Personalized Question Bank Research Based on Particle Swarm Optimization

Zhiyun Chen, Yong Wang and Yue Bai*
School of Data Science & Engineering, East China Normal University, Shanghai, China 200062
*Corresponding author

***Abstract*—Traditional question banks only pay attention to digitalize questions. This article proposes a new algorithm based on particle swarm optimization improved by personal parameters, which combining analyzing user's exercise data. This method can provide an online, on-the-fly and personal service for each student. Experiments show that in response to varied numbers of questions, it takes a shorter time and is able to combine user's understating of themselves knowledge with the right of selecting chapters. This method has the ability to lift students' experience when using question bank.**

Keywords—particle swarm optimization; personalized learning; question bank

I. INTRODUCTION

In the contemporary era of information technology, the question bank system is widely used in teaching.

The personalized question bank system can automatically customize a high-quality, personalized exercise that meets the student's actual level through the judgment of the student's level and test information [1].

How to extract test questions from a large number of test questions that meet the test requirements and ensure the quality of the extraction and the success rate at the same time is one of the focuses of the research. Wang Yuying et al [2] adopted the strategy of random extraction and approximate matching, which improved the performance of the extraction to some extent. In order to avoid the blind randomness of random extraction and reduce the computational complexity of backtracking heuristics, genetic algorithm and particle swarm optimization algorithm are applied to generate the test paper. Du Ming et al [3] proposed an intelligent genetic algorithm based on the knowledge of the students using the binary coded genetic algorithm.

However, only the correct answer of the students is used whether the students have done the question and information such as the distribution of knowledge were ignored. Therefore, this paper proposes an improved solution based on particle swarm optimization algorithm[3], which provide students with real-time, online exercises. Students was able to select the distribution of the concept of knowledge, changing the number and difficulty of questions. The student's personalized matrix and test information matrix proposed in this paper have the ability of calculating the recommendation degree of the test questions that meet the students' requirements, ranking the questions. With the combination of students' understanding of themselves and system's personalized recommendation, the generation of exercise become more intelligent and is able to meet the students' personalized requirements.

II. THE ESTABLISHMENT OF A EXERCISE MODEL

When the question bank is established, it should include indicators such as difficulty, discrimination, and chapter information.

After extracting the test questions from the test bank that meet the requirements of the current students, students were provided with exercise. If the total number of questions in the question bank is k, the generated exercise can be defined as a matrix A. Each line represents a test question, and each column is the attribute of the test question. The number of questions to be extracted is m, and each question has n columns of attributes. The attributes of the n columns can be difficulty, chapters, etc., as shown in Figure I.

$$\begin{bmatrix} Q1 & diff1 & \cdots & chap1 & \cdots & t1 \\ Q2 & diff2 & \cdots & chap2 & \cdots & t2 \\ \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ Qm & diffm & \cdots & chapm & \cdots & tm \end{bmatrix}$$

FIGURE I. QUESTION BANK MATRIX

In addition to using all student data to determine the difficulty of the test questions, personalization also needs to use the records of questions the student has done. The personalized question-extracting model in this paper is shown in Figure II, which is designed from two aspects: test information and student personalized information.

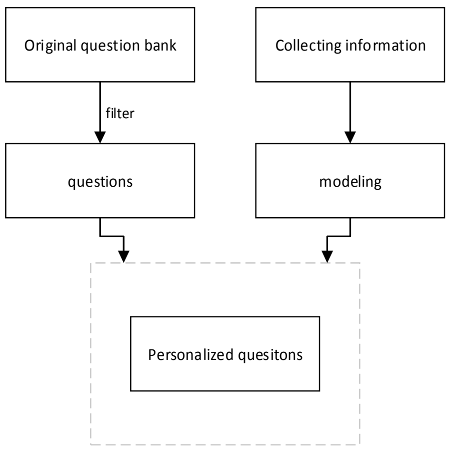

FIGURE II. PERSONALIZED EXTRACTING MODEL

A. Student Personalized Information

There are three dimension for each question using for personalized information, “done”, “done correctly”, “ mastered”. As shows in Table I , 1 means that the question have been done or made a mistake, 0 means that have not been done or made a mistake. If the current student does not want to practice one test question, he can mark the question as being mastered and representing using number 1. In this way, each student has a personalization matrix, as shown in figure. III: the test questions in the test bank appear in order of chapters and the type of questions (choose, fill in the blanks). The id number of the questions starts from 1. The student personalization matrix is designed as the number of columns in the question bank. Generate a matrix of this for each student.

TABLE I. STUDENT PERSONAL INFORMATION TABLE

num / column	1	2	3	4	5	6	7	8	9……
Done	1	0	1	0	1	1	0	0	0……
Correct	0	0	1	0	0	1	0	0	0……
Mastered	0	0	0	1	1	0	0	0	0……

$$\begin{bmatrix} 0 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 0 \end{bmatrix}$$

FIGURE III. STUDENT PERSONALIZED MATRIX

B. Question Information

As shown in Table II, it corresponds to the behavior record of all students in all the questions in the library, which is called test information table. The data in Table II is normalized to interval 0~1 forming the test information matrix.

TABLE II. QUESTION INFORMATION TABLE

number / project	1	2	3	4	5	6	7	8	9……
starred	10	7	100	50	7	87	33	12	63……
incorrect	55	32	11	67	44	28	88	43	73……

$$\begin{bmatrix} 0.3 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 0.55 & \cdots & 0.47 \end{bmatrix}$$

FIGURE IV. QUESTION INFORMATION MATRIX

Next, we should use the student personalization matrix and test question information matrix mentioned above to calculate the recommendation degree of each test question to the current students, which is represented by γ. The larger the γ is, the higher the recommendation level is. For the current students, the test recommendation level of each test question is calculated by formula (1).

$$\gamma = \omega_1\left(-done + error - grasp\right) + \omega_2\left(star + di\right) \quad (1)$$

Where ω_1 represents the weight of the current student's personalized information, and ω_2 represents the test information’s (in this study, ω1, ω2 are set to 0.5 by default, which can be adjusted by the students according to the preference of the personalized information and the test information). Done, error and grasp can be draw from the student's personalized matrix. Star indicates the number of times the test questions in the test matrix are starred by all students, and di indicates the number of times the test questions are incorrected done by all students.

III. Personalized Particle Swarm Optimization Algorithm with Improved Input Parameters

Based on the particle swarm optimization algorithm proposed by Bui1 T et al. [4], the principle of particle swarm optimization algorithm, this paper proposes using student personalization matrix and test information matrix to generate a test for each student. A personalized particle swarm extraction was proposed to improve input parameters, and Figure V shows a flow chart of the improved algorithm.

$$f = \sum_{n=1}^{m}\frac{c_n - c_a}{\varepsilon} + \frac{\sum_{i=1}^{\varepsilon} d_i}{\varepsilon} - d_a \quad (2)$$

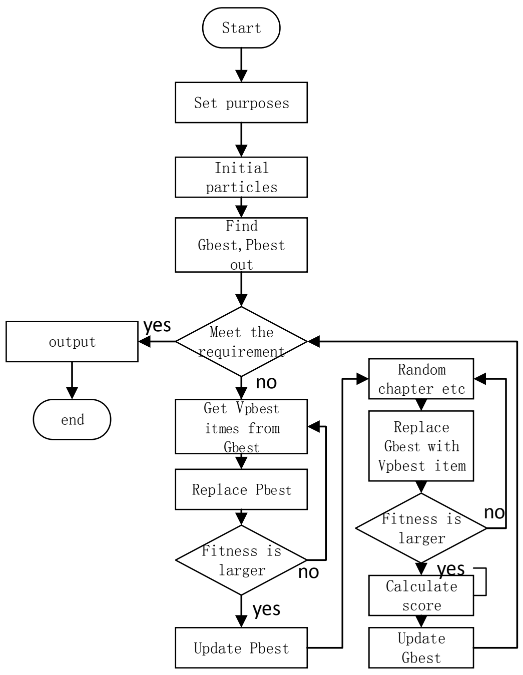

FIGURE V. PERSONALIZED PARTICLE SWARM ALGORITHM FLOW CHART

If the current student wants to get the ε test questions to practice, in order to avoid the problem of binary code length , each set of exercises (corresponds to a particle) is coded as {a1, a_2, a_3,...a_i,..., $a_ε$}, where ai represents the test question number in the exercise question, and the real number code occupies less storage space and the calculation amount is smaller.

The particle swarm algorithm improved by input parameters designed in this paper adopts the following steps for individualized exercises.

A. Students Set Practice Goals

Each student can set the number of questions ε, the overall difficulty (1-5, corresponding to the easiest to the most difficult) and the chapter distribution according to their own knowledge level. These parameters will be used for the following fitness function calculation.

B. Initialize the Particle Swarm

According to the number of questions ε and the size of the population (the default setting is 20, corresponding to the set of all exercises obtained in one iteration), ε questions are extracted in the question bank randomly. The chapter knowledge and difficulty information of the randomly extracted ε test questions, the student's personalized matrix and test information matrix are used to replace the randomly selected test questions with the highly recommended test questions.

C. Calculate the Fitness Value of the Particle According to the Fitness Function

The algorithm calculates the fitness of the particles (the fitness refers to the degree to which the extracted exercises meet the student's practice goals), for the current students, the fitness function f is defined as a formula. 2.

ε is the total number of exercises the current student sets, m is the total number of chapters in the question bank, c_n is the number of questions in the corresponding chapter, and c_a is the number of corresponding chapters the current student sets, di is the number of the respective questions difficulty, da is the difficulty required for the practice goal, and the difficulty value range is 1-5.

By calculating the fitness value of the particle, we find out the initial population global optimal G_{best}, and other local optimal P_{best}, where G_{best} refers to the practice problem that best meets the student's requirements in all iterations. P_{best} refers to all other exercises that meet the requirements of the practice target in a certain iteration.

D. Update the Position of the Particles

Let P_{best} approach G_{best} at speed V_{pbest} and get the tests in $G_{best.}$ V_{pbest}=α*ε(V_{pbest} is the number of questions, α is a coefficient, in order to get G_{best} less than or equal to G_{best} questions, so α is at (0,1]), and V_{pbest} questions are randomly selected in P_{best} as Replace the target, replace it with the V_{pbest} test in G_{best}. If the adapted P_{best} has greater fitness than the original P_{best}, replace the original P_{best} with the updated P_{best}, otherwise keep the original P_{best}.

Make G_{best} approach the target with speed V_{gbest}, V_{gbest}=β*ε (V_{gbest} is the number of questions, β is a coefficient, at (0,1)), randomly select V_{gbest} questions in G_{best} as the replacement target, use the question bank Replace the V_{gbest} questions in the test, and generate personalized questions. The specific steps are as follows:

a) Because there are test questions in the question bank belong to the same chapter and difficulty, if you do not distinguish, random update position will make the running time longer, resulting in poor performance of the algorithm. Therefore, use the method of saving all the questions into a dictionary, for example:

{Chapter 3: {Difficulty 1: 4, 7, 3; Difficulty 2:12, 2, 5, 6...}...}

For the random chapter c (the experimental chapter range 1-6) and the difficulty d (the experimental difficulty 1-5), if there is a difficulty d test in this chapter c, then the corresponding test list is obtained. If it does not exist, loop again until the test list meets the requirements.

b) Read the records of all students before each question, and get the student's personalized matrix.

Considering the number of all students, the database is read once every day. The combination of the current student's personalized information (stored in the student's personalized matrix) and the question database information matrix, using the formula (1), can calculate the score of the recommendation level of all the questions in the test list for the current student, and The recommendation score is sorted from high to low.

Since some questions in G_{best} may be the same as those in the list of questions obtained in step 1, in order to avoid generating the same questions, the algorithm needs to perform the difference between the questions in the test list and the questions in G_{best}, so that it does not appear in G_{best}. After sorting the list of questions, select the V_{gbest} questions with high recommendation γ and join G_{best}.

E. Termination of the Algorithm

Get the best practice questions for the current student. When the fitness function value is 0.01, it can be considered to meet the student's target requirements.

The maximum number of iterations is reached. To avoid wasting time, set a maximum number of iterations of 1000 to prevent students from waiting too long.

If one of the above conditions is met, the algorithm terminates.

IV. Three Personalized Practice Experiments and Analysis

The content of the question bank used in this article is the university computer practice test library, which is implemented by python + MySQL . The experimental environment is CPU: core i5, memory 8G.

A. Question Information of the Question Bank

There are 1029 questions in the question bank, including 677 multiple-choice questions, 352 fill-in-the-blank questions, and the test questions are divided into six chapters. Table III shows the distribution of the test chapters.

TABLE III. THE DISTRIBUTION OF ITEMS

chapter	1	2	3	4	5	6	all
all	249	154	130	234	148	114	1029
choice	168	107	79	157	97	69	677
blanks	81	47	51	77	51	45	352

B. Experiment Analysis

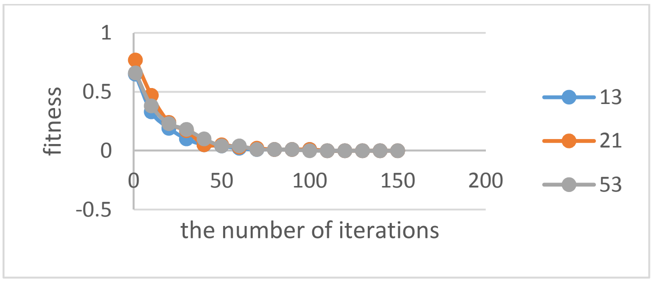

FIGURE VI. ITERATIONS AND FITNESS

Figure VI shows the fitness function when the number of questions required is 36 and the difficulty requirement is 2.5. The three students who randomly numbered 13, 21, and 53 take 20 exercises out of the exercise. When the number of iterations changes, it can be seen that the fitness function value decreases rapidly with the number of iterations. Within 70 iterations, the fitness is close to 0, indicating that the algorithm can obtain exercises that meet the requirements of the student's practice goals with fewer iterations.

As shown in figures VII and VIII, when the difficulty requirement is 2.5, the knowledge points of each chapter are evenly distributed, and the value of the fitness function is 0.1 or less, 20 experiments are performed on the students numbered 13, 21, 53, and the average time used by the algorithm is adapted. The degree of change with the size of the population (the number of practice questions required). The algorithm is time-consuming and stable at around 0.54s.

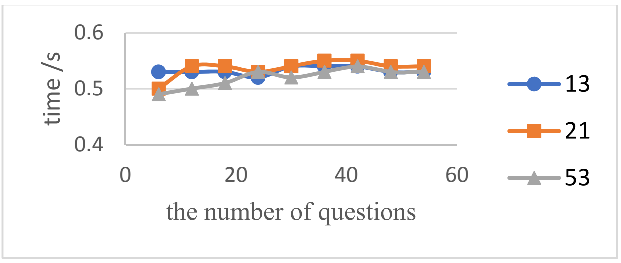

FIGURE VII. TEST NUMBER AND AVERAGE TIME CHANGE TABLE

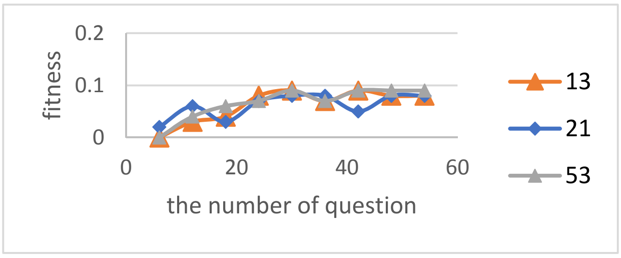

FIGURE VIII. PRACTICE TEST NUMBER AND FITNESS VALUE CHANGE TABLE

As shown in Table IV, the algorithm adds random units and difficulty when updating the particle position, and reduces the time spent on the algorithm operation by storing the dictionary.

TABLE IV. TIME COMPARISON TABLE

method	add	don't add
time（s）	0.53	1.14

As shown in figure. IX, it is a comparison result with the running time of the personalized genetic algorithm proposed in Document 7. For the same student, the goal is set to 60 questions, the difficulty is 3, the crossover probability of the genetic algorithm is 0.93, the mutation probability is 0.07, and the population size is 20, the same exercise effect as the algorithm is produced (the fitness value is the same) When adding the personalization scheme designed in this paper, the figure is a comparison of 9 runs. It can be seen that the algorithm takes a short time, and after running 3 times, the time used by the algorithm is basically stable at about 0.6s.

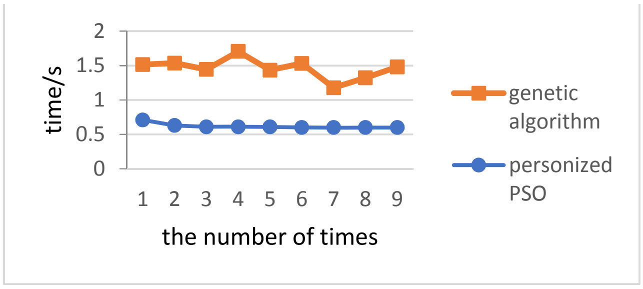

FIGURE IX. THE TIME COMPARISON BETWEEN PERSONIZED PSO AND GA

V. CONCLUSION

Based on the research of particle swarm optimization algorithm, according to the students' needs, the student personalized matrix and the test information matrix are used as the input parameters of the particle group. Through the calculation and sorting of the recommendation degree, students can generate personalized practice questions, so that the question bank has automatic recommendation. The function of personalized exercises. Experiments show that compared with the binary-encoded genetic algorithm mentioned in the literature 2, the algorithm has short running time, high efficiency, fewer parameters, and simpler implementation. It can meet the needs of students to extract practice questions online and improve the use of the test bank.

REFERENCES

[1] Li Junjie, Zhang Jianfei, Hu Jie, Sheng Shouzhuo. Design and application of intelligent personalized language learning platform based on adaptive question bank[J]. Modern Educational Technology, 2018, 28(10): 5-11.

[2] Wang Yuying, Hou Shuang, Guo Maozu. Research on Automatic Generation Algorithm of Test Paper System Test Papers[J].Journal of Harbin Institute of Technology,2003(03):342-346.

[3] DU Ming,WANG Shumei,HAO Guosheng.Design of an Improved Intelligent Genetic Algorithm Based on Knowledge Level[J].Control Engineering,2017,24(10):2112-2117.

[4] Shi, Y.; Eberhart, R.C. (1998). "A modified particle swarm optimizer". Proceedings of IEEE International Conference on Evolutionary Computation. pp. 69–73.

[5] Bui1 T, Nguyen T, Vo B, Nguyen T, Pedrycz W AND Snasel V. (2018). Application of particle swarm optimization to create multiple-choice tests [J]. Journal of Information Science and Engineering, Vol. 34 No. 6, pp. 1405-1423