Research on Intelligent Test Paper Generation Based on Improved Genetic Algorithm

Huang Ming[1], Tang Ling[1], Liang Xu[1]

1. Software Technology Institute, Dalian Jiao Tong University, Dalian 116028
E-mail: dlhm@263.net

Abstract: Traditional algorithms of intelligent test paper generation have the disadvantages of slow convergence, low success rate and poor quality. In this article a new method of test paper generation is given，which is based on partition binary coding and improved genetic algorithm focused on improving the process of selection. This method uses independent question database. The new method is more efficient and easier to get over premature convergence than the traditional algorithms. It is proved by a number of experiments provided by this article.

Key word: improved genetic algorithm; intelligent test paper generation; mathematical model

1 Introduction

Along with the computer in the teaching domain application and development, the on-line test with the computer already become a kind of trend, therefore how to quickly select the question from the database, which satisfied users' every request of the examination paper, has become a question to be solved.

Test paper generation is a problem of multi-boundary combinatorial and optimization. The efficiency of test paper generation system and quality of test paper generation mainly depends on algorithms of test paper generation. At present there are several commonly methods used in test paper generation: random selection, backtrack, genetic algorithm(GA), and so on. GA can choose individual which satisfied condition from population, it has strong intelligence. Traditional GA mainly reproduces offspring by crossover. Traditional GA is prone to the phenomenon which is called "premature convergence". Therefore now there are many researchers applying improved GA to test paper generation, IGA can improve test paper generation operational efficiency. This paper proposed one kind of IGA, this IGA can improve population diversity, speed up convergence and convergence to the global optimal solution to solve the problem of intelligent test paper generation.

2The problem of intelligent test paper generation

2.1 mathematical model of intelligent test paper generation

Intelligent test paper generation can be described as a group of questions combination which satisfied the goal of test paper generation and selected from a certain amount database. Thus test paper generation becomes a problem of multi-objective optimization. As follows: to solve one test paper which is composed of m questions, each question has n attributes. This problem equals to construct a m×n goal matrix named S.

$$S=\begin{pmatrix} a_{11} \cdots a_{1n} \\ \vdots \quad\quad \vdots \\ a_{m1} \cdots a_{mn} \end{pmatrix}$$

Questions common attribute:(1)scores a_1,(2)difficulty factor a_2,(3)capacity level a_3, (4)estimate examination time a_4, (5) questions a_5, (6)times of test paper generation a_6. Goal matrix should satisfy constrained condition as follow:

(1)Total score= $\sum_{i=1}^{m} a_{i1}$;

(2)difficulty factor=1－ $\sum_{i=1}^{m} a_{i1} a_{i2}$ / total score;

(3)total time＝ $\sum_{i=1}^{m} a_{i4}$;

(4) $\sum_{i=1}^{m} c_{1i} a_{i1} = z_p$ (z_p is the score of p capacity), capacity level restraint: $c_{1i} = \begin{cases} 1 & a_{i3} = p \\ 0 & a_{i3} \neq p \end{cases}$;

(5)score of each question＝ $\sum_{i=1}^{m} c_{2i} a_{i1}$,

$c_{2i} = \begin{cases} 1 & a_{i5} = j \\ 0 & a_{i5} \neq j \end{cases}$, j is the number of question requested, question concludes: Judgment topic, List selected topic, Fills up the topic, Multi-selected topic, and so on.

During the process of test paper generation, n attributes of S decide each question of test paper, a_{ij} is No.j attribute of No.i question, i 1,2,….m;j=1,2,…,n.

2.2 Independent question database

For selecting test questions simply, quickly, this article uses independent question database to store various questions. According to times of test paper generation (a_6) in various question databases, we sort questions in various question databases(when firstly select question, questions order randomly generated). If a_6 of question's attribute is smaller, this question is easier to be selected next time, this can satisfy the quality of questions.

3 Design improved genetic algorithm

3.1 Confirm coding scheme

Using partition binary coding, and each section reflects a question, k is the quantity of questions, coding length is decided by the quantity of questions in the various question database. If there are t questions in various question database, the coding form is $b_1 b_2 \cdots b_t$. Randomly generating initial population, strings length is the same.

$$b_i=\begin{cases} 1 & \text{while question I is selected} \\ 0 & \text{while question I is not selected} \end{cases}$$

i=1,2,⋯,t $\sum_{i=1}^{t} b_i$ =m m is the quantity of questions which the test paper contained. $\sum_{i=1}^{r_1} b_i$ =m_1, $\sum_{i=r_1+1}^{r_2} b_i$ =m_2,…, $\sum_{i=r_{k-1}+1}^{r_k} b_i$ =m_k, $\sum_{i=1}^{k} m_i$ =m, $\sum_{i=1}^{k} r_i$ =t, $m_1, m_2, \cdots, m_k$ expresses the number of various questions in the actual test paper, r_1、⋯、r_k expresses the number of questions in various question database.

3.2 Generate initial population p(0)

In order to speed up the genetic algorithm the convergence rate and reduce the times of iterating, according to the total score request, we randomly generated initial population p(0). We made initial population p(0) satisfy the total score request, reduced the difficulty of the problem, improved the efficiency of the solution.

3.3 Confirm fitness function

We use the following forms of fitness function[1]:

$$F=1 \quad (1+\sum_{i=1}^{m} k_i \left| e_i \right|) \qquad i = 1,\dots,n$$

e_i is the error between the i factor of test paper generation and the goal of test paper generation, k_i is weight coefficient and k_i>0.

3.4 improve the genetic operators of intelligent test paper generation

3.4.1 selection operator

In order to realize convenience, according to fitness function, this paper calculated the value of each individual's fitness. According to the value of fitness, we sort the NO.t population. The t initial value is 0. We calculated generalized hamming H of neighboring individual. Hamming distance H[7] refers to the quantity of differences from corresponding position in the same length's two strings. For example:

$x_i(t)$=1011001010011,$x_j(t)$=1100101011001

Hamming distance H ($x_i(t)$, $x_j(t)$)=$\sum_{k=1}^{13} \left| x_{ik}(t) - x_{jk}(t) \right|$ =6.

Judged the relation of H and parameter d, if H<d, we replaces the individual whose value of fitness is smaller with the better individual of the father generation, otherwise retain these two individuals to new generation, and execute crossover.

3.4.2 crossover operator and mutation operator

To speed up searching efficiency of genetic algorithm, we effectively prevented it falling into their local optimum and protecting the fine individual of test paper. In this article, according to population's evolution situation, we use adaptive function to dynamic adjustment of crossover rate and mutation rate. The formula is as follows:

$$p_C=\begin{cases} p_{c1} - \dfrac{(p_{c1}-p_{c2})(f' - f_{avg})}{f_{max}-f_{avg}} \quad , & f' \ge f_{avg} \\ p_{c1} \quad , & f' < f_{avg} \end{cases}$$

In the formula: f ' is the one of individuals which participates in crossover and has bigger value of fitness; f_{max} is the biggest value of fitness in individual of former population; f_{avg} is the value of population average fitness in population. p_{c1}=0.9 p_{c2}=0.6.

$$p_m=\begin{cases} p_{m1} - \dfrac{(p_{m1}-p_{m2})(f_{max} - f)}{f_{max}-f_{avg}} \quad , & f \ge f_{avg} \\ p_{m1} \quad , & f < f_{avg} \end{cases}$$

In the formula: f_{max} is the biggest value of fitness in individual of former population; f_{avg} is the value of population average fitness in population. F is the value of fitness which individual will mutate. p_{m1}=0.1 p_{m2}=0.001.

In this paper, the process of crossover is as follows:

	before crossover	after crossover
$H1$	$a_1 a_2 \cdots a_i a_{i+1} \cdots a_t$	$a_1 a_2 \cdots a_i b_{i+1} \cdots b_t$
$H2$	$b_1 b_2 \cdots b_i b_{i+1} \cdots b_t$	$b_1 b_2 \cdots b_i a_{i+1} \cdots a_t$

If the crossover-points in the i gene section, the first i gene section maintains invariable, we exchange from the i+1 gene section. If crossover also occurs in the gene section, it is possible that identical gene section leads to repetition question. After crossover, we calculate the fitness of two individuals newly generated and compare them with their father individual, if their value of fitness is equal to their parents, we either delete them or retain them, and don't replay their father individual with them, let the new individual executes mutation.

Mutation rate decides the position which is randomly selected and negate it, in the group of this position, we forward or backward search the position whose value is opposite to the value of this position and then negate this position too. Such change may satisfy the request of question, and the quantity of questions.

3.5 Control parameters of intelligent test paper generation

The value of population scale N is 200, MaxGen of algorithm execution is 500.

3.6 Terminal condition

①It satisfies the user's request of test paper generation restraint or obtains the test paper which the user satisfied. ②The biggest value of fitness in the current population nearly equals to the biggest value of fitness in the each previous population. ③It achieves evolution algebra which is assigned.

4 Realize the improved genetic algorithm based on intelligent test paper generation

Confirm parameter: Max generation MaxGen, population scale N, crossover rate $p_{c,}$ mutation rate $p_{m,}$ user's request of test paper generation. The algorithm realizes process is as follows:

Randomly generate N initial population P(0) $P(0)=\{X_1, X_2, \quad , X_n\}$,gen=0;

Calculated the value of fitness and H which is from individual of P(0);

While(doesn't satisfy the termination criterion, gen<MaxGen&&the value of best individual's object function doesn't satisfy the request)do

{

According to times of test paper generation and selection strategy, generate offspring p(gen) which is selected from p(gen);

Generate new individual p(gen+1) by crossover;

Calculate the value of fitness of each new individual p(gen+1), delete the new individual whose value of fitness is the same to the father individual, retain the father individual which is not replaced;

Execute mutation;

According to selection strategy, to select individual whose quantity is equal to N from the population whose quantity is bigger than N but smaller than 2N.This N individual becomes the new population p(gen+1);

gen=gen+1;

}

Output best individual coding.

5 Test result analysis

In order to confirm the improved genetic algorithm the validity and the superiority, for the 800 questions of JAVA Programming Language, we separately establish 4 databases. List selected topic, Multi-selected topic, Fills up the topic, Judgment topic, each database has 200 questions. Each kind of questions has 5 kinds of difficulties. The total score of test paper generation is 100, the estimated time is 120 minutes, the difficulty factor is 0.7. For the simulation of intelligent test paper generation, we separately used this article algorithm and the traditional genetic algorithm. The simulation result is as follows:

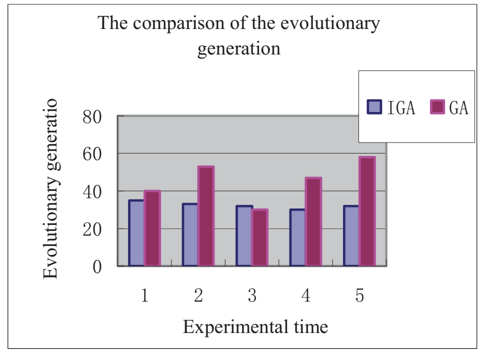

Fig. 5.1 The comparison of the Evolutionary generation between the two algorithms

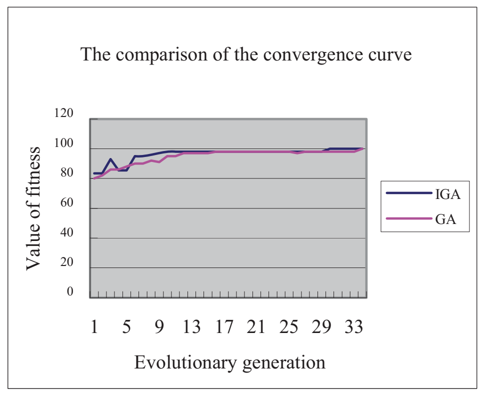

Fig. 5.2 The comparison of the Convergence curve between the two algorithms

By comparison above, we know that the algorithm proposed by this article can obtain the optimal solution. The final evolution generation and convergence rate are superior to traditional genetic algorithm, it confirms this algorithm to be feasible.

6 Concluding

This paper proposed an improved genetic algorithm which uses independent question database to store various questions and partition binary coding. During population initialization, the test paper satisfies the request of score. It reduced the difficulty of question, improved the efficiency of solving the problem, and unceasingly eliminated the similar individual in the evolutionary process, maximum reduced possibility of homogeneous population, unceasingly supplied the new individual, avoided the search space rapidly reducing. We executed crossover and mutation by using p_c,p_m of adaptive dynamic adjustment. It improved the overall optimization performance, could effectively solve the problem of intelligent test paper generation.

REFERENCE:

[1] Yang Luming, Chen Daxin. Application of An Improved Genetic Algorithm in Composing A Test Paper[J]. Computer&Digital Engineering, 2004,(5):77-78.

[2] WANG Shupei, YI Yeqing. Investigation of Test Paper Auto-generation Based on Improved Adaptive Genetic Algorithm[J]. Science Technology and Engineering, 2006,(4):468-05.

[3] LU Jing, ZHOU Chun-yan. Study of Hybrid Optimization Strategy Based on Genetic Algorithm[J]. Computer Technology and Development,2007:17(3):144-146.

[4] WEI Ping1, GAN Hai-guang2, XIONG Wei-qing. A Partheno_Genetic Algorithm Based on Evolution Stable Strategy for Test Paper[J]. Microelectronics&Comuter, 2005:22(1):105-109.

[5] WU Xiao-qin. Research on Intelligent Auto-generating Test Paper Based on Genetic Algorithms[J]. Journal of Hefei University(Natural Sciences), 2007,17(2):47-50.

[6] XI Wan-quan. Algorithm of Intelligent Assembling Examination Paper Based on Minimum Trace-cost[J]. Journal of Hunan University. 2004:31(6):172-174.

[7] WANG Li-fang, WANG Nan, LI Xin-hua. Research on Intelligent Test Paper Generation Based on Improved Genetic Algorithm[J]. Journal of North University of China.Natural Science Edition, 2006,27(4):288-291.

[8] WU Fei. On self-adaptive Genetic Algorithm to Solve the Problem of Test Paper Combination[J]. Journal of Chongqing University of Science and Technology(Natural Sciences Edition), 2007,9(2):132-140.