

- (5) According to new weight coefficient to recalculate network outputs, and return (3) to calculate the deviations again. If $|Z - Y| < \varepsilon$, then stop, otherwise return (1).

After the training, the weight coefficient matrix is $w = (w_1, w_2, \dots, w_n)$.

4 Improvement and implementation of genetic algorithm

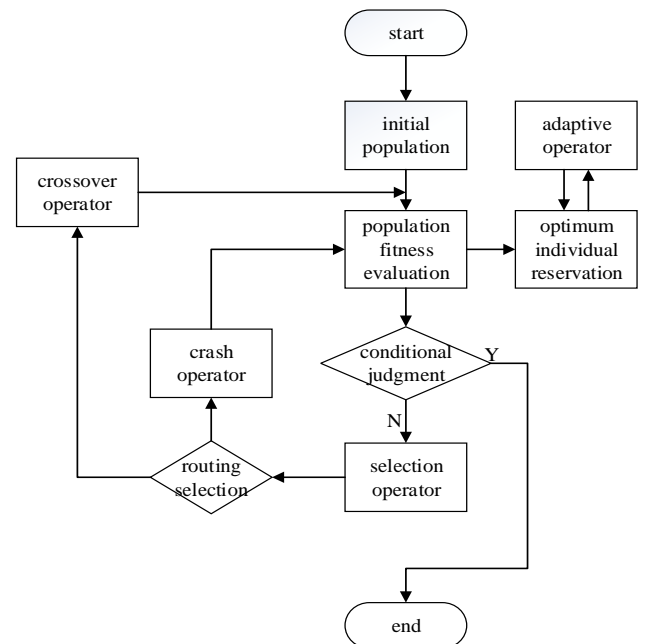


Figure 2: Algorithm flow.

Genetic algorithm^[4] is a global optimization^[5] search algorithm. There are some problems in premature and poor local search capability. For these problems, a reconstruction mutation operator is proposed. It consists of two parts: crash operator and adaptive operator. Use the crash operator to solve premature in the genetic algorithm, and use adaptive operator to enhance the local search capability of genetic algorithm. At the same time, the cross rate is adjusted adaptively. The users are generally not interested in population size, cross rate, mutation rate, etc. As little as possible or no parameter input^[6]. The improved algorithm flow is shown in figure 2.

4.1 Chromosome encoding

The traditional binary encoding can't visually show the problem of test paper generation. Binary encoding requires encoding and decoding operations. With the increase of the examination question database, it needs more space. A real sub-encoding method was given to meet the test paper problems. A paper is mapped to a chromosome and a gene corresponds to a question

type	multiple choice										short answer	
id	2	16	19	23	34	41	53	67	79	94	9	16
type	fill in the blanks					true or false					program	
id	3	7	19	34	41	4	11	31	47	71	11	31

Table 1: Real sub-encoding.

4.2 Initial population

N individuals were randomly selected to form initial population. N directly determines efficiency of genetic algorithm. If the value of N is larger, the search scope is wider, but corresponding operation time will be longer. If the value of N is smaller, the search scope becomes smaller, although the computation time becomes shorter. Considerations of efficiency and performance, the initial population is generally 80-200. N is 100 in this paper.

4.3 Individual fitness

Individual fitness values are calculated by the fitness function. Fitness function was as follows:

$$F = \frac{1}{1 + E} \quad (11)$$

By the above formula, the minimum value problem was transformed into the problem of finding the maximum value.

4.4 Termination conditions

- (1) During certain iteration, if the fitness value is more than or equal to expected value of users, the test paper generation is successful.
- (2) Iteration number reaches the maximum, the test paper fails.

4.5 Optimum individual reservations

Optimum individual reservations mean to retain the individual with the largest fitness value in the population. The optimum individual was reserved first during certain iteration. It was guaranteed not to participate in genetic operation. The adaptive operator was used for its random mutation operation. If the individual fitness value after the mutation was larger, it replaced the original optimum individual. Loop, and then increase the local search scope.

4.6 Selection operations

Selection operations mean select some chromosomes from their parents according to some criteria or methods after constructing the fitness function. In this paper, roulette method.

$$P_i = F_i / \sum_{i=1}^N F_i \quad (12)$$

N refers to the population size. F_i refers to the fitness value of the i -th individual. P_i refers to the probability that the i -th individual is selected.

4.7 Routing selections

In order to accelerate the convergence rate of the algorithm, routing selections were added to the genetic algorithm. The routing selections were used to balance the proportion of mutation operator and crossover operator in genetic algorithm. Its control parameter is expressed by S . The concrete description is as follows:

$$S = \begin{cases} \text{cross-rate}=70\%, \text{mutation-rate}=30\% & 0 \leq x_{total} < 35 \\ \text{cross-rate}=30\%, \text{mutation-rate}=70\% & x_{total} \geq 35 \end{cases} \quad (13)$$

x_{total} refers to the cumulative iteration sum of the optimum individual without updating for a long time. After a lot of experiments, x_{total} was 35 more scientific.

4.8 Crossover Operations

In order to ensure that crossover operations did not affect distribution of question types, crossover operations were set in the segment. Single point crossover was performed in the segment, and the whole chromosome was multi-point crossover. Cross rate p_c directly affects the performance of genetic algorithm. A method of determining p_c adaptively was presented in this paper.

$$p_c = \begin{cases} p_{c1} - \frac{(p_{c1} - p_{c2}) \times (f_h - \bar{f})}{f_{max} - \bar{f}}, f_h \geq \bar{f} \\ p_{c1}, f_h < \bar{f} \end{cases} \quad (14)$$

\bar{f} refers to the average fitness value of current population. f_h refers to the larger one of the two for crossover operations. f_{max} refers to the maximum fitness value of current population. $p_{c1}=0.9, p_{c2}=0.6$.

4.9 Mutation operations

A reconstruction mutation operator was proposed. It consists of two parts: crash operator and adaptive operator. Use the crash operator to balance the proportion of mutation operator and crossover operator in genetic algorithm. Experiments show that the critical period of population evolution is in the middle period.

$$p_m = \begin{cases} K \times \frac{1}{200 \times \sqrt{2\pi}} e^{-\frac{(x-300)^2}{2 \times 200^2}}, & 0 \leq x_{total} < 35 \\ K \times \frac{1}{200 \times \sqrt{2\pi}}, & x_{total} \geq 35 \end{cases} \quad (15)$$

K determines the steepness of above function. K is 45 in this paper. Experiments show that when the optimum individuals are not changed for 35 generations, it is necessary to increase the mutation rate by intervening. When $x_{total} < 35$, its search scope is in line with the trend of the function. Mutation operations were in the segment. Before mutation operations, other constraints can be used to filter. For example, the new test score is the same as the original.

5 Experimental results and analyses

Taking "Data Structure" course as an example, performance of the proposed genetic algorithm was verified by experiments. The examination question database was composed of 911 questions.

The BP neural network was used to train the network with MATLAB. After training, the final weight coefficients were $w = (0.3051, 0.2089, 0.0487, 0.1604, 0.1001, 0.1768)$.

The proposed genetic algorithm(PGA), simple genetic algorithm(SGA) and random algorithm (RA) were used to generate test paper 50 times. The experimental results are shown in table 2.

Algorithm	Averagetime /s	Accuracy /%
RA	69	74
SGA	42	90
PGA	31	98

Table 2: Algorithm comparison.

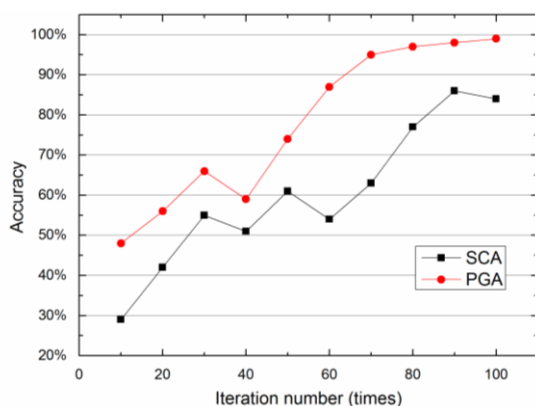


Figure 3: Iteration accuracy.

The proposed genetic algorithm was compared with simple genetic algorithm. Their results are shown in figure 3. In terms of time-consuming, the results of statistical simulation are shown in figure 4.

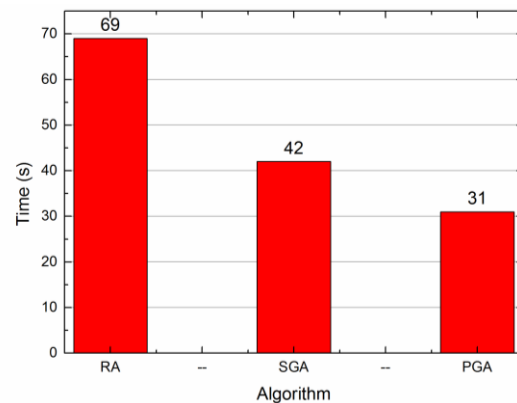


Figure 4: Algorithm time.

The experimental results show that the proposed genetic algorithm is superior to random algorithm and simple genetic algorithm in test paper generation.

6 Conclusions

The intelligent test paper composition is a typical application of multi-objective optimization problem, and our mathematical model in this paper is established according to the characteristics of test paper. BP neural network is used to optimize the weight coefficient of the index so that the test paper could meet the individual needs of teachers. The genetic algorithm adapted here is improved and optimized with the mutation operator being re-constructed, so that it could be more suitable to solve the problem of generating test paper. In addition, the routing selection was used to balance the proportion of mutation operator and cross operator. And it's eligible to claim that the proposed algorithm can solve multi-objective optimization problems and be applied to test paper generation.

References

- [1] Chen Xiao-geng, Wang Ding-ming. On the Development Process and Main Characteristic of MOOC[J]. Modern Educational Technology, 2013, 23(11):5-10.
- [2] Liu Shuo. Research on Composition Papers System Based on Improved Genetic Algorithms[D]. Jilin University, 2010.
- [3] Stuart J Russell, Peter Norvig. Artificial Intelligence: A Modern Approach (3rd Edition) [M]. Beijing : Tsinghua University press, 2013.
- [4] Wang Xiao-ping, Cao Li-ming. Genetic algorithm: theory, application and software implementation[M]. Xi'an: Xi'an Jiao Tong University press, 2002.
- [5] Goldberg D E, Samtani M P. Engineering optimization via genetic algorithm[C]//Proc.9th Conf. Electronic Computation, ASCE. 2015:471-482.
- [6] Harik G R, Lobo F G. A parameter-less genetic algorithm[J]. IEEE Transactions on Evolutionary Computation, 2015:523--528.