Application of Improved Genetic Algorithm in Automatic Test Paper Generation

Kui Zhang
Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial
College of Computer Science & Technology, Wuhan University of Science & Technology
Wuhan, China

Lingchen Zhu
Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial
College of Computer Science & Technology, Wuhan University of Science & Technology
Wuhan, China

***Abstract*—Automatic test paper generation system is to automatically generate papers by computer from test database with many constraint conditions according to requirements of teachers and teaching. It could greatly reduce teachers' work, and make the difficulty coefficient of test paper reasonable. The system plays an important role in reform of examination system. Traditional genetic algorithm uses binary code, but because the binary string is too long, it cannot control well the number of question types. However, the system with decimal code could avoid that problem. In the process of genetic manipulation, crossover operation takes subsection crossover, i.e. single point crossover within one question type. Therefore, the whole chromosome is multi-point crossover, which makes the result more reasonable. This system makes use of global optimization and fast convergence speed of genetic algorithm to design an intelligent algorithm for automatically generating test papers. We have established and described the chromosome structure of test paper and the fitness function, designed genetic operators, and completed corresponding genetic algorithm application software to realize the automatic generation of test papers. Experimental results show that the automatic test paper generation system based on genetic algorithm achieves optimization of efficiency and reasonability of difficulty coefficient of test paper.**

Keywords—Automatic test paper generation system; Genetic algorithm, Subsection crossover

I. INTRODUCTION

Automatic test paper generation system（ATPGS）aims for automation of examination system and it can fulfill different requirements from users by inputting different constraint conditions, such as the total score of the paper, the number or type of questions, distribution of knowledge points and difficulty coefficient etc.. In addition, it selects test questions from the database automatically to meet the requirements of the test paper. ATPGS can generate an objective test paper and improve the efficiency of test paper generation.

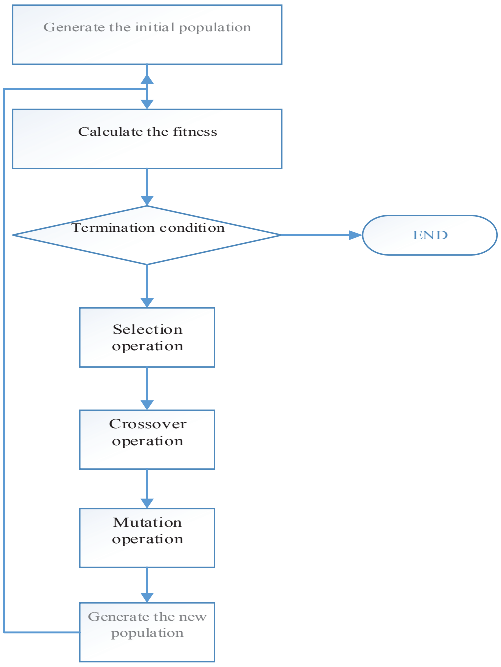
There are a variety of test paper generation algorithms in the existing examination system [1], such as random test method, backtracking test method and genetic algorithm method. Random test method has many advantages, such as test paper generation strategy intuitive and simple programming. However, it has not pre-process of constraint conditions, which leads to lack of rationality in the paper and make it difficult to control the difficulty coefficient of the test paper. Backtracking test method has high success rate of paper composition. But if there are larger numbers of paper test libraries, paper making would take a relatively long time, which reduces the working efficiency. This system takes advantage of global optimization and fast convergence speed of GA to design an intelligent algorithm for ATPGS. Genetic algorithm method is of high efficiency, shorter time and high success rate [2].

Traditional genetic algorithm (GA) uses binary code. If the number of questions in a test database is large, then the binary string is too long to control genetic operations. Therefore, most ATPGS uses decimal code[3], which is able to avoid this problem. Generally in the decimal coding, crossover points and mutation points are generated in the whole coding. It makes the number of question types changed. Hence, we have put forward to an improved genetic algorithm. We use segmented decimal coding which is generated according to the number of question types and the number of each question type. The crossover operation could take in each segmented decimal coding. Then the chromosome is expressed in the form of multipoint crossover, which makes results more reasonable[4]. In this paper, we have established the chromosome structure of test paper and the fitness function, and designed genetic operators.

II. BASIC THEORY OF GENETIC ALGORITHM

Genetic algorithm is a kind of random search method based on the biological evolution law i.e. (the survival of the fittest genetic mechanism)[5],[6]. It was first proposed by Professor J. Holland in 1975 in America. Its main characteristic is to operate directly on the structure of the object. There is no limit of derivation and continuity of function. It has built-in implicit parallelism and better global optimization capabilities. It uses probabilistic optimization methods, which is able to automatically obtain and guide the optimization of the search space, adaptively adjust the search direction, no the need determining the rules.

Genetic algorithm starts from a population which is a potential solution of the problem. A population is composed by a certain number of individual by encoding genes. Each individual is actually the chromosome with the characteristic entities. Chromosome, as the main carrier of genetic material is a collection of genes as internal performance which determines the external manifestation of individual shape and features, like the black hair is decided by a combination of genes in chromosome which control such the characteristics. Therefore, at the beginning of genetic algorithm, it is necessary to realize the mapping from phenotype to genotype, i.e. the coding work. In realistic world, the work of gene coding is very complex, so we tend to simplify it. After the initial population generating, better and better approximate solutions are generated from evolution of many generations. In every generation, individuals are selected by value of their fitness in problem field, and crossover operation and mutation operation are produced by genetic operators in natural genetics. Then new population is generated which is new solution of the problem. This process will result that the new population has more adaptation than previous one as natural evolution. The optimum individual of the last population is decoded and used as approximately optimal solution.The basic process of genetic algorithm is as follows:



a. Initialization: setting the generation counter of evolution t=0, set the maximum generation number of evolution as T, and randomly generate M individuals as the initial population P(0).

b. Individual evaluation: Calculate the fitness of each individual in population P(t) which is t generation.

c. Selection operation: act a select operator on the population P(t). The purpose is that the optimization individuals are inherited to the next generation directly. Selection operation is based on individual fitness evaluation.

d. Crossover operation: act a crossover operator on the population *P(t)*. The so-called crossover is an operation that part of structures of two parent individuals are replaced and reconstructed to generate a new individual. The core of genetic algorithm is crossover operator.

e. Mutation operation: act a mutation operator on the population *P(t)*. That is to change the value of some gene locus of individuals in the population. Therefore, after selection, crossover, and mutation operations of population *P(t),* the next generation of population is generated, denoted by *P(t1)*. Then *P(t)* is replaced by *P(t1)*.

f. Termination condition: if $t=T$, the evolution process is ended and the individual with the maximum fitness is the output solution. If not, then turn to the second step.

III. Improved Genetic Algorithm in Automatic Test Paper Generation

In automatic test paper generation, traditional genetic algorithms usually use binary encoding. When binary encoding is used, each item in the question database will be in the binary string, and "1" means that the item is selected, "0" means that the item is not selected. Such binary encoding has many disadvantages. First, such string is very long. Second, it is difficult to control the number of questions of different question types at the time of crossover and mutation of genetic operator. Third, the number of "1" is equal to question number n of test paper in each binary string of the initial population. After genetic manipulation (crossover and mutation), the number of "1" in the string may be more or less than n, and thus the solution becomes illegal, which have to be amended. Then, the corresponding calculation is performed to make the number of "1" in the string equal to n. Generally speaking, the process is more complex, greatly increasing the amount of computation. In addition, we have different requests for the attribute index of the generated paper. We hope that no errors are made in the test scores, and certain requirements are met for other attributes (such as type, difficulty coefficient, discrimination, answer time). Therefore, we have improved the traditional binary genetic algorithm as follows:

1) Coding

Genetic algorithm uses the decimal coding scheme. The value of genes is directly represented in item number. Item numbers of each type are gathered to make question types segmented. Each question type has its own table in the database. So even in the decimal string there have the same item number, they are not the same question.

2) Designing the initial population

The initial population of paper by the traditional method is completely randomly. But our method produces initial population randomly which meets requires of those such as the proportion of question types, total score, answer time, different points of knowledge and different chapters.

3) Selecting operator

The roulette wheel method is employed to make the operator more reasonable. The method determines the probability of the individual into the next generation depending on size ratio of the fitness.

4) Doing crossover operator

We adopt segmented decimal coding, which enables the segmenting of the single point crossover (cross section according to types) in the cross, and the chromosome is expressed in the form of multipoint crossover.

Based on the improved genetic algorithm for automatic test paper generation, the steps of the algorithm are as follows:

Step1: Chromosome encoding and the design of the initialize population

In the decimal encoding scheme, maps a paper into a chromosome, in which the genes are directly represented as item number, item numbers of each type are gathered to make question types segmented which could ensure that the total number of each type is unchanged. For example, to make up a paper of Basic College Computer, if there are 5 multiple choice questions, 5 bank fillings, 3 short answers, the chromosome encoding is:

(10、76、23、52、101 | 52、36、67、11、123 | 99、85、45)

Multiple choice Blank filling Short answer

The initial population is produced randomly which meets requires of those such as the proportion of question types, total score, answer time, different points of knowledge and different chapters. It marks the initial population meet requirements of test paper generation in the first. This way accelerates convergence of genetic algorithm and reduces the number of iterations. Because different kinds of questions are taken from the different tables, it is possible the same topic number will appear in the same gene series. But because they are of different types, it does not affect the set of test paper. The segmented decimal coding can overcome the disadvantages of binary code, that is, the search space is too large and the encoded length is too long. Therefore, the decimal coding does not need individual decoding and fast the speed of solution.

Step 2: Designing of the fitness function

Fitness function is indicators used to judge superiority of the individuals in the test group. Genetic algorithm using fitness value to guide the direction of search, it doesn't need continuous or differentiable fitness function, its other auxiliary information. Therefore, the difficulty coefficient of test formula is converted the fitness function. The formula is as follows:

$$Df=\sum Di \times Si/\sum Si$$

Where i =1, 2, ... n, n is the total number of test paper topic, Di and Si is the difficulty coefficient and the score of item i respectively, Df is the difficulty coefficient of the test, then objective function f=|EP-Df|, where EP is user's expectative difficulty coefficient. By conversion of the objective function f to the fitness function F=e-0.03f, the fitness function with weighted error can reflect the characteristics of test paper better. The error of constraints for the individual test is smaller, the value of fitness function is larger, and the generated paper is closer to the target paper.

Step 3: Doing selecting operation

Selecting operator decides whether the individual will be eliminated or copied in the next generation according to its superiority. By selecting, individuals with high fitness have more chances to survive. We adopt the roulette method, which is the most commonly and the most classic selection method in genetic algorithm.

Its implementation is: if population P is with the size of M. denoted by P = {a1, a2, …… am}, where i =1, 2, ... m, m is the total number of test paper topics, ai is the fitness value of i individual. The selected probability is ai/∑ai. The random function produces a random number between from 0 to 1. Then judge its location in the sequence of cumulative probability. In the sequence, find the index i of the first one whose cumulative probability is more than the random number. Then the ith individual is chosen.

Table Ⅰ gives fitness, selected probability and cumulative probability of 11 individuals. In order to select the better individuals, require multiple rounds of selection. Each round generates a uniform random number between 0 and 1, and the random number is used to determine the probability of the selected individual. Assuming the random number is 0.81 in the first round, and then the 6th individual is selected. Assuming the random number is 0.32 in the second round, the 2nd individual is selected. The rest can be done in the same manner. Therefore, in the round of from 3rd to 6th, assuming random numbers are 0.96, 0.01, 0.65, 0.42, and then 9th, 1st, 5th, 3rd are selected in turn. So the individual 1, 2, 3, 5, 6, 9 are selected in next population.

TABLE Ⅰ. Roulette Selection Probability Calculation

Individuality	1	2	3	4	5	6	7	8	9	10
Fitness	2.0	1.8	1.6	1.4	1.2	1.0	0.8	0.6	0.4	0.2
Selected probability	0.18	0.16	0.15	0.13	0.11	0.09	0.07	0.06	0.03	0.02
Cumulative probability	0.18	0.34	0.49	0.62	0.73	0.82	0.89	0.95	0.98	1.00

In general, through the selection strategy, individuals with high fitness have more chances to survive, and individuals with low fitness have more chances to eliminate.

Step 4: Doing crossover operation

The implementation of the crossover is: choose randomly two chromosomes to match in the population. Then generate a random number r in the range of [0, 1]. If r ≤ pc (according to experience, the value of pc is from 0.6 to 0.8, here we take 0.7), the crossover operation will be done. In the process of crossover operation, generate randomly a crossing point denoted by crossPoint. If the size of segment is L, the crossing point is in the range of [1, L-1]. Then exchange all genes of the right of the crossing point, i.e. [crossPoint+1, L], to get next generation. The next generation after crossover may have the same item to make it illegal. On this occasion, the repeated item must be replaced by other one which is not used in the segment string.

For example, S1 and S2 are two chromosomes to be crossed which have three segments. If the values of crossPoint are 2, 2, 2, Then the result of crossover is as follows:

Before crossover:

S1 (10 76 23 52 | 60 99 25 | 43 56 72 44 81)

S2 (20 35 55 17 | 54 62 78 | 85 98 66 21 45)

After crossover:

S1 (10 76 55 17 | 60 99 78 | 43 56 66 21 45)

S2 (20 35 23 52 | 54 62 25 | 85 98 72 44 81)

Step 5: Doing mutation operation

In genetic algorithm, probability of mutation operation is generally small.

The implementation of the mutation is: Firstly generate a random number r in the range of [0, 1]. If r ⩽ pm (according to experience, the value of pm is from 0.01 to 0.02, here we take 0.015), the mutation operation will be done. Then generate randomly a segment number f. If the size of segment f is L, then the mutation point P generated randomly should be in the range of [1, L]. Lastly select a different gene from the table correspondent to the segment number f. For example, if the mutation point in the 1st segment is 3, then chromosome S1 is mutated to S2 as follow:

Before mutation: S1 (10 76 23 | 52 10 99)

After mutation: S2 (10 76 43 | 52 10 99)

Step 6: Judging terminating condition

Record the number of generation during evolution, and determine whether the termination conditions are met. The terminating conditions are as follows:

Set maximum number of generation MAXG. When the number of generation is up to MAXG, generic algorithm stops. We set MAXG to 200.

According to degree of convergence. When the ratio of average fitness of current generation and that of previous generation is up to an acceptable range, genetic algorithm stops.

IV. Analysis of Experimental Results

In order to prove the improved genetic algorithm for automatic test paper generation system is superior to traditional, it made a comparison between the two, specifically the comparison of the final fitness value, the number of iterations and time of calculation.

TABLE Ⅱ. Comparison of General GA and Improved GA in ATPGS

Algorithm	Input						Output		
	MC	BF	T/F	SA	DP	DC	FV	IN	CT
General GA	30	20	10	25	15	0.1	2.620781	8	26s
	30	20	10	25	15	0.6	2.671181	15	29s
Improved GA	30	20	10	25	15	0.1	2.628681	22	61s
	30	20	10	25	15	0.6	2.671281	16	45s

From Table Ⅱ, MC representative of Multiple Choice, and Bf is Blank Filling, T/F is True-False, Sa is Short Answer, Dp is Design Problem, Dc is Difficulty Coefficient, FV is Fitness Value, IN is Iteration number, CT is Calculation Time. We could see that the number of iteration of the improved GA is more than the number of general GA in the same of input conditions. Therefor the improved GA can select more optimal individual by the iterative operation. And the improved GA has larger value of fitness than that of general GA. We could make out that the improved GA in ATPGS has more superiority and reasonability than general GA, but Calculation Time needs to improve.

V. Conclusion

Automatic test paper generation problem is a multi-objective optimization problem under certain conditions. Its constraint conditions cannot be described in mathematical form, so it is very difficult to use the traditional mathematical method to solve it. It can be proved that GA is one of the best approaches to dealing with this problem efficiently. The improved GA in ATPGS uses the decimal encoding and make segments according to the question types. Genetic operators are carried out at each segment to ensure that the number of each question type is unchanged. The segmented decimal coding overcomes shortcomings of binary coding, such as too large search space and too long code. It lessens decoding time of individual and improves the speed of the solution. The initial population is generated randomly which meets requires of those such as the proportion of question types, total score, answer time, different points of knowledge and different chapters. The improved GA speeds up the convergence of genetic algorithm and reduce the number of iterations. The crossover operator is used in each segment to make the whole chromosome multi-point crossover, which makes results more reasonable. Therefore, this paper provides a new method in ATPGS and could solve similar multi-objective constrained problem.

References

[1] Xiao lian, Design of Test Paper Generation System Based on Genetic Algorithm and implementation [J]. Computer Application, 2008, pp.1362~1364

[2] Zhu Guofeng, Application of genetic algorithm in automatic test generation [J]. Computer Knowledge and Technology, 2013, pp.1626~1629

[3] Lei Yingjie, Zhang Shanwen, Genetic algorithm toolbox and its application MATLAB [M].Xi'an Electronic and Science University press, 2014, pp. 206~320

[4] Yuan Changan, The principle and application of gene expression programming algorithm [M]. Science Press, 2010

[5] Mao Bingyi. Research on the Intelligent Test Paper Generation System [J]. Computer Engineering, 2002, pp.278~279

[6] Shi Feng, Wang Hui, Hu Fei, Yu Lei, Analysis of MATLAB intelligent algorithm in 30 cases [M]. Beihang University press, 2011, pp.106~246

[7] Ulrich Bodenhofer. Genetic Algorithms: Theory and Applications [M]. (3rd Ed). Linz: Johannes Kepler University, 2001, pp. 33