http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 13 (17): 2666-2673, 2014
ISSN 1812-5638 / DOI: 10.3923/itj.2014.2666.2673

Research on Intelligent Test Paper Based on Hierarchical and Self-Adapting Genetic Algorithm

[1]Weimin Zuo, [2]Xuyu Xiang, [1]Yuanyuan Fu, [2]Jiaohua Qin and [2]Xiaoyu Guo
[1]Department of Computer, Hunan City University, Yiyang, Hunan, 413000, China
[2]College of Transportation and Logistics, Central South University of Forestry and Technology, Changsha, 410004, China

Abstract: The generating test study is a research in constrained multi-object optimization. It is one of the key technologies in examination management systems. It relates directly to the efficiency and quality of the generating test paper. The test paper generation method suggested in this paper is based on hierarchical adaptive genetic algorithm. It is able to solve the problem of premature convergence or slow convergence in global optimization. On the one hand, the M subpopulations operate on adaptive genetic algorithm and save the intermediate result; then they operate with the top adaptive genetic algorithm until a satisfied paper is found. On the other hand, the minimum weighted mean square error model is used to establish the objective function and to inspect the error between the expectations and the actual value on types, knowledge topics, difficulty and the degree of differentiation of the test paper. It discusses also the error of the answer time, total score and luminosity. It improved the speed of generating test paper of the system. It avoided the problem of premature convergence which often appears in standard genetic algorithm. The high quality of paper generation and the good robustness generated in this algorithm can meet the practical needs of users.

Key words: Intelligent test paper, genetic algorithm, hierarchy, self-adapting

INTRODUCTION

The generating test paper is to optimize the multi-objects constraint. It is one of the key technologies in the examination management systems, thus it directly relates to the quality of test paper and the running time and space overhead. There are problems of time-consuming, low success rate and generated papers could not satisfy the need of actual test in traditional algorithms of generating test paper such as the priority strategy, random strategy, parallel strategy and backtracking strategy etc., (Zitzler and Thiele, 1999; Holland, 1975; Dorronsoro and Alba, 2006; Leung and Wang, 2000; Jerald *et al*., 2006; Huang, 2011). However, the genetic algorithm can meet the requirements of intelligent-generating test paper with characteristics of adaptive global optimization, intelligent search and good convergence (Martinez-Gomez *et al*., 2010). The random algorithm is also popular for its simpleness (Nebro *et al*., 2009; Ombuki *et al*., 2006). The system proposed in this study combined those two algorithms to pumping test paper generation.

Test theory (Golino, 2005; Huang, 2011) pointed that the study quality is related to the following aspects:

- Whether the test points of the paper cover the most topics to be examined
- Whether the types and numbers of the questions are reasonable
- Whether the profundity and the difficulty of the paper meet the requirements
- Whether the distinction degree of the paper is suitable, that is whether the paper can test the real level of the students
- Whether the test is fair, that is whether the test questions are frequent in the study process

Intelligent-generating test paper is to optimize the multi-objects constraint (Galante, 1996). The constraints are so many that it is hard to satisfy all. What's worse it can lead to generating test paper failure easily.

Genetic algorithm is a simulation of the natural biology evolution and the genetic mechanism of the biological evolution process. It is a method which

Corresponding Author: Xuyu Xiang, College of Transportation and Logistics, Central South University of Forestry and Technology, Shaoshan South Road, No. 498, Changsha, Hunan, 410004, People's Republic of China Tel: 86-13607493419

lsearches the optimal solution by simulating the natural biological evolution process. It was first proposed by Professor J. Holland of Michigan University in 1975 (Holland, 1975).

In application, the secondary constraints are relaxed to make sure that the importantly constraints are satisfied. In this paper, the following aspects of generating test study are taken into account:

- Determine the proportion of questions by types and knowledgetopics in the paper with the syllabus
- Determine the proportion of questions with different difficulty level in the study according to the test target
- Determine the questions proportion of each knowledge level in the study

MATERIALS AND METHODS

Genetic algorithm: The mathematical model of genetic algorithm can be expressed as:

$$GA = (C, E, P_0, M, \Phi, \tau, \Psi, T)$$

In this Equation, C is the individual coding method. Fixed length encoding is generally used in GA, E is the individual fitness function, p_0 is the initial total group, M is the number of total individuals in the group, Φ is the selection operator, Ψ is the crossover operator, T is the mutation operator, T is the algorithm termination condition. The program flow chart of the standard Genetic Algorithm is shown in Fig. 1.

Mathematical model of intelligent test paper: A hierarchical adaptive genetic algorithm is adopted for intelligent test paper in this model. The weighted least square error model is used to establish the mathematical model.

Paper constraint and attribute: It is supposed that A test paper consist of M questions, each question consist of N attribute values, thus the attribute values can constitute a matrix A of a m×n:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} \qquad (1)$$

where, A is the goal state matrix in solving the problem of intelligent-generating test paper, a_{ij} is the j attribute of the I question. The value range of I is (1 , 2 , . . . , m), a_{i1} is the

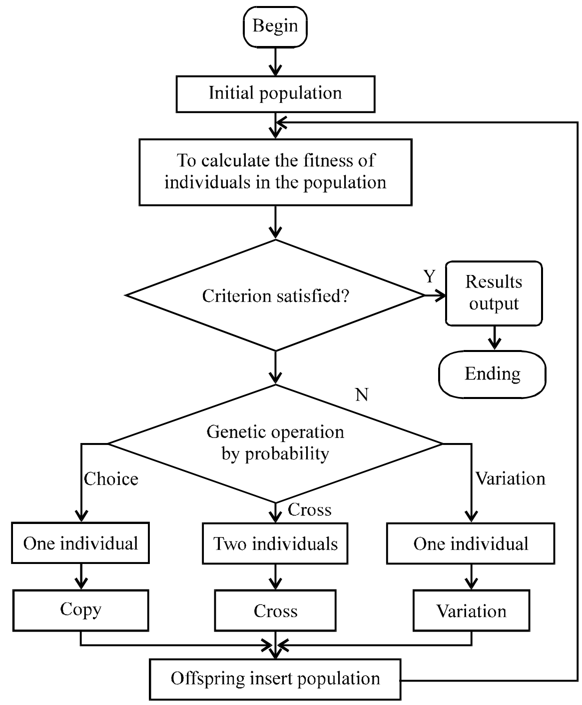

Fig. 1: Flow chart of the standard genetic algorithm

I score, a_{i2} is the type of I question, a_{i3} is the knowledge topic of the I question, a_{i4} is the difficulty level of I question, a_{i5} is the distinguish degree of I question, a_{i6} is the answer time of I question, a_{i7} is the total score of I question, a_{i8} is the exposure degree of I question.

In this system, the proportion of each tests column and each attribute value in matrix A put forward by users is defined as pumping test paper constraint (target value). The difficulty, discrimination, out of the value and the answer time metering constraint are taken the arithmetic mean values. Counting constraints of the types, chapters and knowledge topics are directly set by the users. The task of pumping test paper is to make each of the generated attribute value in the papers as close as possible to the target value.

The difficulty of the test is the difficulty degree of questions in the test. P represents the difficulty of a question; $\overline{S}$ represents the average score on this question of a student; S represents the full mark; the difficulty can be calculated as the following equation:

$$P = 1 - \frac{\overline{S}}{S} \qquad (2)$$

The smaller value of P is, the easier the question is; the bigger value of P is, the harder the question is.

The distinction degree of a paper means discriminability of the real level of a student. Distinction degree D can be calculated according to the following equation:

$$D = \frac{X_H - X_L}{N(H - L)} \quad (3)$$

In the equation, X_H is the total score of the high score group, X_L is the total score of low score group, N is 25% of the total number of students, H is the highest score, L is the lowest score.

Objective function: The weighted minimum mean square error model is used in this paper to establish the objective function. The error between the expected value and the real value of types, knowledge topics, difficulty and degree of differentiation is calculated as the following:

$$Q_k(j) = \sum_{i=1}^{m} a_{i1}\delta_k(i, j)$$

where, Q_k (j) is the score of jth k attribute (k = 1, 2, 3, 4 represent the type, the knowledge topic, the difficulty and the degree of differentiation).

$$\delta_k(i, j) = \begin{cases} 1 & a_{ik} = j \\ 0 & a_{ik} \neq j \end{cases} \quad k = 1, 2, 3, 4, \; j = 1, 2, \ldots t$$

so:

$$e_k = \sum_{j=1}^{t}((L_k(j) - Q_k(j)) / L_k(j)) \quad (4)$$

where, L^j_k is the objective score of j-th k attribute.

The error of the answer time and the total score is calculated as follows:

$$e_k = (L_k - \sum_{i=1}^{m} a_{ik}) \Big/ L_k \quad (5)$$

In the equation, k = 5, 6 represent the answer time and the total score. L_k represents the target score of the corresponding attribute.

The exposure degree is calculated by the following equation:

$$e_k = (L_k - 1 \Big/ (\sum_{i=1}^{m} a_{ik} + 1)) \Big/ L_k, \; k = 7 \quad (6)$$

The objectivefunction is W-MMSE:

$$\min f = \sum_{i=1}^{7} \omega_i e_i^2 \quad (7)$$

where, ω_i is the weight of each indicator:

$$\sum_{i=1}^{7} \omega_i = 1$$

Sampling strategy design

Step 1: Users determine pumping problem constraints
Step 2: Question module selects samples in random according to the value of the types, chapters, knowledge topics in the constraints to generate the initial population--- chromosome (paper). The gene of each chromosome is expressed by question number
Step 3: Calculate and inspect the fitness degree of each gene in chromosome; quit pumping if the error of the result and the objective value is in the allowable range or reaching the maximum number of iterations
Step 4: Choose 2/3 chromosome of the high fitness degree as the male parent individuals in the population; then cross by cross over probability; mutate by mutation probability and generate new chromosome individuals
Step 5: Compare the fitness degree between new individual and the male parent individual and keep the better one
Step 6: Return to step 3 to the next generation evolution.

Constraints and process of generating

Setting generating constraints: Interact with the operator; complete the generating constraints setting of the course (test item). The program flow chart is shown in Fig. 2.

Pumping generating paper: Complete the pumping of the course (test item) according to the pumping constraints. The program flow chart is shown in Fig. 3.

Hierarchic-adapted genetic algorithm: GA intelligent-generating test paper operation is composed of chromosome coding, initial population, fitness function, genetic operator and controls parameter.

Chromosome coding: Chromosome coding converts the solution space of the problem into searching space which GA could deal with. Setting the number of questions in

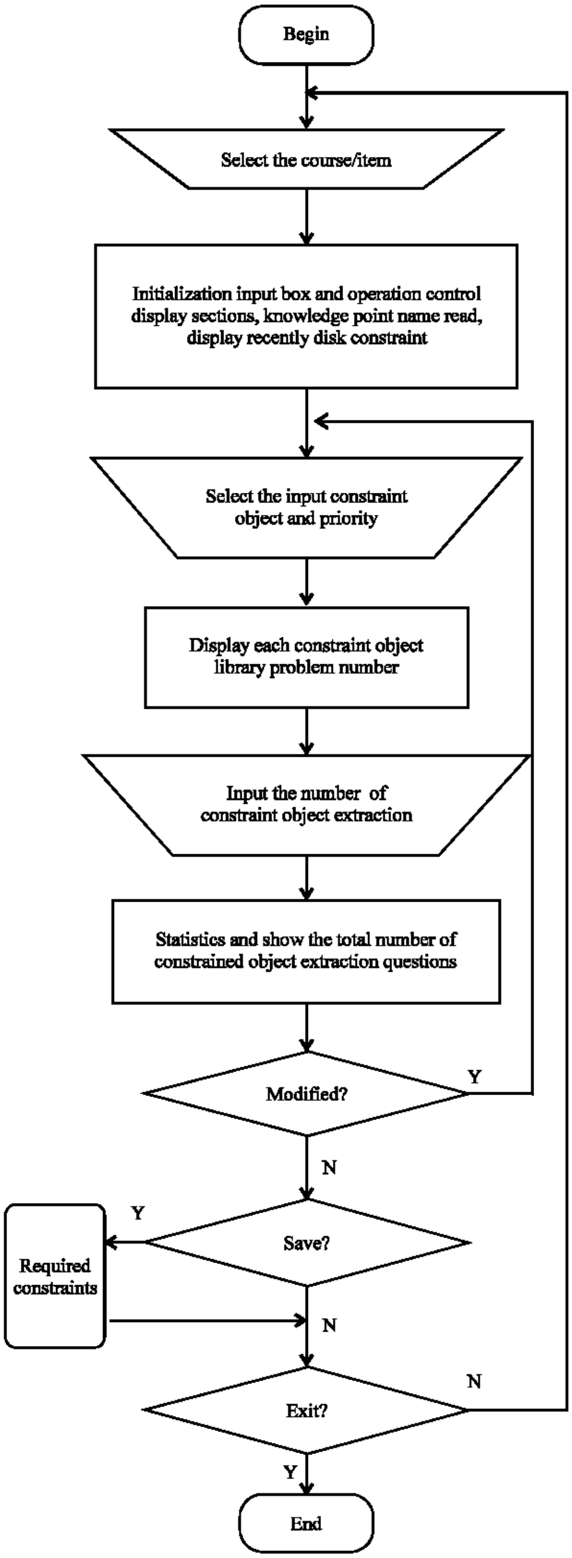

Fig. 2: Flow chart of generating constraints

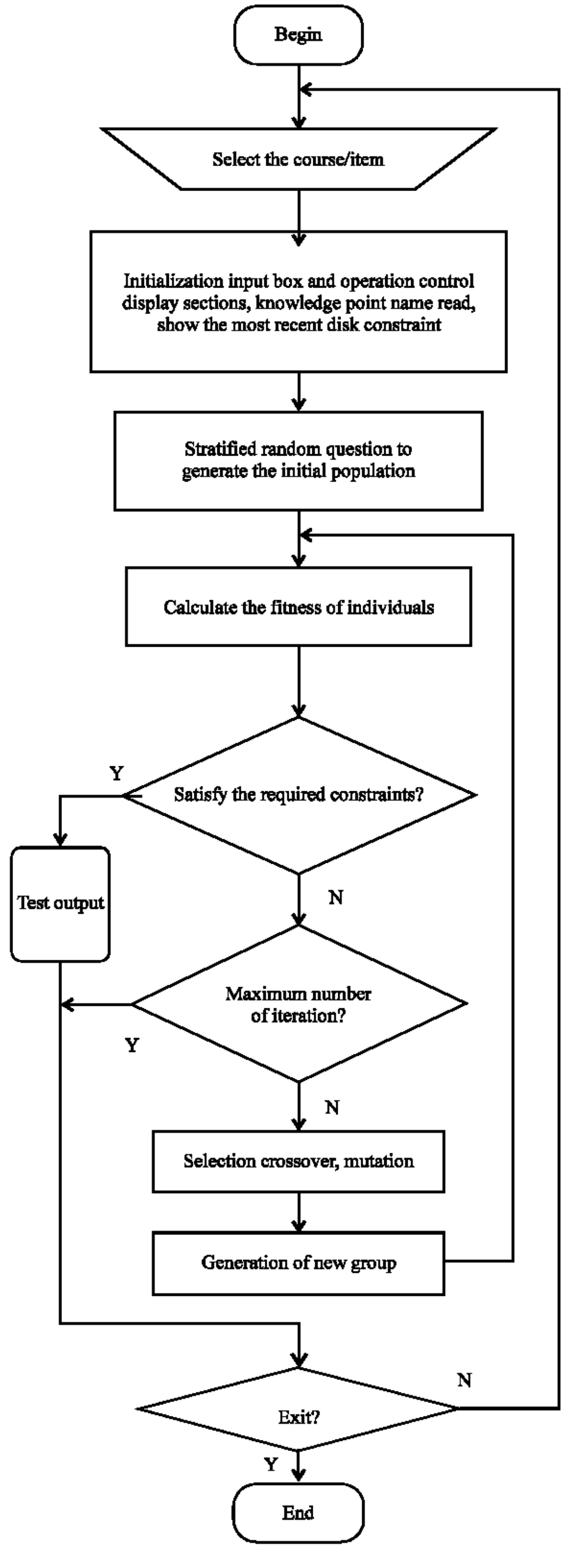

Fig. 3: Flow diagram of pumping test paper

the question bank is L which coded in a binary string; the form is x_1, x_2 x_3...x_L:

$$x_i = \begin{cases} 1 & \text{if number i is selected} \\ 0 & \text{else} \end{cases} \quad (8)$$

Initial population generation: The initial population iscomposedof a number of chromosomes in a scale or the set of solutions. In this article, the initial population consist of different constrained chromosomes which are generated randomly.

Fitness function: The fitness function is the gist for the optimal solution based on genetic algorithm. The higher the fitness degree of an individual is, the higher the chosen probability is. The fitness function can be defined as the following:

$$F = 1/(1+f) = 1\Big/(1+\sum_{i=1}^{7}\omega_i e_i^2) \tag{9}$$

Genetic operator: The genetic operator includes selection, cross over and mutation. In the searching process, it firstly calculate fitness degree, then choose corresponding individual to cross and mutate according to the choice mechanism, thus it generates new individuals and keep the variety of individuals.

Select: This algorithm combines the best individual conservation method and the roulette to realize the selection operation; calculate fitness value of each individual in the population. The highest degree of the individual is directly copied to the next generation of adaptation in each generation. For the rest of the M-1 individuals, roulette wheel selection method is used. The best individuals are directly genetic to the next generation which can guarantee the emergence of the optimal individual in the next generation and improve the local search ability and the rate of convergence.

The roulette wheel selection probability equation is as follow:

$$P_s = F_i\Big/\sum F_i \tag{10}$$

where, F_i represents the I individual fitness value.

Crossover: It is in favor of small genetic preservation of population to use a large cross probability and mutation probability. However, it will cause the difference of individual fitness among populations getting smaller. The small crossover probability and the big mutation probability can increase the diversity of individuals populations, enhance the global search ability of the algorithm and avoid the premature phenomenon of the algorithm. This study adopts the self-adaptive crossover mutation probability. The calculation equation is as follows:

$$P_c = \begin{cases} P_{c1} - (P_{c1} - P_{c2})(F_c - \overline{F})/(F_{max} - \overline{F}) & \text{if } F_c \geq \overline{F} \\ P_{c1} & \text{else} \end{cases} \tag{11}$$

$P_{c1} = 0.9$, $P_{c2} = 0.6$, $P_{m1} = 0.01$, $P_{m2} = 0.001$, F_{max} indicates the maximum fitness value in the group; $\overline{F}$ represents the F_c average colony adaptation degree of each generation; is the larger fitness value of the two individuals in the crossover.

The cross probability was calculated by using the cross formula. If P_c> random (0, 1), crossover operation is performed at the crossing position to get two new individuals, otherwise the cross operation is not performed.

Mutation: mutation equation is as follows:

$$P_m = \begin{cases} P_{m1} - (P_{m1} - P_{m2})(F_{max} - F_m)/(F_{max} - \overline{F}) & \text{if } F_m \geq \overline{F} \\ P_{m1} & \text{else} \end{cases} \tag{12}$$

where, F_m is the fitness value of the variation individual.

Mutation probability can be calculated through the variation equation. If P_m>random (0,1), variability will generate in individual variation. It will increase chromosome types in a certain extent which will maintain the diversity of species and avoid the premature. In addition, the variation only change the value of a position in the chromosome which can avoid convergence problems arising due to generating too many new chromosomes.

Genetic arithmetic termination condition: The test paper generating algorithm adopts the following three termination conditions:

- Population meets fitness requirements
- Genetic evolution reaches to a specified maximum number of iterations
- Best individual in the group has not improved in continuous several generations

Adaptive test paper algorithm based on hierarchical genetic algorithm: In the proposed algorithm, firstly, the M populations operate on adaptive genetic algorithm and store the intermediate results, then operate the top adaptive genetic algorithm (mainly the selection, crossover and mutation operations). The calculation keeps circulating until the paper which meets the constraints is found (Huang, 2011).

The intelligent test paper based on hierarchical and self-adapting genetic algorithm is as following:

Step 1: Setting test constraints and control parameters (the constraint conditions: types, quantity, difficulty, discrimination, knowledge topics and the control parameters: the maximum number of iterations, the size of the initial population initialization and the crossover and mutation probability)

Step 2: Segment coding

Step 3: Determining the objective function and the fitness function

Step 4: Randomly generating the initial population according to the constraints and dividing them into M sub populations in which each sub population has the same number of paper individuals

Step 5: Encoding each individual sample in the sub populations and calculating the fitness value of each individual according to the fitness function with the first I test index and the error value between the weight of the individual sample index and the expected index

Step 6: Sub populations G(i) got from the initial M operate respectively self-adaptive genetic operation for a certain number of times. If there is a test paper meets the performance requirements, the results of the test paper output; the paper will exit, otherwise it goes to the fourth step

Step 7: Saving the above M genetic algorithm results to an array of G(M, m) and calculating the average fitness values of M populations; then recording the results in the array A(M)

Step 8: Selecting in the G(M, m) in M sub populations, the highest average fitness sub populations will be directly copied to the next generation. As for the remaining M -l sub populations, the biased Roulette Wheel Selection (RWS), method is adopted to select roulette wheel selection probability equation:

$$F_i/\sum F_i$$

For each population individual inside, a similar choice method is also used; then the individuals crossover according to the crossover probability

Step 9: Mutation operation is done to G(M, m) in accordance with the mutation probability

Step 10: The newly produced sub populations again operate the self-adaptive genetic operation respectively for a certain number (such as the 10 generation, the 20 generation)

Step 11: If there is a individual sample which meets the quality requirements, the test paper output and exit the test paper generating system; otherwise it executes step 7. The operation goes repeatedly, until the satisfied test sample is found

RESULTS

Encoding operation: Ranking in layer and numbering the questions according to the constraint pumping level settled by users (for example, the first layer is the type, the second layer is the section or the knowledge topic, the third layer is the difficulty or the discrimination). The question number, as the genetic material of the chromosome, is unified by 4 decimal digit string Table 1.

Question extraction operation: Pumping test randomly within constraints according to the user specified layer rank of count constraint hierarchy (such as types, chapters, knowledge). The selected question numbers constitute chromosomal genes Table 1.

Genetic operation: In order to meet the user specified count constraints, genetic operations such as the selection, crossover, mutation (Table 2 and 3) and other genetic operations are operated in the same counting

Table 1: Sketch map of chromosome gene coding

Multiple choice					Blank filling					Calculation questions					Design				
1009	1012	1022	1024	1025	2016	2017	2021	2022	2028	2049	2060	2066	2067	2070	2095	2105	3003	3005	3008

Table 2: Sketch map of chromosome genetic crossover operation

Multiple choice					Blank filling					Calculation questions					Design				
(a) Before cross operation																			
Chromosome A																			
1009	1012	1022	1024	1025	2016	2017	2021	2022	2028	2049	2060	2066	2067	2070	2095	2105	3003	3005	3008
Chromosome B																			
1028	1046	2002	2004	2014	2029	2031	2035	2038	2040	2073	2076	2077	2081	2089	3011	3027	3028	3054	3011
(b) After cross operation																			
Chromosome A																			
1009	1012	1022	1024	1025	2016	2017	2021	2038	2040	2049	2060	2066	2067	2070	2095	2105	3003	3005	3008
Chromosome B																			
1028	1046	2002	2004	2014	2029	2031	2035	2022	2028	2073	2076	2077	2081	2089	3011	3027	3028	3054	3011

Table 3: Sketch map of chromosome gene mutation

Multiple choice					Blank filling					Calculation questions					Design				
(a) Before the mutation																			
Chromosome A																			
1009	1012	1022	1024	1025	2016	2017	2021	2022	2028	2049	2060	2066	2067	2070	2095	2105	3003	3005	3008
(b) After the mutation																			
Chromosome A																			
1009	1012	1022	1024	1025	2016	2017	2021	2030	2028	2049	2060	2066	2067	2070	2095	2105	3003	3005	3008

Table 4: Difficulty value of the item bank

Level of difficulty	Easy	Easier	Medium	Harder	Hard
Coefficient of difficulty	0, 0.2	0.2, 0.4	0.4, 0.6	0.6, 0.8	0.8, 1

Table 5: Differentiation degree of the item bank

Differentiation degree	High	Higher	Medium	Poorer	Poor
Coefficient of differentiation degree	0.01-0.19	0.20-0.39	0.40-0.59	0.60-0.79	0.80-0.99

constraint hierarchies in random. For example, when the user specifies the types of constraints, the crossover and mutation are proceeded in the same gene questions randomly.

Recode elimination: The adoption of gene encoding may cause chromosome gene duplication in the genetic operation. The duplicated mutation to the duplicated gene can eliminate the duplication.

Control parameters: The probability of crossover directly determines the speed of generation of new chromosomes. However, if the crossover probability is too big may undermine the excellent characteristics of the population. It is set as 0.8. Mutation probability also affects the generation speed of the individual. It will reduce the performance of the algorithm if the mutation probability is too big, thus it is 0.1. The maximum number of iterations is 200 generation; the average difficulty, discrimination and the allowable completion time error service is ±0.05.

Transformation of difficulty: “Easer, easy, medium, hard, difficult” 5 levels of difficulty are mapped to Table 4, in degree of difficulty involved in calculation of average difficulty.

Distinction of conversion: “Higher, high, moderate, poor, poorer” 5 grades respectively mapped to Table 5 and join the average discrimination calculation.

DISCUSSION

Experiment comparison: The comparison of the standard genetic algorithm with the hierarchical and self-adapting genetic algorithm (HSAGA) (Huang, 2011) is shown in Fig. 4.

In the experiment, we test the intelligent test paper algorithm by running the algorithm 200 times with the standard genetic algorithm and the HSAGA. The experimental results are shown in Fig. 4. The blue curve indicates the standard genetic algorithm and the red curve is the HSAGA. From Fig. 4, it can be seen that the standard genetic algorithm keeps the better solutions than the HSAGA in the first 60 generations. However, with the increase of the number of iterations, in the 90 generation or so, the HSAGA becomes gradually stabilized and the advantage of HSAGA is shown. The solution is obvious better than standard genetic algorithm.

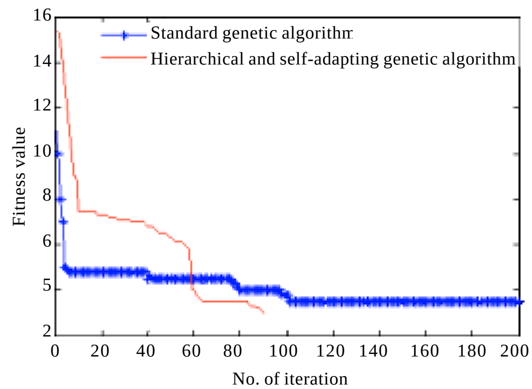

Fig. 4: Comparison of standard genetic algorithm with HSAGA

CONCLUSION

Intelligent-generating test paper is to automatically select the test from the database by computer according to the requirements of the teacher and teaching to form a test paper which can meet the requirements of the knowledge distribution, the type distribution, the cognitive level distribution, the difficulty distribution, the distinction degree distribution, the time distribution and the fraction distribution. The prematurity and slow convergence problem are easy to appear in global optimization in the traditional genetic algorithms. This study adopts the packet coding and establish objective function with the weighted minimum mean square error model which can avoid the individual decoding. The calculation speed and the quality of generating paper are improved. The robustness is good. The generated test papers can satisfy the actual needs of users. With the combination of the generating test paper algorithm and the test database, this system, generates papers by extracting questions satisfy the constraints. It realized the separation of teaching and examination and liberated the teachers from the tedious examinations. At the same time, this system can also ensure the fairness of the exam and reflect the true ability level of student.

ACKNOWLEDGMENTS

This project is supported by the National Natural Science Foundation of China (No. 61202496), Hunan Provincial Natural Science Foundation of China (No. 13JJ2031), the Science and Technology Program of Hunan Province (No. 2014SK2025, 2012GK3065). The "12.5" education planning project of Hunan province (No. XJK013CXX014), Hunan province and decision-making consulting research (No. 2013BZZ54) and Science and Technology Program of Yiyang city (No. 2011JZ15).

REFERENCES

Dorronsoro, B. and E. Alba, 2006. A simple cellular genetic algorithm for continuous optimization. Proceedings of the IEEE Congress on Evolutionary Computation, July 17-24, 2006, Vancouver, BC., pp: 2838-2844.

Galante, M., 1996. Genetic algorithms as an approach to optimize real-world trusses. Int. J. Numer. Methods Eng., 39: 361-382.

Golino, G., 2005. Improved genetic algorithm for the design of the optimal antenna division in sub-arrays: A multi-objective genetic algorithm. Proceedings of the IEEE International Radar Conference, May 9-12, 2005, Italy, pp: 629-634.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. 1st Edn., University of Michigan Press, Ann Arbor, Michigan, ISBN: 9780472084609, Pages: 183.

Huang, B.L., 2011. Application of adaptive genetic algorithm in intelligent test paper composition. Comput. Eng., 37: 161-163.

Jerald, J., P. Asokan, R. Saravanan and A.D.C. Rani, 2006. Simultaneous scheduling of parts and automated guided vehicles in an FMS environment using adaptive genetic algorithm. Int. J. Adv. Manuf. Technol., 29: 584-589.

Leung, Y.W. and Y. Wang, 2000. Multiobjective programming using uniform design and genetic algorithm. IEEE Trans. Syst. Man Cybern. Part C: Applic. Rev., 30: 293-304.

Martinez-Gomez, G.J., J.A. Gamez and V.I. Garcia-Varea, 2010. Comparing Cellular and Panmictic Genetic Algorithms for Real-Time Object Detection. In: Applications of Evolutionary Computation, Di Chio, C., A. Brabazon, G. di Caro, M. Ebner and M. Farooq *et al*. (Eds.). Springer Science and Business Media, New York, USA., ISBN-13: 9783642122392, pp: 261-271.

Nebro, A.J., J.J. Durillo, F. Luna, B. Dorronsoro and E. Alba, 2009. MOCell: A cellular genetic algorithm for multiobjective optimization. Int. J. Intell. Syst., 7: 726-746.

Ombuki, B., B.J. Ross and F. Hanshar, 2006. Multi-objective genetic algorithms for vehicle routing problem with time windows. Applied Intell., 24: 17-30.

Zitzler, E. and L. Thiele, 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. IEEE Trans. Evol. Comput., 3: 257-271.