

# An Improved Personalized Genetic Algorithm Incorporated Item Distribution for Test Sheet Assembling

Peipei Gu<sup>1</sup>, Zhendong Niu<sup>1,2,\*</sup>, Wei Chen<sup>3</sup>, Xuting Chen<sup>1</sup>, Ke Niu<sup>1</sup> and Jia Sun<sup>1</sup>

<sup>1</sup> School of Computer Science and Technology, Beijing Institute of Technology, 100081 Beijing, China

<sup>2</sup> Information School, University of Pittsburgh, 15260 Pennsylvania, USA

<sup>3</sup> Agricultural Information Institute, Chinese Academy of Agricultural Sciences, 100081 Beijing, China

Received: 22 Jul. 2013, Revised: 24 Oct. 2013, Accepted: 25 Oct. 2013

Published online: 1 Jul. 2014

**Abstract:** In recent years, computer-based testing has become an effective approach to evaluate students' learning level. In our previous work, a personalized genetic algorithm (PGA) for test sheet assembling was proposed. In this paper, an improved personalized genetic algorithm named PGAC which makes an improvement in the crossover process of PGA is presented. Considering item distribution, an improved algorithm incorporated item distribution (IGAID) based on PGAC is presented to assemble simulation test sheets which have good item distribution in knowledge hierarchy for each student. Experiments and comparison with random assembling algorithm and GA are conducted. The results show that PGAC supports effectively assembling a test sheet with more non-mastered items for different students, and IGAID is capable of effectively constructing simulation test sheets with good item distribution in knowledge hierarchy for each student.

**Keywords:** computer-aided testing, genetic algorithm, test-sheet composition problem, test assembling

## 1 Introduction

With the rapid development of information technology, computer-based testing is adopted by many colleges and universities throughout the world [1]. Computer-based testing is an important part in E-Learning systems and intelligent computer-aided instruction systems [2]. Students' learning level can be evaluated by using computer-based testing.

The algorithms which selecting items to construct final test sheet play an important role in the high quality testing [3,4]. A lot of experts and researchers concentrate on proposing test sheet generation algorithms with high efficiency. Some test assembling systems use random or manual strategy [5,6]. Some experts use methods in machine learning to solve the test sheet generation problem: Tabu search [7], genetic algorithm (GA) [8], particle swarm optimization [9,10], etc. Hwang *et al.* adopt clustering, dynamic programming, Tabu search and heuristic algorithm to improve the construction procedure of the test sheet in high quality with specified constraints

[6,7]. Yin *et al.* incorporate the particle swarm optimization (PSO) idea into the test sheet generation procedure to facilitate the efficiency of assembling near optimal serial test sheets from large item bank [9]. Lee *et al.* propose an Immune algorithm to improve the efficiency of generating near optimal test sheet [11]. Wang *et al.* adopt discrete coding strategy in a discrete differential evolution algorithm to assemble appropriate test sheets [12]. Ou-Yang *et al.* propose an approach to generate test paper based on students' learning situation [13]. Moreover, there is Item Response Theory (IRT) which is different from Classical Test Theory (CTT) that all above algorithms are based on. IRT establishes some non-linear models between subjects' ability and item parameters [14].

Before a student takes regular examinations, he/she would like to practice simulation tests which are generated according to his/her learning level and preference information, with the same difficulty and discrimination degree as regular examinations. In the final simulation test sheets, comparing with the mastered

\* Corresponding author e-mail: [zniu@bit.edu.cn](mailto:zniu@bit.edu.cn)

items, students would prefer non-mastered ones occupy a bigger proportion in these simulation tests. These personalized simulated tests can help the student to explore his/her inner strengths and guide him/her to learn more knowledge which he/she hasn't mastered. However the above approaches don't support students' personalized simulated tests.

Moreover, when a student prepares to attend a test which is assembled by above approaches that are based on CTT, he/she might find there are some test items corresponding with the same concept, section or chapter no matter whether he/she mastered this knowledge in the final test sheet or not. In a normal test, the test sheet should contain items which have a good distribution in concept, section and chapter. Excess test items with the same concept, section or chapter would play bad influence on evaluating students' real ability of mastering knowledge. Moreover, it would help little in mastering more knowledge for each student.

In our previous work [15], a model which formulates the personalized test generation problem was presented. Based on the model, a personalized genetic algorithm (PGA) which assembles appropriate test sheet for each student, according to their different learning level in concepts and preference information of items was proposed.

In this paper, the definition of good item distribution in knowledge hierarchy is presented. According to this definition and the model which formulates the personalized test generation problem [15], an improved personalized genetic algorithm which named PGAC and an improved genetic algorithm incorporated item distribution (IGAID) are presented. PGAC improves PGA in crossover process. PGA compares and crosses father individual  $ind_{dad}$  and mother individual  $ind_{mum}$  in the position which is chosen in a random probability. PGAC compares and crosses father individual  $ind_{dad}$  and mother individual  $ind_{mum}$  from the initial position to the end position. This will increase the probability of individual which includes more non-mastered items. IGAID which is PGAC incorporated item distribution assembles simulation test sheets which have good item distribution of mastered and non-mastered level in knowledge hierarchy for different students. Experiments and comparison are conducted to demonstrate the efficiency of PGAC and IGAID.

The rest of the paper is organized as follows. In Section 2, model and definition are presented. An improved genetic algorithm incorporated item distribution for students is proposed in Section 3. The experiments and evaluation are conducted in Section 4. Conclusion and future work are described in Section 5.

## 2 Model and definition

In this section, the model which formulates the personalized simulation test sheet generation problem is

described referring to our previous work [15], and the definition of good item distribution in knowledge hierarchy is presented.

Knowledge hierarchy of one course which includes 3 layers, namely chapter, section and concept is used as an example. One course includes several chapters, one chapter includes several sections and one section includes several concepts. The presented method can also be easily extended to practical situation.

The purpose of IGAID is to select  $q$  items from item bank to compose a test sheet that satisfies specific requirements, learning level and good item distribution in one course for each student.

Assume that there are  $n$  items  $Q_1, Q_2, \dots, Q_n$  in the item bank. There are  $m\_concept$  concepts  $concept_1, concept_2, \dots, concept_{m\_concept}$ ,  $m\_section$  sections  $section_1, section_2, \dots, section_{m\_section}$ ,  $m\_chapter$  chapters  $chapter_1, chapter_2, \dots, chapter_{m\_chapter}$  involved in the test. The relationship between concept and test item is one-to-many. The item  $Q_i (1 \leq i \leq n)$  for student  $j$  has 3 states: 0, 1, and 2. State 0 indicates student  $j$  answers item  $Q_i$  right. State 1 indicates student  $j$  answers item  $Q_i$  wrong. State 2 indicates student  $j$  has not answered item  $Q_i$ .

Constructing appropriate simulation test sheets which have good item distribution in knowledge hierarchy should consider the learning level of student  $j$  in mastered concept  $pref_{jconcept_s, 1 \leq s \leq m\_concept}$ , mastered section  $pref_{jsection_s, 1 \leq s \leq m\_section}$ , mastered chapter  $pref_{jchapter_s, 1 \leq s \leq m\_chapter}$  and preference information  $pref_{ji, 1 \leq i \leq n}$  for item  $Q_i$  which is corresponding to concept  $concept_s$ .

The definition of learning level for student  $j$  is the percentages of mastered concept  $pref_{jconcept_s}$ , mastered section  $pref_{jsection_s}$ , and mastered chapter  $pref_{jchapter_s}$ .

The definition of  $pref_{jconcept_s, 1 \leq s \leq m\_concept}$  is as follows:

$$pref_{jconcept_s, 1 \leq s \leq m\_concept} = \begin{cases} 0, YItem \geq answerNum * af, 0 < af \leq 1 \\ 1, YItem < answerNum * af, 0 < af \leq 1. \end{cases} \quad (1)$$

The definition of  $pref_{jsection_s, 1 \leq s \leq m\_section}$  is as follows:

$$pref_{jsection_s, 1 \leq s \leq m\_section} = \begin{cases} 0, YConcept \geq NumSection_s * bf, 0 < bf \leq 1 \\ 1, YConcept < NumSection_s * bf, 0 < bf \leq 1. \end{cases} \quad (2)$$

The definition of  $pref_{jchapter_s, 1 \leq s \leq m\_chapter}$  is as follows:

$$pref_{jchapter_s, 1 \leq s \leq m\_chapter} = \begin{cases} 0, YSection \geq NumChapter_s * bf, 0 < bf \leq 1 \\ 1, YSection < NumChapter_s * bf, 0 < bf \leq 1. \end{cases} \quad (3)$$

In equation (1), variable  $YItem$  indicates the number of correct answers to the items which correspond concept

$concept_s$ . Variable  $answerNum$  indicates the number of items which student  $j$  has answered. If all items which student  $j$  hasn't answered,  $pref_{jconcept_s}$  is 1 which represents student  $j$  hasn't mastered  $concept_s$ .

Variable  $af$  is a percentage which indicates that when no less than  $af$  items are answered right,  $pref_{jconcept_s}$  is 0 (0 represents student  $j$  has mastered the concept  $concept_s$ , 1 is not).

In equation (2), variable  $YConcept$  indicates the number of mastered concepts which correspond with section  $section_s$ . Variable  $NumSection_s$  indicates the number of concepts which section  $section_s$  contains. Variable  $bf$  is a percentage which indicates that when no less than  $bf$  concepts are mastered,  $pref_{jsection_s}$  is 0 (0 represents student  $j$  has mastered the section  $section_s$ , 1 is not).

In equation (3), variable  $YSection$  indicates the number of mastered sections which correspond with chapter  $chapter_s$ . Variable  $NumChapter_s$  indicates the number of sections which chapter  $chapter_s$  contains. Variable  $bf$  is a percentage which indicates that when no less than  $bf$  sections are mastered,  $pref_{jchapter_s}$  is 0 (0 represents student  $j$  has mastered the chapter  $chapter_s$ , 1 is not).

The expected percentages  $af$  and  $bf$  which are level of mastered concepts, sections and chapters in experiments could be set to 60%, for the reason that usually 60 is a passing grade. The percentages  $af$  and  $bf$  can also be set according to practical situation.

The definition of  $pref_{ji, 1 \leq i \leq n}$  for item  $Q_i$  which is corresponding to concept  $concept_s$  is as follows:

$$pref_{ji, 1 \leq i \leq n} = \begin{cases} 0, & \text{if student } j \text{ answers } Q_i \text{ right} \\ 1, & \text{if student } j \text{ answers } Q_i \text{ wrong} \\ pref_{jconcept_s}, & \text{if student } j \text{ has not answered } Q_i. \end{cases} \quad (4)$$

In equation (4),  $pref_{ji} = 0$  represents that student  $j$  masters item  $Q_i$ , and  $pref_{ji} = 1$  otherwise.  $pref_{ji}$  is a preference gene bit that is incorporated into crossover operator in PGAC and IGAID processing.

### 2.1 Model of personalized simulation test sheet generation problem

Based on our previous work [15], we can define each test item  $Q_i$  has three considerations including difficulty degree  $diff_{i, 1 \leq i \leq n}$ , discrimination degree  $dis_{i, 1 \leq i \leq n}$  and  $pref_{ji, 1 \leq i \leq n}$ . The multiple test requirements are item quantity  $q$ , expected difficulty degree  $diff$  and expected discrimination degree  $dis$ . The objective function can be defined as follows:

$$Minf = \frac{\sum_{i=1}^n |diff_i - diff|t_i + \sum_{i=1}^n |dis_i - dis|t_i}{2 * \sum_{i=1}^n t_i} - Z(pref_{j1}, \dots, pref_{ji}, \dots, pref_{jn}), \quad (5)$$

where

$$Z(pref_{j1}, \dots, pref_{ji}, \dots, pref_{jn}) = \begin{cases} 0.5, & \sum_{i=1}^n pref_{jit_i} < q * pf, 0 < pf \leq 1 \\ 1, & \sum_{i=1}^n pref_{jit_i} \geq q * pf, 0 < pf \leq 1. \end{cases} \quad (6)$$

In equation (5), variable  $t_i = 1$  represents that item  $Q_i$  is included in the test, and  $t_i = 0$  otherwise. In equation (6),  $pref_{ji}$  indicates whether student  $j$  masters item  $Q_i$  or not.  $\sum_{i=1}^n pref_{jit_i}$  indicates the number of items in the test that are not mastered by student  $j$ .  $q * pf$  represents the expected quantity of non-mastered items that should be selected in the final test sheet and  $Z(pref_{j1}, pref_{j2}, \dots, pref_{ji}, \dots, pref_{jn})$  is the preference information for student  $j$ . The objective function reduces the acceptable variation range of  $\frac{\sum_{i=1}^n |diff_i - diff|t_i + \sum_{i=1}^n |dis_i - dis|t_i}{2 * \sum_{i=1}^n t_i}$  when no less than  $q * pf$  of the final test items are non-mastered items relative to the model [15].

### 2.2 Definition of good item distribution in knowledge hierarchy

In this definition, assume that there are  $1, 2, \dots, q$  items in one test sheet and each concept, section and chapter have the same weight. The variables used in the formulated definition are as follows:

$sim\_concept_{it, 1 \leq i, t \leq q}$ : The knowledge similarity degree between item  $Q_i$  and item  $Q_t$  in concept level.  $sim\_concept_{it} = 1$  represents item  $Q_i$  belongs to the same concept  $concept_k$  with item  $Q_t$ , and  $sim\_concept_{it} = 0$  otherwise.

$sim\_section_{it, 1 \leq i, t \leq q}$ : The knowledge similarity degree between item  $Q_i$  and item  $Q_t$  in section level.  $sim\_section_{it} = 1$  represents item  $Q_i$  belongs to the same section  $section_k$  with item  $Q_t$ , and  $sim\_section_{it} = 0$  otherwise.

$sim\_chapter_{it, 1 \leq i, t \leq q}$ : The knowledge similarity degree between item  $Q_i$  and item  $Q_t$  in chapter level.  $sim\_chapter_{it} = 1$  represents item  $Q_i$  belongs to the same chapter  $chapter_k$  with item  $Q_t$ , and  $sim\_chapter_{it} = 0$  otherwise.

The definition of good item distribution is described from three levels which are chapter, section and concept. When item distribution in the test sheet satisfies constraints in equation (7 ~ 12), there is good item distribution in the test sheet.

In the chapter level, if  $chapter_k$  is a non-mastered chapter, item distribution in  $chapter_k$  should satisfy the following constraint:

$$\sqrt{\sum_{i=1}^q \sum_{t=1}^q sim\_chapter_{it}} \leq \left\lceil \frac{q}{\sum_{s=1}^{m\_chapter} pref_{jchapter_s}} \right\rceil. \quad (7)$$

$$\sqrt{\sum_{i=1}^q \sum_{t=1}^q sim\_concept_{it}} \leq \left\lceil \frac{q}{\sum_{s=1}^{m\_concept} pref_{jconcept_s}} \right\rceil. \quad (11)$$

In equation (7), when  $chapter_k$  is a non-mastered chapter, the arithmetic square root of summation of similarity degree between items belongs to  $chapter_k$  in chapter level is not more than  $\left\lceil \frac{q}{\sum_{s=1}^{m\_chapter} pref_{jchapter_s}} \right\rceil$ .

If  $chapter_k$  is a mastered chapter, item distribution in  $chapter_k$  should satisfy the following constraint:

$$\sqrt{\sum_{i=1}^q \sum_{t=1}^q sim\_chapter_{it}} \leq \left\lceil \frac{q}{m\_chapter} \right\rceil. \quad (8)$$

In equation (8), when  $chapter_k$  is a mastered chapter, the arithmetic square root of summation of similarity degree between items belongs to  $chapter_k$  in chapter level is not more than  $\left\lceil \frac{q}{m\_chapter} \right\rceil$ .

In the section level, if  $section_k$  is a non-mastered section, item distribution in  $section_k$  should satisfy the following constraint:

$$\sqrt{\sum_{i=1}^q \sum_{t=1}^q sim\_section_{it}} \leq \left\lceil \frac{q}{\sum_{s=1}^{m\_section} pref_{jsection_s}} \right\rceil. \quad (9)$$

In equation (9), when  $section_k$  is a non-mastered section, the arithmetic square root of summation of similarity degree between items belongs to  $section_k$  in section level is not more than  $\left\lceil \frac{q}{\sum_{s=1}^{m\_section} pref_{jsection_s}} \right\rceil$ .

If  $section_k$  is a mastered section, item distribution in  $section_k$  should satisfy the following constraint:

$$\sqrt{\sum_{i=1}^q \sum_{t=1}^q sim\_section_{it}} \leq \left\lceil \frac{q}{m\_section} \right\rceil. \quad (10)$$

In equation (10), when  $section_k$  is a mastered section, the arithmetic square root of summation of similarity

degree between items belongs to  $section_k$  in section level is not more than  $\left\lceil \frac{q}{m\_section} \right\rceil$ .

In the concept level, if  $concept_k$  is a non-mastered concept, item distribution in  $concept_k$  should satisfy the following constraint:

$$\sqrt{\sum_{i=1}^q \sum_{t=1}^q sim\_concept_{it}} \leq \left\lceil \frac{q}{\sum_{s=1}^{m\_concept} pref_{jconcept_s}} \right\rceil. \quad (11)$$

In equation (11), when  $concept_k$  is a non-mastered concept, the arithmetic square root of summation of similarity degree between items belongs to  $concept_k$  in concept level is not more than  $\left\lceil \frac{q}{\sum_{s=1}^{m\_concept} pref_{jconcept_s}} \right\rceil$ .

If  $concept_k$  is a mastered concept, item distribution in  $concept_k$  should satisfy the following constraint:

$$\sqrt{\sum_{i=1}^q \sum_{t=1}^q sim\_concept_{it}} \leq \left\lceil \frac{q}{m\_concept} \right\rceil. \quad (12)$$

In equation (12), when  $concept_k$  is a mastered concept, the arithmetic square root of summation of similarity degree between items belongs to  $concept_k$  in concept level is not more than  $\left\lceil \frac{q}{m\_concept} \right\rceil$ .

### 3 An improved genetic algorithm incorporated item distribution for students

In this part, IGAID which used for each student is presented. IGAID consists of two steps: items assembling with preference information and assembling with item distribution processing.

#### 3.1 Items assembling with preference information

In this procedure, PGAC is presented based on the algorithm named PGA that was presented in our previous work [15]. PGAC improves PGA in crossover process. PGAC chooses the crossover position of father individual  $ind_{dad}$  and mother individual  $ind_{mum}$  from the first item not in a random position. This will increase the probability of individuals which include more non-mastered items. The process of PGAC is described as follows.

Input: test items  $Q_1, Q_2, \dots, Q_n$ , expected difficulty degree, expected discrimination degree, item quantity  $q$  and expected percentage of level of non-mastered concepts of subject in test  $pf$ .

step 1 Initialize the population.

- step 2 Compute fitness according to equation (5).
- step 3 Go to step 6 if the termination criterion is satisfied.
- step 4 Generate new population according to the loop from step 4.1 to 4.3.
  - step 4.1 Select  $ind_{mum}$  and  $ind_{dad}$  according to roulette algorithm.
  - step 4.2 Crossover  $ind_{mum}$  and  $ind_{dad}$ .
    - 1) Decode  $ind_{mum}$  and  $ind_{dad}$ .
    - 2) Choose the head item position in  $ind_{mum}$  and  $ind_{dad}$  as the crossover position  $cp$ ,  $cp = 0$ .
    - 3) Get item  $Q_i$  in  $cp$  of  $ind_{mum}$  and item  $Q_r$  in  $cp$  of  $ind_{dad}$ .
    - 4) Compare gene bit  $pref_{ji}$  of  $Q_i$  and  $pref_{jr}$  of  $Q_r$ .
    - 5) When  $pref_{ji} = 0$  and  $pref_{jr} = 1$ , exchange  $Q_i$  and  $Q_r$ .
    - 6) Crossover position  $cp$  moves forward.
    - 7) If  $cp = q$ , encode  $ind_{mum}$  and  $ind_{dad}$ , and then go to step 4.3.
    - 8) Go to step 3).
  - step 4.3 Mutate.
- step 5 Go to step 2.
- step 6 Output the item set  $Q'_1, Q'_2, \dots, Q'_q$ .

In step 3, the convergence threshold of objective function which is defined in equation (5) is 0.5. The termination criterion of PGAC is  $Minf \leq 0.5$  or exceeding the maximum iteration. If this is satisfied, PGAC will find the best item sheet for the constraints set.

Step 4.2 describes the procedure of crossover operator. The crossover approach between  $ind_{mum}$  and  $ind_{dad}$  plays an important role to the convergence speed in assembling personalized test sheets for different students which meet students' preference information and learning level. The improved crossover pattern is in favor of assembling procedure.

The genes length of each item in the  $ind_k$  depends on the bounds of the items quantity  $q$ .

### 3.2 Item distribution processing

The procedure of item distribution processing is conducting detection and adjustment of item distribution on the output of items  $Q'_1, Q'_2, \dots, Q'_q$  in PGAC. It is described as follows.

- step 1 Get the output of the final item set assembling with preference information: test items  $Q'_1, Q'_2, \dots, Q'_q$ .
- step 2 The procedure in chapter hierarchy. For each chapter in the final set, calculate and compare item distribution in chapter level as equation (7) or equation (8), and execute the following step 2.1~ step 2.2 steps.
- step 2.1 If the chapter is not mastered and not satisfying

the constraint, random pick  $\sqrt{\sum_{i=1}^q \sum_{t=1}^q sim\_chapter_{it}}$   
 $-\lceil \frac{q}{\sum_{s=1}^{m\_chapter} pref_{jchapters}} \rceil$  items belong to this chapter out, and execute the following 1)~4) steps.

- 1) If there is a chapter which satisfies item distribution constraint in chapter level, contains the least items in current item set, and not be adjusted in chapter level procedure, pick the same number of new items with the same difficulty and discrimination degree and state is 2 from this chapter into the test set. If there are enough suitable items, go to step 2.
- 2) If there is not a suitable chapter in 1), pick the same number of new items with the same difficulty and discrimination degree and state is 2 from a new non-mastered chapter into the test set. If there are enough suitable items, go to step 2.
- 3) If there is not a suitable chapter in 2), pick the same number of new items with the same difficulty and discrimination degree and state is 2 from a new mastered chapter into the test set. If there are enough suitable items, go to step 2.
- 4) If there is not a suitable chapter in 3), pick the same number of new items with the same difficulty and discrimination degree and state is 2 from non-existing chapters corresponding items of the test set into the test set. If there are not enough items which state are 2, pick the same number of new items with the same difficulty and discrimination degree from non-existing chapters corresponding items of the test set into the test set. If there are enough suitable items, go to step 2.

step 2.2 If the chapter is mastered and not satisfying the

constraint, random pick  $\sqrt{\sum_{i=1}^q \sum_{t=1}^q sim\_chapter_{it}}$   
 $-\lceil \frac{q}{m\_chapter} \rceil$  items belong to this chapter out, and execute the above 1)~4) steps.

- step 3 The procedure in section hierarchy. For each section in the final set, calculate and compare item distribution in section level as equation (9) or equation (10), and execute the following step 3.1~ step 3.2 steps.
- step 3.1 If the section is not mastered and not satisfying

the constraint, random pick  $\sqrt{\sum_{i=1}^q \sum_{t=1}^q sim\_section_{it}}$   
 $-\lceil \frac{q}{\sum_{s=1}^{m\_section} pref_{jsections}} \rceil$  items belong to this section out, and execute the following 1)~4) steps.

- 1) If there is a section which satisfies item distribution constraint in section level, contains the least items in current item set, and not be adjusted in section level procedure, pick the same number of new items with the same difficulty and discrimination degree and state is 2 from this section into the test set. If there are enough suitable items, go to step 3.
- 2) If there is not a suitable section in 1), pick the same number of new items with the same

difficulty and discrimination degree and state is 2 from a new non-mastered section into the test set. If there are enough suitable items, go to step 3.

3) If there is not a suitable section in 2), pick the same number of new items with the same difficulty and discrimination degree and state is 2 from a new mastered section into the test set. If there are enough suitable items, go to step 3.

4) If there is not a suitable section in 3), pick the same number of new items with the same difficulty and discrimination degree and state is 2 from non-existing sections corresponding items of the test set into the test set. If there are not enough items which state are 2, pick the same number of new items with the same difficulty and discrimination degree from non-existing sections corresponding items of the test set into the test set. If there are enough suitable items, go to step 3.

step 3.2 If the section is mastered and not satisfying the

constraint, random pick  $\sqrt{\sum_{i=1}^q \sum_{t=1}^q sim\_section_{it}}$   
 $-\lceil \frac{q}{m\_section} \rceil$  items belong to this section out, and execute the above 1)~4) steps.

step 4 The procedure in concept hierarchy. For each concept in the final set, calculate and compare item distribution in concept level as equation (11) or equation (12), and execute the following step 4.1 ~ step 4.2 steps.

step 4.1 If the concept is not mastered and not satisfying

the constraint, random pick  $\sqrt{\sum_{i=1}^q \sum_{t=1}^q sim\_concept_{it}}$   
 $-\lceil \frac{q}{\sum_{s=1}^m pref_{jconcept_s}} \rceil$  items belong to this concept out, and execute the following 1)~4) steps.

1) If there is a concept which satisfies item distribution constraint in concept level, contains the least items in current item set, and not be adjusted in concept level procedure, pick the same number of new items with the same difficulty and discrimination degree and state is 2 from this concept into the test set. If there are enough suitable items, go to step 4.

2) If there is not a suitable concept in 1), pick the same number of new items with the same difficulty and discrimination degree and state is 2 from a new non-mastered concept into the test set. If there are enough suitable items, go to step 4.

3) If there is not a suitable concept in 2), pick the same number of new items with the same difficulty and discrimination degree and state is 2 from a new mastered concept into the test set. If there are enough suitable items, go to step 4.

4) If there is not a suitable concept in 3), pick the same number of new items with the same difficulty and discrimination degree and state is 2 from non-existing concepts corresponding items of the test set into the test set. If there are not enough items which state are 2, pick the same number of new items with the same difficulty and discrimination degree from non-existing concepts corresponding items of the test set into the test set. If there are enough suitable items, go to step 4.

step 4.2 If the concept is mastered and not satisfying the

constraint, random pick  $\sqrt{\sum_{i=1}^q \sum_{t=1}^q sim\_concept_{it}}$   
 $-\lceil \frac{q}{m\_concept} \rceil$  items belong to this concept out, and execute the above 1)~4) steps.

step 5 Output the final item set of IGAIID.

In step 2~step 4, if there are not enough items with the same difficulty and discrimination degree, the most similar ones would be picked to replace them.

## 4 Experiments and evaluation

To evaluate the performance of PGAC and IGAIID, a series of experiments are conducted by comparing them with random assembling algorithm and GA. The termination criterion of random assembling algorithm is

$\frac{\sum_{i=1}^q |diff_i - diff| + \sum_{i=1}^q |dis_i - dis|}{2q} < 0.5$  or it has been conducted 100 times. The objective function of GA is to calculate

the fitness of  $Min f = \frac{\sum_{i=1}^n |diff_i - diff|_{t_i} + \sum_{i=1}^n |dis_i - dis|_{t_i}}{2 * \sum_{i=1}^n t_i}$ . The

termination criterion of GA is  $f < 0.5$ .

The simulation experiments are conducted on four item banks. They are 6000 items, 10000 items, 20000 items and 30000 items, respectively. These items correspond to the same knowledge hierarchy. The knowledge hierarchy in our experiments contains 6 chapters, 21 sections and 94 concepts, which come from CETV-Web Evaluation Assessment System. CETV-Web Evaluation Assessment System can evaluate students' ability and learning attitude. Degree of difficulty and discrimination of items contains 5 grades: 1.0-the lowest, 2.0-the lower, 3.0-normal, 4.0-the higher, and 5.0-the highest. The expected percentage  $af$  and  $bf$  which are level of mastered concepts, sections and chapters in test are set to 60%.  $pf$  which represents the expected quantity of non-mastered items is set to 60% for the reason that usually 60 is a passing grade. The percentage  $pf$  can also be set according to practical situation. The initialized population size is set to 40.

The simulation experiments are conducted on five students named Zhao, Qian, Wu, Li, and Wang, respectively. The experiments aim to assemble test sheets

**Table 1** Students' mastered level in the item bank of 6000.

Knowledge	Student				
	Zhao	Qian	Wu	Li	Wang
Concepts	72.34%	41.49%	80.85%	65.96%	82.98%
Sections	76.19%	9.52%	80.95%	57.14%	90.48%
Chapters	66.67%	0%	83.33%	50%	83.33%

**Table 2** Students' mastered level in the item bank of 10000.

Knowledge	Student				
	Zhao	Qian	Wu	Li	Wang
Concepts	47.87%	13.83%	48.94%	64.89%	51.06%
Sections	14.29%	0%	23.81%	47.62%	28.57%
Chapters	16.67%	0%	0%	50%	16.67%

**Table 3** Students' mastered level in the item bank of 20000.

Knowledge	Student				
	Zhao	Qian	Wu	Li	Wang
Concepts	12.77%	5.32%	4.26%	19.15%	36.17%
Sections	4.76%	0%	0%	0%	9.52%
Chapters	0%	0%	0%	0%	0%

**Table 4** Students' mastered level in the item bank of 30000.

Knowledge	Student				
	Zhao	Qian	Wu	Li	Wang
Concepts	0%	0%	0%	1%	0%
Sections	0%	0%	0%	0%	0%
Chapters	0%	0%	0%	0%	0%

which assemble appropriate simulation test sheets with more non-mastered items and good item distribution. All algorithms used in the experiments are coded in Java Language conducted on a personal computer with Intel (R) Core (TM) 2 Duo CPU @ 2.53 GHz and 1.93 GB memory.

The students' mastered level contains three aspects: the percentage of mastered concepts, the percentage of mastered sections, and the percentage of mastered chapters. Table 1, Table 2, Table 3 and Table 4 describe the five students' mastered level in the item bank which contains 6000 items, 10000 items, 20000 items and 30000 items, respectively. The values are larger mean students get better mastered level. For example, as Table 2 shows that in the item bank of 10000 items, student Li has a good mastered level in concepts while student Qian has a bad mastered level in concepts. Students' learning level which is different from mastered level in calculation

method in concepts is an important attribute which influences the calculation of preference information for each student in equation (5) and equation (6).

**Table 5** Experiment results of difficulty and discrimination degree in test assembling for students in the item bank of 6000.

Student	V			
	Random	GA	PGAC	IGAID
Zhao	1.200	0.891	0.989	0.936
Qian	1.239	0.880	1.144	0.971
Wu	1.206	0.883	0.849	0.976
Li	1.211	0.880	1.117	1.005
Wang	1.159	0.898	0.837	0.916

**Table 6** Experiment results of difficulty and discrimination degree in test assembling for students in the item bank of 10000.

Student	V			
	Random	GA	PGAC	IGAID
Zhao	1.187	0.893	1.064	1.073
Qian	1.183	0.878	0.937	1.012
Wu	1.209	0.885	1.113	0.901
Li	1.220	0.900	1.138	1.167
Wang	1.197	0.890	1.047	0.985

**Table 7** Experiment results of difficulty and discrimination degree in test assembling for students in the item bank of 20000.

Student	V			
	Random	GA	PGAC	IGAID
Zhao	1.219	0.895	0.957	1.025
Qian	1.206	0.899	0.197	0.384
Wu	1.207	0.887	0.949	0.897
Li	1.228	0.894	1.003	1.173
Wang	1.192	0.895	0.999	1.016

The simulation experiments are conducted by applying four algorithms 10 times for achieving the best difficulty and discrimination degree when assembling item sheets of 30 items for 5 students. The average results are adopted to compare these four algorithms: random assembling, GA, PGAC, and IGAID.

In Table 5, Table 6, Table 7 and Table 8, the expected difficulty and discrimination degree of (*diff, dis*) are (1.0, 1.0), (2.0, 2.0), (3.0, 3.0), (4.0, 4.0) and (5.0, 5.0). V

**Table 8** Experiment results of difficulty and discrimination degree in test assembling for students in the item bank of 30000.

Student	V			
	Random	GA	PGAC	IGAID
Zhao	1.253	0.885	0.872	0.976
Qian	1.243	0.892	0.893	0.924
Wu	1.181	0.876	0.907	1.030
Li	1.173	0.887	0.926	0.902
Wang	1.203	0.891	0.933	1.009

indicates the average value of *diff* difference plus *dis* difference in 10 times for 5 students in every expected (*diff*, *dis*). *diff* difference between best difficulty and expected difficulty degree in the last test sheet is an absolute value. Also, *dis* difference between best discrimination and expected discrimination degree in the last test sheet is an absolute value. The smaller value of *V* indicates that the algorithm gets better experiment results of difficulty and discrimination degree.

There are more items corresponding to medium difficulty and discrimination degree. When assembling under expected difficulty degree and discrimination degree are (1.0, 1.0) and (5.0, 5.0), algorithms might not assemble appropriate items for last final sheet. It gives rise to the probability of bad final result quantity. Moreover, the average value of *V* isn't small.

#### 4.1 PGAC

To evaluate the performance of the presented PGAC, a series of experiments are conducted by comparing it with random assembling algorithm and GA in three aspects: final result quality, quantity of non-mastered items and execution time. As experiment results from Table 5 to Table 8 show, though final result quality of PGAC gets a medium position between random assembling algorithm and GA, PGAC gets 4 times best result quality, and the other result quality is more close to GA than random assembling algorithm. Considering  $Z(pref_{j1}, pref_{j2}, \dots, pref_{ji}, \dots, pref_{jn})$  which is the preference information for student *j* in the fitness function of PGAC would have an influence on the result quality for the value of *V*, the variation range of *V* is acceptable.

As Table 1, Table 2, Table 3, Table 4 and Table 9 show, random assembling algorithm might get good percentage which is above 60% when the percentage of mastered concepts of student mastered level is smaller than 5.32%. GA might get good percentage which is above 60% when the percentage of mastered concepts of student mastered level is smaller than 5.32%. PGAC might get good percentage which is above 60% when the percentage of mastered concepts of student mastered level is smaller 36.17%. The good times of random assembling

**Table 9** Experiment results on the percentage of non-mastered items in final test sheet applying random, GA and PGAC on each student in the different item banks

Algorithm for students	Item bank			
	6000	10000	20000	30000
Zhao_random	30.13%	45.26%	56.46%	74.60%
Qian_random	50.13%	58.66%	63.53%	77.46%
Wu_random	25.53%	39.20%	58.66%	90.00%
Li_random	36.66%	36.20%	50.53%	68.66%
Wang_random	19.80%	46.00%	49.93%	75.66%
Zhao_GA	30.93%	42.40%	54.26%	78.33%
Qian_GA	39.46%	59.26%	64.93%	79.40%
Wu_GA	23.80%	41.73%	56.80%	87.20%
Li_GA	38.53%	37.26%	50.20%	69.86%
Wang_GA	23.6%	41.53%	50.66%	77.86%
Zhao_PGAC	41.06%	59.26%	61.93%	76.73%
Qian_PGAC	59.33%	62.60%	65.33%	78.00%
Wu_PGAC	25.46%	58.26%	63.73%	89.66%
Li_PGAC	54.00%	54.33%	61.33%	71.53%
Wang_PGAC	23.73%	59.40%	60.33%	78.73%

algorithm, GA and PGAC achieved in Table 9 are 6, 6 and 11, respectively. PGAC gets the best results in quantity of non-mastered items in the final test sheet. The items of IGAID are conducted item distribution process on the items assembled by PGAC. The quantity of non-mastered items in the final test sheet of IGAID is almost the same as or more than PGAC.

#### 4.2 IGAID

To evaluate the performance of the presented IGAID, a series of experiments are conducted by comparing it with random assembling algorithm, GA and PGAC in three aspects: final result quality, average percentage of good item distribution and execution time. As experiment results from Table 5 to Table 8 show, final result quality of IGAID gets a relatively medium position between random assembling algorithm and GA.

Fig.1 presents the average percentage of good item distribution in final item sheets applying by four algorithms with selecting 30 items in different difficulty and discrimination levels for 5 students 10 times. Though the average percentages of good item distribution in final test sheets applying by random, GA and PGA are above 0.9, there are excess items corresponding with the same concept, section or chapter in more than one final test sheet. IGAID supports good item distribution in each assembled test sheet. As Fig.1 shows, the items by using IGAID get best item distribution in concept, section and chapter than random, GA and PGAC.

Fig.2 presents the average execution time of GA, PGAC, and IGAID with selecting 30 items in different



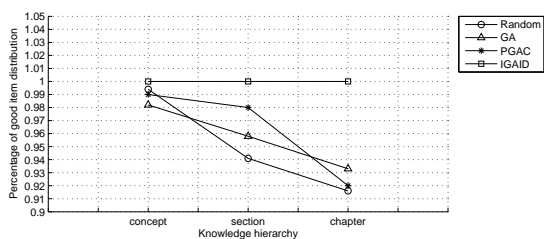


Fig. 1 The percentage of good item distribution in final test sheets applying four algorithms.

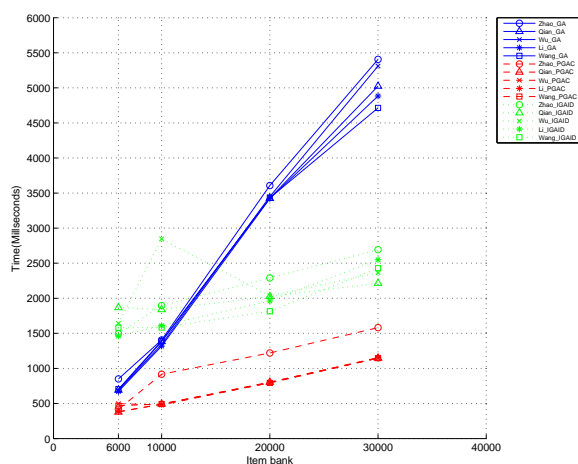


Fig. 2 Average execution time with selecting 30 items in different item banks for 5 students.

item banks for 5 students. Random assembling algorithm doesn't have the heuristic function to be restricted, it is the fast one in the all algorithms. It is took out in our experiments when comparing execution time applying these algorithms.

In execution time of GA, PGAC and IGAID, PGAC gets the best in these algorithms. It means the improvements in crossover procedure and heuristic function are very effective.

By contrast PGAC, IGAID has extra procedure of item detection and adjustment. As Fig.2 shows, IGAID only costs the more time which is almost a fixed value than PGAC costs. And IGAID costs far less time than GA when item bank are 20000 items and 30000 items. In other words, with the expansion of the scale of item bank, IGAID will cost less time than GA. It means IGAID is effective in assembling test sheets with good item distribution which satisfy multiple requirements and students' learning level.

## 5 Conclusion

In this paper, the detection and adjustment of item distribution in test sheet generation problem is formulated. PGAC and IGAID are presented. PGAC incorporates preference information of item and concept into the procedure of crossover operator and heuristic function to assemble more non-mastered items in final test sheet. Based on PGAC, IGAID incorporates item distribution into the test sheet assembling procedure to assemble test sheets which contain more non-mastered items and have good item distribution.

Experiment results show that PGAC can effectively select personalized test sheet for each student. Moreover, IGAID can effectively select personalized and good item distribution test sheet for each student.

In our future work, different knowledge weight could be incorporated to the procedure of item distribution detection and adjustment. Moreover, we could consider test sheets generation in distributed systems in the future work.

## Acknowledgement

This work was supported by the National Natural Science Foundation of China (project no. 61250010), CETV learning mall project, Beijing Municipal Commission of Education (grant no. 1320037010601) and the 111 Project of Beijing Institute of Technology. The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

## References

- [1] F. R. Wang, W. H. Wang, Q. K. Pan, F. C. Zuo, J. J. Liang, Proceedings of IEEE International Symposium on IT in Medicine & Education, **1**, 700-705 (2009).
- [2] P. Chen, A. B. Meng, C. H. Zhao, Proceedings of IEEE Congress on Evolutionary Computation, 2158-2162 (2008).
- [3] G.J. Hwang, B. M. T. Lin, T. L. Lin, Computers & Education, **46**, 122-139 (2006).
- [4] K. H. Tsai, T. I. Wang, T. C. Hsieh, T. K. Chiu, M. C. Lee, Expert Systems with Applications, **37**, 774-786 (2010).
- [5] G. J. Hwang, B. M. T. Lin, H. H. Tseng, T. L. Lin, IEEE Transactions on Systems Man and Cybernetics Part C-Applications and Reviews, **35**, 590-594 (2005).
- [6] G. J. Hwang, IEEE Transactions on Education, **46**, 329-337 (2003).
- [7] G. J. Hwang, P. Y. Yin, S. H. Yeh, IEEE Transactions on Education, **49**, 88-97 (2006).
- [8] M. Yildirim, Computer Applications in Engineering Education, **18**, 298-305 (2010).
- [9] P. Y. Yin, K. C. Chang, G. J. Hwang, G. H. Hwang, Y. Chan, Educational Technology & Society, **9**, 3-15 (2006).
- [10] S. C. Cheng, Y. T. Lin, Y. M. Huang, Expert Systems with Applications, **36**, 616-624 (2009).

- [11] C. L. Lee, C. H. Huang, C. J. Lin, Proceedings of New Trends in Applied Artificial Intelligence, **4570**, 823-833 (2007).
- [12] F. R. Wang, W. H. Wang, H. Q. Yang, Q. K. Pan, Proceedings of International Conference on Information Engineering and Computer Science, 1-4 (2009).
- [13] Y. Ou-Yang, H. F. Luo, Proceedings of 4th International Conference on Computer Science & Education, 170-173 (2009).
- [14] H. wang, C. Q. Ma, N. N. Chen, Proceedings of 5th International Conference on Computer Science and Education, 19-22 (2010).
- [15] P. P. Gu, Z. D. Niu, X. T. Chen, W. Chen, Proceedings of 10th International Conference on Web-based Learning, **7048**, 164-173 (2011).



mining and intelligent tutoring systems.

**Peipei Gu** received her master degree from Zhengzhou University. She is currently a PhD Student in the School of Computer Science and Technology at Beijing Institute of Technology, China. Her research interests include artificial intelligence, data



Pittsburgh. His research interests include computer software architecture, intelligent education system, digital library and neural information system.

**Zhendong Niu** received his PhD degree from Beijing Institute of Technology. He is currently a Professor in the School of Computer Science and Technology at Beijing Institute of Technology and an Adjunct Research Professor in the Information School at University of



intelligent tutoring systems.

**Wei Chen** received his PhD degree from City University of Hong Kong. He is currently a researcher in Agricultural Information Institute of Chinese Academy of Agricultural Sciences. His research interests include web data mining, distributed information retrieval, and



examination systems, and knowledge base management systems.

**Xuting Chen** received her master degree in Software Engineering at Beijing Institute of Technology. Her research interests are in the areas of artificial intelligence, web data mining, intelligent tutoring systems, computer-aided instruction systems, web-based



tutoring systems.

**Ke Niu** received his master degree from Beijing Institute of Technology. He is currently a PhD Student in the School of Computer Science and Technology at Beijing Institute of Technology, China. His research interests include artificial intelligence, data mining and intelligent tutoring systems.



intelligent tutoring systems.

**Jia Sun** is currently a Master Student in the School of Computer Science and Technology at Beijing Institute of Technology, China. Her research interests include data mining, distributed information retrieval and intelligent tutoring systems.