Information Sciences 532 (2020) 72–90

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Exam paper generation based on performance prediction of student group

Zhengyang Wu [a], Tao He [b], Chenjie Mao [a], Changqin Huang [c,*]

[a] School of Computer Science, South China Normal University, Guangzhou, China
[b] School of Information Technology in Education, South China Normal University, Guangzhou, China
[c] Department of Educational Technology, Zhejiang Normal University, Jinhua, China

ARTICLE INFO

Article history:
Received 16 December 2019
Received in revised form 6 March 2020
Accepted 26 April 2020
Available online 4 May 2020

Keywords:
Educational data mining
Exam paper generation
Deep knowledge tracing
Dynamic programming
Genetic algorithm

ABSTRACT

Exam paper generation is an indispensable part of teaching. Existing methods focus on the use of question extraction algorithms with labels for each question provided. Obviously, manual labeling is inefficient and cannot avoid label bias. Furthermore, the quality of the exam papers generated by the existing methods is not guaranteed. To address these problems, we propose a novel approach to generating exam papers based on prediction of exam performance. As such, we update the quality of the initially generated questions one by using dynamic programming, as well as in batches by using genetic algorithms. We performed the prediction task by using Deep Knowledge Tracing. Our approach considered the skill weight, difficulty, and distribution of exam scores. By comparisons, experimental results indicate that our approach performed better than the two baselines. Furthermore, it can generate exam papers with adaptive difficulties closely to the expected levels, and the related student exam scores will be guaranteed to be relatively reasonable distribution. In addition, our approach was evaluated in a real learning scenarios and shows advantages.

1. Introduction

The generation of exam questions is a challenging task in educational technology. The related research are roughly two categorizes. One is to use methods such as Natural Language Processing and Semantic Ontology to generate new questions from text or paragraphs [6,28], and these methods focus on generating natural questions. Another is to extract questions from the question bank [13,37], and the related methods consider the characteristics of the questions and their relevance to the student's learning state. The task of exam paper generation (EPG) extracts several different questions from the question bank. Similarly, EPG depends on the characteristics of the questions, and needs to consider the students' learning status.

EPG must consider various factors of an exam paper, such as its difficulty level, the coverage of assessed skills (synonyms of the knowledge points in this article), and the score of each question. Therefore, EPG is an optimization process with multiple objectives [8,22]. However, with regard to translating the difficulty of each question into the difficulty of the entire exam paper, the existing methods do not propose a reasonable solution. Moreover, the exam is designed for the whole student group, rather than for a single student, and thus the difficulty is essentially a relative indicator. Therefore, different students may have different feelings about the difficulty level of the same question. Unfortunately, most existing EPG methods

* Corresponding author.
E-mail address: cqhuang@zju.edu.cn (C. Huang).

https://doi.org/10.1016/j.ins.2020.04.043
0020-0255/

[36,16,24] rely on manually labeling the difficulty level of the question. This may be inefficient and produce label bias. Obviously, these methods cannot ensure that the difficulty of the generated exam paper is reasonable. Moreover, the existing EPG methods ignore an important issue that a good exam paper should be verified by the results of the exam [14,10]. That is to say, the existing EPG methods cannot measure the quality of the exam papers generated by them in practical applications. Although some research has noticed the relationship between the EPG and the rationality of the exam results[12], there is still a long way to go before reaching the ideal goal.

In actual teaching activities, a common solution to EPG is to update an old version exam paper. The teacher can update the questions and adjust the difficulty of the exam paper based on his/her knowledge of students' learning status. Constant adjustments make the distribution of skills and difficulty of the exam paper more and more reasonable. Inspired by this, we propose a novel exam paper generation approach based on performance prediction student groups. Where, the method of optimizing exam papers using dynamic programming named PDP-EG, and the other is using genetic algorithm named PGA-EG. In our study, skill is considered as the basic unit because it is the backbone embedded in various entities that appears during the learning process. Therefore, students' skill mastery level determines whether they can correctly answer question related to the corresponding skill.

Our approach adopts *Deep Knowledge Tracing* [25] to achieve the students performance prediction task based on students' exercise answered records. The exam paper generated by PDP-EG or PGA-EG could meet the difficulty level and skill weight requirements. The distribution of the achieved scores on our generated exam is close to that of requirement without manually setting the difficulty levels of the questions. The experimental results show that the exam papers generated by our approach are more advantageous than the baseline methods in terms of the main evaluation metrics of exam paper. In conclusion, the main contributions of our research are as follows:

- We propose a novel EPG approach that can generate the exam paper of a given difficulty without setting the difficult level of each question. In addition, the exam papers generated by our approach can ensure that the distribution of the achieved scores on exam are more reasonable.
- Our approach is able to well match the predicted student mastery levels of skills into the difficulty levels of the exam. This is achieved by using the multi-objective optimization algorithms.
- To the best of our knowledge, this is the first work on introducing the distribution of skill into EPG.

The organization of the paper is as follows: Section 2 starts with a focused review of some related work. Section 3 describes preliminaries and important notations. Sections 4 and 5 detail the proposed frameworks of PDP-EG and PGA-EG. Section 6 presents the setup of the experiments and results. And Section 7 draws conclusions for this paper.

2. Related work

In this section, we first review the relevant research on EPG, and then briefly introduce the recent research on learning performance prediction.

2.1. Exam paper generation

Existing work about EPG mostly focuses on the algorithms which select questions from a question bank, or designing and implementing the online EPG system based on the algorithm. The research direction can be roughly divided into four categories. The first category mostly focuses on the efficiency of question selection. Literature [16] proposes a system which utilizes fuzzy logic algorithm to randomly select questions in a different question bank without bias, thereby reducing manual operations. However, this approach requires dividing the question bank into a series of unbiased sub-banks in the first place. The second category focuses on balancing and improving the overall characteristics of the exam paper. In these studies, the EPG was treated as a multi-objective optimization problem, where the genetic algorithm is commonly used for multi-objective optimization problem solving [3]. The genetic algorithm-based EPG [36] or genetic algorithm variants-based EPG [34] shows obvious advantages. In these studies, an exam paper set is considered as a biological group, an exam paper is considered as an individual in this biological group, and a question in the exam paper is treated as chromosome that constitutes the individual. The EPG problem is transformed into a multi-objective optimization problem that combines factors such as difficulty and skill coverage. The third category is devoted to finding and matching questions based on the difficulty of the exam paper. For example, the model proposed by [7] employs an Apriori algorithm to match the difficulty of the existing question according to the difficulty set by the administrator, and then to select the corresponding difficulty questions from the question bank. The fourth category of research is to improve the randomness of the extraction to reduce the number of duplicate questions that appear in the exam paper. To achieve this goal, a shuffling algorithm is used in the EPG for question selection [12,20,31]. In addition, some researchers have studied the dynamic adjustment questions. For instance, the system proposed in [27] could automatically generate questions with certain keywords using randomization technique.

All existing methods have something in common: they rely on a question bank and manually setting the difficulty label for each question in the question bank, which will inevitably result in label bias.

2.2. Student learning performance prediction

Learning performance prediction is a hot research issue in educational data mining. There are several research dimensions for predicting learning effectiveness. Some are based on learning behavior [26,21], some based on student motivation [4,11], and others based on student emotions [32,2]. The method of predicting students' performance based on their skills level via test or exercise is called *Knowledge Tracing* [9]. The original version of knowledge tracing model is based on Bayesian method, so it is also called *Bayesian Knowledge Tracing* (BKT). In BKT, the learning process is divided into two stages based on the student's mastery of skills. The first stage is called the *unlearned stage*, in which students do not correctly answer questions about certain relevant skills. The second one is called the *learned stage* where the students correctly answer the questions about the relevant skills. In this model, the Hidden Markov model (HMM) is used to represent the probability distributions along the discrete time-line. BKT is a classic method of performance prediction, and there are also some improved studies on it. Based on three ways decisions theory, [39] proposed a model which add a *learning* stage between *unlearned* and *learned* stages of BKT. This method improves the accuracy of the prediction.

Recurrent Neural Networks (RNN) are special neural networks with the ability to capture input data at multiple time points. Therefore, they are particularly suitable for learning models of time series data. The question answer data is also time series data capturing the student learning process. And some researchers have used RNN to simulate the student learning process by analyzing the question answer data [33]. This method extends the task of *Knowledge Tracking* by applying flexible RNN, and is named *Deep Knowledge Tracing* (DKT). The learning model of DKT is Long Short Term Memory (LSTM) networks, which is a complex variant of RNN and is widely used in models of sequence prediction. Prior studies have shown that DKT is very effective in predicting and assessing learning performance [35,33]. After that, a number of studies on learning prediction have been developed by using RNN. Literature [19] used a DKT framework to predict a student next item response in MOOCs based on clickstream and video interaction sequence data. Literature [40] adopted LSTM on personalized learning full-path recommendation. Literature [30] proposed a model using RNN-based model to predict the performance of students based on trial text and answer records. Literature [15] traced student's state with RNN, and learned the encoding from exercise questions by a bidirectional LSTM.

3. Preliminaries

In this section, we first describe how to represent question and exam paper by skill. Then, we explain the working process of *Deep Knowledge Tracking* in our research and how to predict the exam score based on it. Table 1 shows some important notations. In the following section, we will give a more detailed explanation of their roles.

3.1. Representation of some entities in learning system

Since the entities in learning system, such as course, exercise, question, and exam paper are all related to skill, this research tries to use skill to represent some of their characteristics. A course consists of a range of skills, and each skill has a different level of importance (or weight). Therefore, a course with m skills could be represented as a vector $W = [w_1, w_2, w_3, \ldots, w_m]$, where each element of W represents the weight of corresponding skill. For example, W_A represents the weight vector of skill of the course A, where w_3 represents the weight of the third skill.

A question can be considered as a summary of one or more skills of a course. If a course includes m skills, then each question in this course may contain at least 1 and up to m skills. Similar to the representation of the course, we can also use a

Table 1
Some Important Notations.

Notation	Description
W	Weight vector of skills of a course
V	Allover skills list of the course
E	Exam paper with skill occurrence
$\overline{E}$	Occurrence probability of skills in exam paper
Q	The question of exercise(Q^e) or exam(Q^a)
$H = \{Q^e, a\}$	Exercise question answer records
x	Input data of DKT model
y	Output data of DKT model
S	A group students' skill mastery level
R	A list of exam scores of a group students
$i \in [1, k]$	The index of question in an exam paper
$j \in [1, m]$	The index of a skill in a course
$l \in [1, n]$	The index of a student in a group
$t \in [1, T]$	The index of an exercise question answered by a student
P	The probability distribution
QB	Question bank

vector $Q = [q_1, q_2, q_3 \ldots q_m]$to represent a question, where q_j does not represent the weight of the skill, but does the j-th skill appears or not in the question Q. For instance, in question Q, if the j-th skill does not appear, then q_j is 0, otherwise q_j is 1. During the process of studying, there may be many exercises raised for each student to answer. We denote the combination of exercise questions answered correctly or not by $H_t = \{Q_t^e, a_t\}$, where Q_t^e represents the t-th question answered by a student, and a_t represents the result of the answer.

The exam paper is a combination of a series of different questions. We assume an exam paper consisting of k questions, corresponding to a course with m skills. Then the exam paper can be represented by a matrix:

$$E = \begin{pmatrix} e_{11} & e_{12} & \ldots & e_{1m} \\ e_{21} & e_{22} & \ldots & e_{2m} \\ \ldots & \ldots & \ldots & \ldots \\ e_{k1} & e_{k2} & \ldots & e_{km} \end{pmatrix},$$

where the value of e_{ij} $(1 \leqslant i \leqslant k; 1 \leqslant j \leqslant m)$ is 0 or 1. Each row represents a question Q^a. $e_{ij} = 1$ indicates that the i-th question Q_i^a examine the j-th skill, $e_{ij} = 0$ indicates that the i-th question Q_i^a does not examine the j-th skill. We use the probability of occurrence of each skill to indicate the weight of each skill in this exam paper. For instance, scalar ξ_j represent the occurrence probability of j-skill, i.e.,

$$\xi_j = \frac{\sum_{i=1}^{k} e_{ij}}{\sum_{j=1}^{m}\sum_{i=1}^{k} e_{ij}} \tag{1}$$

The occurrence probability of each skill of an exam paper could be combined to a m length vector, i.e. $\overline{E} = [\xi_1, \xi_2, \xi_3, \ldots, \xi_m]$, which also shows the weight of the skills in the exam paper.

3.2. Deep knowledge tracing

DKT model adopts LSTM networks to achieve the goal that predicts the probability of correctly answered question by inputting the exercise answer records of the student. The details of process are shown in the Fig. 1. The input x is the one-hot encoded sequence data of exercise answered by a student at time t, each x is a vector representing the skill coverage and corresponding answer result label. Each element of vector y represents a prediction of the probability that the corresponding skill is correctly answered in next step. In our study, the probability of correctly answer skill can also be regarded as the skill mastery level.

3.3. Prediction of exam score

We denote the skill mastery level of l-th student by the vector $s_l = [s_{l1}, s_{l2}, s_{l3}, \ldots, s_{lm}]$, where s_{lj} represents j-th skill mastery level of l-th student. If there are n students in a group, the skill mastery level of all students can be connected into a $n \times m$ matrix, i.e:

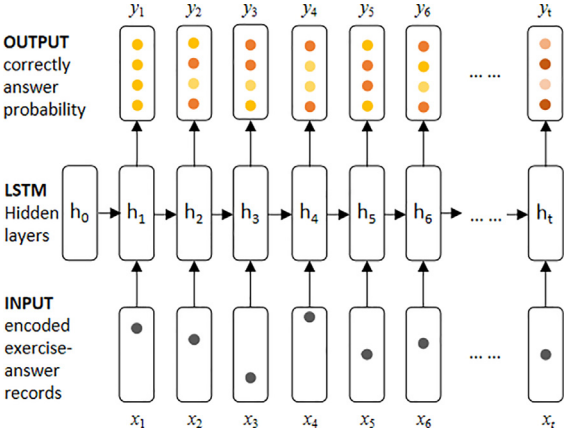

Fig. 1. Process of Deep Knowledge Tracing.

$$S = \begin{pmatrix} s_{11} & s_{12} & \dots & s_{1m} \\ s_{21} & s_{22} & \dots & s_{2m} \\ \dots & \dots & \dots & \dots \\ s_{n1} & s_{n2} & \dots & s_{nm} \end{pmatrix}.$$

The exam score is the sum of the scores for each question in the exam paper. Generally, the higher the skill mastery level of a student, the higher the score the student will receive. In this view, the scores obtained by the student are positively related to the similarity between the skills tested and the skills he/she mastered. Fig. 2 shows a toy example of the general process of student answer exercise questions and exam questions. The left part of the student icon shows the process of the student answering the exercise questions in a learning cycle($t \in [1, T]$), where the solid dot indicates that the corresponding skill in the exercise question appears, and the open dot indicates that the corresponding skill does not appear in the exercise question. Through doing exercises, the level of skill mastery of the student can be revealed, which is represented as a rounded rectangle with gray dots on top of the student icon, and different gray levels indicate the different mastery level on the corresponding skill. The right part of the student icon represents the process of exam, the solid dot indicates that the corresponding skill in the exam question appears, and the open dot indicates that the corresponding skill does not appear.

After obtaining the skill mastery level of the students (denoted by S) by using the method mentioned in Section 3.2, we will predict each student's exam score by simulating the process of answering the exam paper. We denote an exam paper by the matrix E. The score of each question can be obtained by the degree of association between the occurrence of the skill ($e_{ij} \in E$) and the skill mastery level of student ($s_{lj} \in S$). We denote the probability that l-th student correctly answers the i-th question by r_{li}, which can be expressed as:

$$r_{li} = \prod_{j=1}^{m} (s_{lj} | e_{ij} == 1), \tag{2}$$

where the right part of the Eq. (2) indicates that j-th skill appears in i-th question ($e_{ij} == 1$), and the skill mastery level of l-th student (s_{lj}) will be used as the multiplier for calculating r_{li}. For instance, there are five skills numbered from 1 to 5 in a course. A student's mastery level of each skill is combined into a vector of length 5, such as $s = [0.8, 0.4, 0.3, 0.9, 0.7]$. At the same time, we can also express a question in the exam paper as a 0–1 vector of length 5, where each element indicates whether the corresponding skill has appeared, such as $e = [1, 0, 0, 1, 1]$, which indicates that the first, fourth, and fifth skill appeared in this question. According to the Eq. (2), the probability that the student correctly answers this question can be obtained by multiplying the probability of the corresponding position in s, that is $0.8 \times 0.9 \times 0.7 = 0.504$.

Furthermore, we use the Eq. (3) to calculate the exam score, where r_l represents the score of l-th student on the exam paper. Then, the exam score r_l of l-th student can be expressed as:

$$r_l = \sum_{i=1}^{k} (r_{li} \times Score_i), \tag{3}$$

where the $Score_i$ represents the full score of i-th question. The exam score of each student is combined to a list $R = \{r_1, r_2, r_3, \dots, r_n\}$, which represents the exam scores of all students of a group. We denote the probability distribution of R by $P(R)$.

4. PDP-EG model

An idealized exam paper should cover all the skills of the course, but when the number of skills is large, such exam papers are actually difficult to obtain. In practice, the closer the weight distribution of skills in the exam paper is to the weight distribution of skills in the course, the better the exam paper is. A group of students whose skill mastery level is influenced by a number of random factors, such as ability and intelligence. As we know, when the value of a variable is affected by many random accidental factors, the shape of the probability distribution curve of the variable obeys the normal distribution function. The distribution of all values of this variable on the number axis is concentrated in the middle part. Therefore, the distribution of exam scores for a group of students should be consistent with a normal distribution. As mentioned above, we give two necessary conditions of a reasonable exam paper:

- Condition 1: The probability of occurrence of skills in the exam paper($\overline{E}$) should follow the skill weight of the course(W). It can be formalized as:

  $$\overline{E} \sim W.$$

- Condition 2: The exam scores distribution of a group students $P(R)$ should closer to a reasonable normal distribution $N(\mu; \sigma^2)$. It can be formalized as:

  $$P(R) \sim N(\mu; \sigma^2),$$

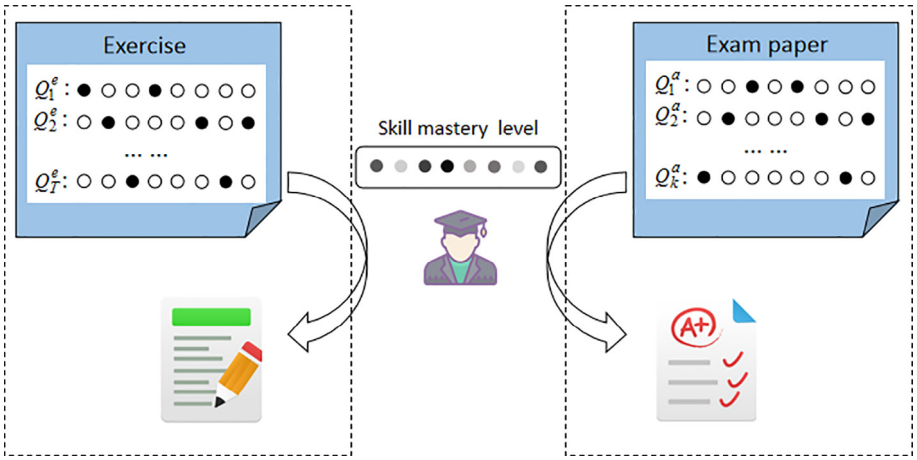

Fig. 2. Process of the student answer exercise questions and exam questions.

where $N(\mu;\sigma^2)$ is a normal distribution with mean μ and standard deviation σ.

In order to generate an exam paper that satisfies the above conditions, we propose a student group performance prediction based EPG model by adopting the dynamic programming (abbr.PDP-EG). Dynamic programming is a classic method that decomposes a complex problem into relatively simple sub-problems [17]. The outside data interaction of PDP-EG is shown in Fig. 3. The left part illustrates that the input of PDP-EG comes from students' exercise question answer records(H) and course characters(V and W), the exam paper(E) was organized by the question extracted from question bank. The middle is the main body of our approach, which will be explained in detail in Section 4.1, and the output is a valid exam paper set.

4.1. Framework of PDP-EG

Fig. 4 shows that there are two major steps in the framework of PDP-EG. Step 1 is the process of predicting the skill mastery level of the all students, i.e, *S*. In this step, the DKT model is used to implement the prediction task. Step 2 is the process of generating the exam paper, in which the exam paper will be synthesized, adjusted and optimized.

4.1.1. Prediction of students' skill mastery level

Step 1 of the PDP-EG framework is to apply a trained DKT model to predicting the skill mastery level of each student in a group. The training data of the DKT model is the exercise question answer records of all students who have participated in this course. During the training process, the LSTM network will output the predicted results in terms of the length of *V*. The rounded rectangle labeled DKT in the upper left corner represents the trained DKT model. After inputting the exercise question answer records one by one of each student, the model generates prediction vector for each student and these vectors will be combined to a $n \times m$ matrix *S*, which represents the skill mastery level of all students of a group.

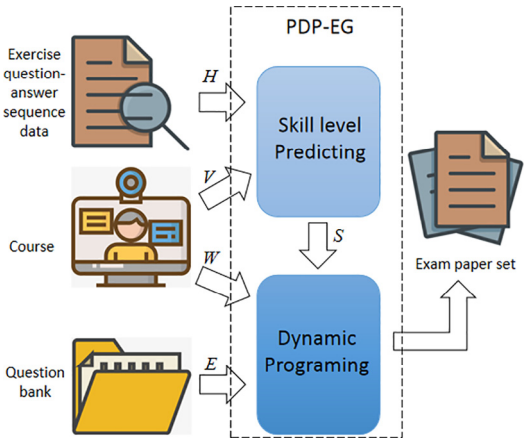

Fig. 3. The outside data flew interaction of PDP-EG.

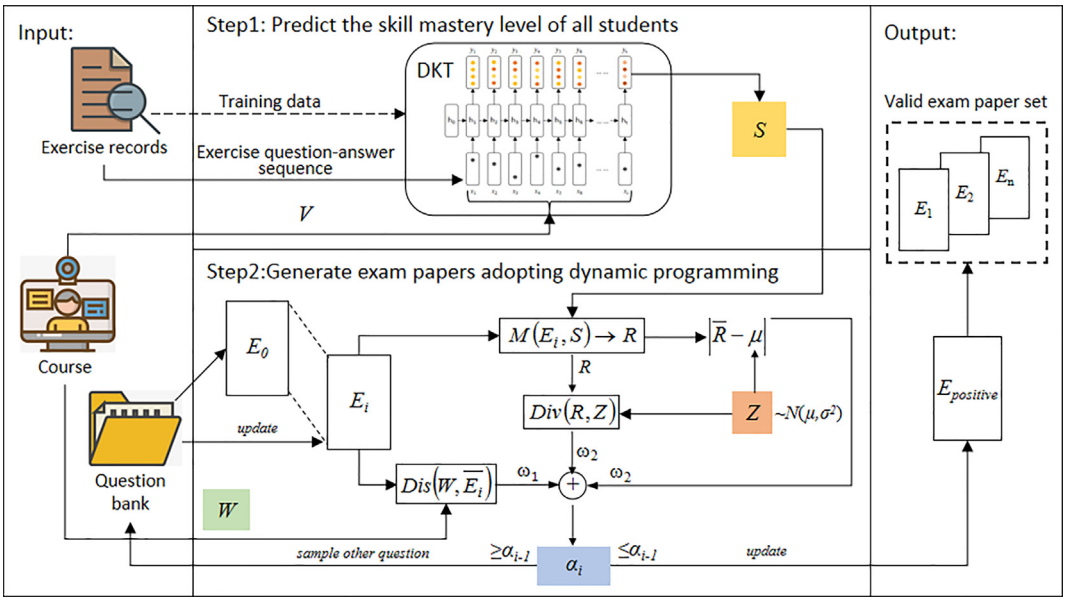

Fig. 4. Framework of PDP-EG.

4.1.2. Exam paper generating by adopting the dynamic programming

Step 2 is the core step in PDP-EG. First, a large number of exam papers are generated by randomly extracting questions from *QB*, and then an exam paper with the largest optimization factor value is filtered as the original version exam paper E_o. Next, in the process of comparing the optimization factors, the new version is continuously generated by updating the question in the exam paper one by one. As mentioned at the beginning of Section 4, an reasonable exam paper should satisfy two conditions. Therefore, we will determine the optimization factors based on these conditions.

In order to verify whether the exam paper satisfies Condition 1, we set the skill weight factor *Dis* to indicate the skill weight distance between the exam paper and the course. The *Dis* can be expressed as:

$$Dis = Distance(\overline{E}, W). \tag{4}$$

Here we use the Euclidean distance to calculate the skill weight distance between the exam paper and the course, i.e,

$$Dis = Dis_{Euclidean}(\overline{E}, W) = \sqrt{\sum_{j=1}^{m}(\xi_j - w_j)^2}. \tag{5}$$

In order to verify whether the exam paper satisfies Condition 2, we set the difficulty factor *Dif* and the score distribution factor *Div*. The difficulty factor reflects the difficulty level of the exam paper, and in general, we can use the average score instead. We denote the exam scores for all students by *R*. *R* is a collection of r_l, which can be obtained in the following steps. First, we use the method represented by the Eq. (2) to get the r_{li} by entering *S* and *E*. *l*-th student's score r_l be get by using the method represented by Eq. (3). Then, the difficulty factor *Dif* can be expressed as:

$$Dif = |\mu - \overline{R}|, \tag{6}$$

where μ is the expected difficulty, $\overline{R}$ is the average value of *R*.

In order to detect whether the scores meet the reasonable distribution, we generate a list *Z* satisfied the normal distribution, i.e, $Z \sim N(\mu, \sigma^2)$ to simulate the expected exam scores. Where the μ represents the average of the exam scores, and $\frac{\mu}{100}$ is used to represent the expected difficulty. The element of *Z* is the random integer in range $[0, 100]$. Then, the exam score distribution factor *Div* can be expressed as the divergence between the *R* and *Z*, i.e:

$$Div = Divergence(P(R), P(Z)). \tag{7}$$

Here we use the Kullback-Leibler divergence [5] to calculate the divergence between $P(R)$ and $P(Z)$, i.e,

$$Div = Div_{kl}(P(R), P(Z)) = -\sum_{l=1}^{n} P(R) log_2 \frac{P(Z)}{P(R)}. \tag{8}$$

4.2. Algorithm description

Algorithm 1 summarizes the process of generating exam papers using PDP-EG. The symbols V, k, n represent the length of valid exam paper set(VE-set), the number of questions in an exam paper, and the number of students in a group. W is a vector representing the skill weight of the course, and Z is a list of expected exam scores that follows the normal distribution. ω_1, ω_2 and ω_3 are the weights of optimization factors. At the beginning, the exercise question answer records H was input LSTM networks to train the DKT model. The first loop is the process of predicting the skill mastery of all students and obtaining the matrix S. After that, an exam paper set will be generated by sampling the question from QB. The Dis_i, Dif_i, Div_i and α_i of each exam paper are obtained in the second loop, where α is the weighted sum of Dis, Dif and Div, i.e, $\alpha = [Dis, Dif, Div] \cdot [\omega_1, \omega_2, \omega_3]$. Then, the original version exam paper is filtered according to the minimal α, noted as α_0. The third loop is the process of dynamic updating the exam paper by trying to replace each question in it in turn according to the value of α. First, the prediction scores of n students on the exam paper were obtained by Eq. (2) and Eq. (3), then the v-th version score list R^v will be get. After that, the Dis, Dif, Div and α are obtained in the same way as in the second loop. We noticed that each time a question is updated, the optimization factors of each temporary version exam paper is recalculated. In order to improve the efficiency of the algorithm, we transform the calculation process of the predicted score list by the following steps:

$$\begin{aligned} r_l^v &= r_l^{v-1} + \delta r_l \\ &= r_l^{v-1} + \left(\sum_{i=1}^{k} (r_{li}^v \times Score_i) - \sum_{i=1}^{k} (r_{li}^{v-1} \times Score_i) \right) \\ &= r_l^{v-1} + Score_v \times (r_{lv}^{'} - r_{lv}), \end{aligned} \tag{9}$$

where the v-th question update corresponds to the v-th version exam paper. r_l^v is the prediction score of l-th student on the v-th version exam paper. r_{lv} is the probability of l-th student correctly answers the v-th question of the exam paper, as the question update will be done one by one. $r_{lv}^{'}$ is the new probability that the v-th question will be answered correctly after it be updated. $Score_v$ is the all score of v-th question. Then, when the v-th question is updated, the R^v can be calculated as:

$$\begin{aligned} R^v &= \{r_l^v | 1 \leqslant l \leqslant n\} \\ &= \{r_l^{v-1} + Score_v \times (r_{lv}^{'} - r_{lv}) | 1 \leqslant l \leqslant n\}. \end{aligned} \tag{10}$$

Each time we update a question, a new version of the exam paper will be generated, and the version with the smaller α will be retained, formally:

$$\arg\min_{E} (\alpha^v, \alpha^{v-1}). \tag{11}$$

Algorithm 1 PDP-EG Algorithm

Input: H, V, QB, W, $Z \sim N(\mu; \sigma^2)$, n, γ, k, ω_1, ω_2, ω_3
Output: E
1: $DKT \longleftarrow LSTM(H)$ Training DKT model
2: **for** $l \leqslant n$
3: $s_l \longleftarrow DKT(V_l)$ Predict each student's skill mastery level
4: $S \longleftarrow add(s_l)$
5: **end for**
6: Extract questions from QB to get an exam paper set $\mathbb{E}$.
7: **for** $E_i \in \mathbb{E}$
8: $R_i \longleftarrow M(E_i, S)$Get the score list
9: $Dis_i \longleftarrow Dis_{Euclidean}(W, \overline{E_i})$Get the Dis by the method defined by Eq. (5)
10: $Dif_i \longleftarrow |\mu - \overline{R_i}|$Get the Dif by the method defined by Eq. (6)
11: $Div_i \longleftarrow Div_{kl}(P(R_i), P(Z))$Get the Div by the method defined by Eq. (8)
12: $\alpha_i \longleftarrow [Dis_i, Dif_i, Div_i] \cdot [\omega_1, \omega_2, \omega_3]$Record each optimization factor
13: **end for**
14: $\alpha^0 \longleftarrow min(\alpha_i)$Minimal optimization factor as initial value
15: $E^0 \longleftarrow E_{\alpha^0}$ Initialize the first version exam paper
16: $R^0 \longleftarrow M\left(E^0, S\right)$ Initialize the score list
17: **for** $v \leqslant \gamma$
18: Randomly select a new question Q from QB.

(*continued on next page*)

19: Get E^{v} by replacing v-th question of E^{v-1} with Q.
20: $R^{v} \longleftarrow M(Q,S)$update the score list by the method defined by Eq. (10)
21: $Dis \longleftarrow Dis_{Euclidean}\left(W, \overline{E^{v}}\right)$
22: $Dif \longleftarrow |\mu - \overline{R^{v}}|$
23: $Div \longleftarrow Div_{kl}(P(R^{v}), P(Z))$
24: $\alpha^{v} \longleftarrow [Dis, Dif, Div] \cdot [\omega_1, \omega_2, \omega_3]$Recalculate the optimization factor
25: $E \longleftarrow \arg\min_{E}(\alpha^{v}, \alpha^{v-1})$ Retain the version with smaller optimization factor
26: **end for**

5. PGA-EG model

Generating a good exam paper needs to meet the three goals, namely skill weight, difficulty, and distribution of scores simultaneously, and thus it is a multi-objective optimization problem. In the solution of multi-objective problems, genetic algorithms have good performance. Therefore, in this section, we propose another performance prediction based EPG approach, which adopts improved genetic algorithm (abbr.PGA-EG). Like the PDP-EG method, this method also uses *Dis*, *Dif* and *Div* as optimization factors.

5.1. Framework of PGA-EG

Fig. 5 shows the framework of PGA-EG includes three parts. Left part is *Population* representing a set of exam papers consisting of questions selected from the question bank.

The upper part of the right side is the *Evaluation* module for evaluating the exam papers in the *Population*. In *Evaluation*, we set three metrics for the evaluation of each exam paper, i.e. *Dis*, *Dif* and *Div*. Where *Dis* indicates the distance between the

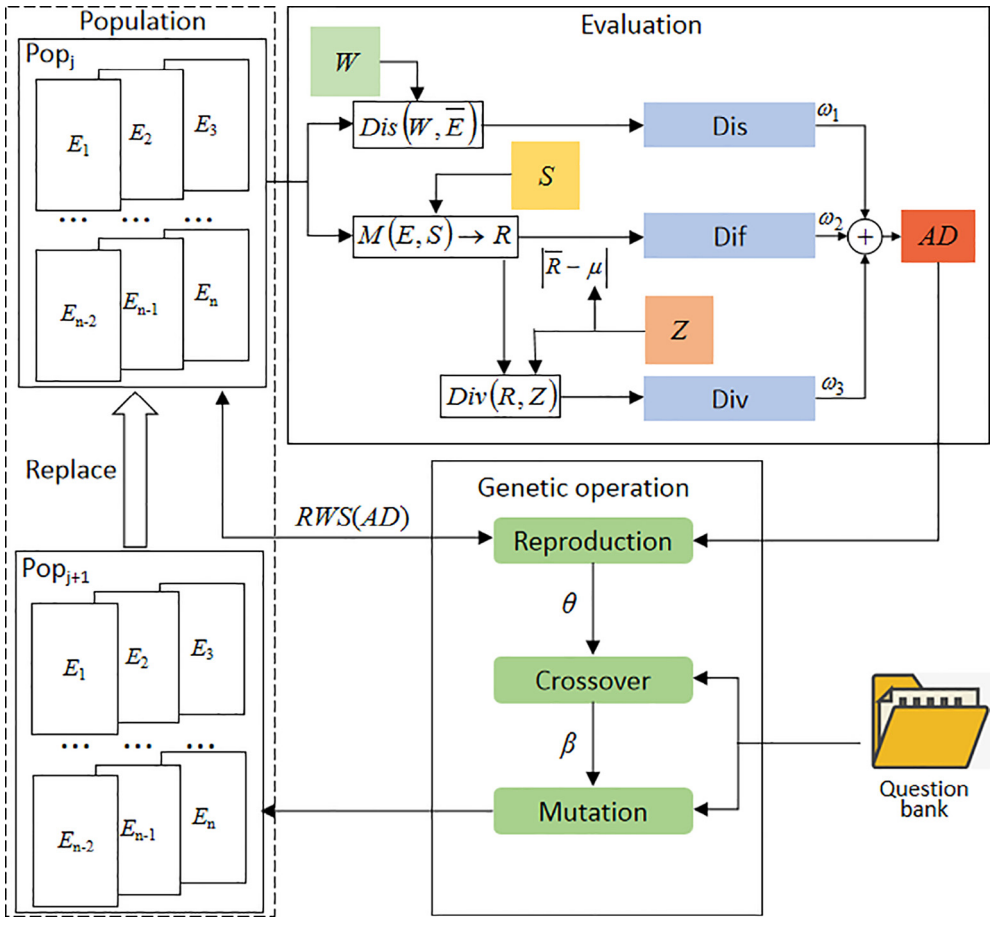

Fig. 5. Framework of PGA-EG.

skills' weight distribution of the course W and skill occurrence probability of exam paper $\overline{E}$. *Dif* indicates the distance between the prediction difficulty and the expected difficulty. *Div* represents the divergence between $P(Z)$ and $P(R)$. The values of these three indicators are multiplied by the weights $(\omega_1, \omega_2, \omega_3)$, and then the three products are summed, which is the *adaptation degree* of each exam paper, i.e. *AD*.

The lower part of the right side is the module of *Genetic operation* which includes three genetic operations, i.e. *Reproduction*, *Crossover* and *Mutation*. The *Reproduction* determines whether the exam paper is retained or eliminated in the next generation based on its *AD*. Here we use the classic Roulette-Wheel Selection (RWS) algorithm [1] based on *AD* to select the exam paper that can enter the next genetic operation. If the *AD* value of the exam paper is higher, the probability of being selected is greater. In the *Crossover* stage, any two exam papers are paired, and each pair of them exchanges questions according to the ratio θ, thereby generating the new exam paper. The *Mutation* is the stage of updating the question in the exam paper. Just like a genetic mutation, we set the probability of its occurrence to β. In this method, *Mutation* is a small probability event. After the above process, a new generation of exam paper *Population* will be produced.

5.2. Algorithm description

Algorithm 2 summarizes the process of PGA-EG. The symbols k, n represent the number of questions in an exam paper, length of exam paper Population. The symbols γ is the number of iterations of evolution. As same as Algorithm 1, the symbol W represents skill weight of the course, Z represents a list of expected exam scores, and the first four lines of code are the process of getting S. Next, we initialize the first generation exam paper population by randomly selecting question from *QB*. In the first sub-loop, we traversal each exam paper from Pop_j, and calculate their *Dis*, *Dif* and *Div*, furthermore get the *adaptation degree* of each exam paper AD_j. In the second sub-loop, first select two exam papers from Pop_j by using RWS algorithm. And then do the *Crossover* operation(Exchange a small part of questions) and *Mutation* operation(Update questions slightly) according to the preset probability of its occurrence, i.e. θ and β. At the end, new two exam papers will be added to the next generation population Pop_{j+1} until the population set is full. The new generation of population will replace the previous generation of populations and do the same operation until the number of iterations equals to γ.

Algorithm 2 Performance Prediction Based Exam paper Generation Gene Algorithm

Input: H, V, QB, W, $Z \sim N(\mu; \sigma^2)$, n, γ, k, θ, β, ω_1, ω_2, ω_3
Output: Exam paper-Population
$DKT \longleftarrow LSTM(H)$ Training DKT model
for $l \leqslant n$
  $s_l \longleftarrow DKT(V_l)$ Predict skills level of student l
  $S \longleftarrow add(s_l)$ Add the $s_l$to a group students' skill mastery level matrix S
end for
$Pop_0 \longleftarrow loop_n(Sample(QB, k))$ initialize the first generation exam paper population
for$j \leqslant \gamma$
  for $i \leqslant n$
    $E_i \longleftarrow Select\left(Pop_j\right)$ Sequential traversal exam paper sample E_i from Pop_j
    $Dis_i \longleftarrow Dis_{Euclidean}\left(W, \overline{E_i}\right)$ Calculate the *Distance* of E_i
    $R_i \longleftarrow M(E_i, S)$ Calculate exam scores
    $Dif_i \longleftarrow |Average(R_i) - \mu|$ Calculate the *Difficulty* of E_i
    $Div_i \longleftarrow Div_{kl}(P(R_i), P(Z))$ Calculate the *Divergence* of E_i
    $ad_i \longleftarrow 1 - (Dis_i * \omega_1 + Dif_i * \omega_2 + Div_i * \omega_3)$ Calculate the adaptation degree of E_i
    $AD_j \longleftarrow add(ad_i)$ Add each adaptation degree to AD_j
  end for
  for $i \leqslant n$
    $E_a, E_b \longleftarrow RWS\left(Pop_j, AD_j, 2\right)$ Roulette-wheel select any two exam papers from Pop
    if $Random(x) \leqslant \theta$
      $\acute{E}_a, \acute{E}_b \longleftarrow Crossover(E_a, E_b)$ Exchange a small part of questions
    end if
    if$Random(x) \leqslant \beta$
      $E'_a, E'_b \longleftarrow Mutation\left(\acute{E}_a, \acute{E}_b\right)$ Update questions slightly
      $Pop_{G_{j+1}} \longleftarrow \{E'_a, E'_b\}$ Add two new exam papers to the next generation population
    end if
  end for
end for

5.3. Comparison between PGA-EG and PDP-EG

The PDP-EG and the PDP-EG proposed in Section 4 are both based on the performance prediction of a group of students. As can be seen from the comparison of Fig. 4 and Fig. 5, there are three same inputs for them. The first one is the students' skill mastery level, which is represented by a yellow square labeled *S*. The other one is the skill weight of the course, which is represented by a cyan square labeled *W*. The third is a list of expected exam scores that meet a normal distribution, which is represented by a pink square labeled *Z*. In Algorithm 1 and Algorithm 2, all these three inputs are used to calculate the optimization factor or adaptation degree to detect the performance of the generated exam paper sample. Their major difference is the method of exam paper version updating. In the PGA-EG, the Algorithm 1 performs a batch update of the exam paper samples. After each iteration, the PGA-EG will produce as many exam paper samples as the size of population, differently, PDP-EG gets the best exam paper sample from the population at the beginning of iteration, and obtains one exam paper sample after each iteration. PDP-EG is a multi-stage decision-making method. This method does not consider the past states and decisions, but adopts a strategy that only considers the states formed by the current decisions, and all subsequent decisions must form the optimal solution. Therefore, PDP-EG can be regarded as a continuous recursive process, and the exam papers of each stage version are better than or at least equal to the previous version. This method updates the exam papers step by step with randomly sampled questions, just like the way that teacher manually update the exam paper. However, the random sampling method is difficult to ensure that the global optimal solution is found quickly. PGA-EG searches in the solution space according to probability, so it can effectively avoid local extremum and approach the global optimal solution with higher efficiency. However, PGA-EG transforms the problem of exam paper generation into a chromosome coding sequence problem, which has nothing to do with the connotation and characteristics of the problem of exam paper generation, so this method is weakly interpretable.

6. Experiments

In this section, we conduct experiments to test the performance of the proposed PDP-EG and PGA-EG, and compare them to two baselines.

6.1. Datasets and data preprocessing

Our experiments use three open datasets. Two of them are the ASSISTments dataset [23] which are the electronic tutor that teaches and evaluates students in grade-school math. These two datasets are large and gathered in different school years. The one named ASSISTments0910 was collected in the 2009–2010 school year and consisted of 278,607 records. The other named ASSISTments2015 was collected in the year of 2015 and consisted of 708,631 records. The third dataset was named OLI Engineering Statics which was collected from a college-level engineering statics course in the year of 2011 (abbr.OLIES2011), and consisted of 45,002 records. OLIES2011 is available from the PSLC DataShop website [29]. We can extract the exercise question bank from these three datasets, but there is no exam question bank, so we simulated three banks of exam questions with reference to the appearance of skills in the exercises, which corresponding to three datasets. The processing details of question banks for each dataset are as follows.

The ASSISTments0910 dataset contains the *problem_id* field, so we can extract an exercise bank from it, which includes 17,751 questions. We randomly select 15,000 questions from this exercise bank to form a simulated question bank.

The OLIES2011 dataset also contains the *problem_id* field, and we can extract 300 questions from it. In order to increase the number of optional questions, we randomly generated 700 questions by referring to the skill distribution of these 300 questions to form a simulated question bank containing 1,000 questions.

The ASSISTments2015 dataset does not contain the *problem_id* field. It is the same as the ASSISTments0910 dataset for grade-school math courses, but obtained from different school year. In the exercise bank of ASSISTments0910, a question contains up to 5 skills. Therefore, we randomly generated a simulated question bank for ASSISTments2015 dataset with up to 5 skills in each question.

We set the number of questions for each dataset to be different in order to compare the impact on the exam paper generation effect when the size of the question bank is very different(15,000, 10,000, 1000). However, this effect is not obvious from the experimental point of view. We wanted to set the ASSISTments2015 exam bank size to 7500 at beginning, but considering that this dataset has the most interaction records(708,631 records) in the three datasets, it may be that the question bank is relatively large, so we set the size of it to 10,000. For the comparative experiment, we set the difficulty label for each question in the question bank. Since there is no difficulty label in the original datasets, we simulate the difficulty label for each question based on the correct rate of each skill in the exercise question answer records. The difficulty label $difficulty_{Q^a}$ of the question Q^a can be expressed as:

$$difficulty_{Q^a} = \frac{\sum_{j=1}^{m} (correct_j | q_j == 1)}{\sum_{j=1}^{m} q_j}, \tag{12}$$

where m is the number of skills of the course, q_j represents whether the j-th skill appears or not in the question Q^a(as mentioned in Section 3.1), $correct_j$ is the correct rate of j-th skill.

In addition, we train the DKT model on each dataset. Before training, we converted the dataset into the exercise question answer record sequence data with each student as a unit. We use the first 70% of the sequence data as the training data, use 20% of the sequence data as the verification data, and reserve the last 10% of the sequence data as the test data. We measure the area under the curve (AUC) on each dataset to evaluate the performance of the DKT model. The AUC of the DKT model on ASSISTments0910 is 0.799, on ASSISTments2015 is 0.753, and on OLIES2011 is 0.701. The Table 2 shows the datasets summary.

6.2. Baseline methods

We compare PDP-EG and PGA-EG with two classic EPG methods. The one is genetic algorithm-based EPG method abbreviated as GA [36], another is that reference the randomization method to generate exam papers [20] and filter according to the preset conditions, which is called RSF. The details of the two methods are as follows.

- **GA** is a classic method of generating exam papers. In this method, the exam paper is regarded as an individual, and the question is regarded as a chromosome. The combined value of expected difficulty and skill coverage is taken as the goal of evolution. The difficulty of the exam paper is based on the difficulty label of the questions that make up the exam paper, which were obtained by Eq. (12).
- **RSF** adopts randomly draws questions from QB to compose the samples of exam paper, and filters these samples based on the conditions that the skill coverage and expected difficulty. Like GA, the difficulty of the exam paper is also based on the difficulty label which is obtained by Eq. (12).

A comparison of the our approach and baselines is shown in Table 3.

6.3. Evaluation setup

In order to imitate the real exam scenario, we randomly sample 50 students from each dataset to form a class, and obtain skill mastery level of them by using correspond DKT model. For the sake of simplicity, we set the all score of each question in *QB* of 1, and set the each exam paper with 100 questions. To compare the effects more objectively, we generate 100 exam paper samples by each method. see Table 4.

Generally, in a large-size question bank, the probability of occurrence of skills in the question reflects the skill weight of the corresponding course. In other words, the higher the probability that the skill appears in question bank, the greater the skill weight in the course. Therefore, in the absence of preset course skill weights, we use the question appearance probability vector of the *QB* as the skill weight vector of the course W. The parameters used in the different method are set as follows:

- In GA, we set the expected difficulty of exam paper of 0.7. For genetic operation, we set the crossover rate of 0.8, and the mutation rate of 0.003; the size of population of 1,000, and the number of evolution generations of 100.
- In PDP-EG, we set the mean of Z of 70 and the standard deviation of 15. The termination condition of the iteration is updated once for each question of exam paper, or the optimization factor stops decreasing over 50 iterations.
- In PGA-EG, we also set the mean of Z of 70 and the standard deviation of 15. As same as GA, we set the crossover rate of 0.8 and the mutation rate of 0.003; we set the size of population of 1,000, and the number of evolution generations of 100.
- In RSF, we set the expected difficulty of exam paper of 0.7. We set the threshold of the distance between the difficulty of the exam paper and the expected difficulty of 0.085, and the threshold of the skill coverage of 0.9.

In Section 6.4, we will explain why the expected difficulty is 0.7, the mean of Z is 70, and the standard deviation of Z is 15.

6.4. Evaluation metrics

We evaluate methods based on the actual needs applied to the exam assignments. The literature [10,38] give the evaluation index of the exam paper. Among the widely used are the *Difficulty* and *Validity* two indicators. The *Difficulty* index

Table 2
Overview of the Datasets.

Data set	Record	Skill	Student	Size of QB
ASSISTments0910	278,607	123	4,163	15,000
ASSISTments2015	708,631	100	19,917	10,000
OLIES2011	45,002	85	335	1,000

Table 3
Comparison of proposed methods and baselines.

Method	GA	PDP-EG	PGA-EG	RSF
Need preset questions' difficulty	Yes	No	No	Yes
Reference exercise answer record	No	Yes	Yes	No
For a given student group	No	Yes	Yes	No
Reference the skill weight of the course	No	Yes	Yes	No

Table 4
Summary of method parameter.

Method	GA	PDP-EG	PGA-EG	RSF
Number of sample	100	100	100	100
Target of difficulty	0.70	0.70	0.70	0.70
Target of skill included	Coverage	Weight	Weight	Coverage
Target of scores distribution	Not set	$N\left(70, 15^2\right)$	$N\left(70, 15^2\right)$	Not set
Number of iterations	100	$\leqslant$150	100	Unrestricted
Crossover and mutation rate	0.8,0.003	Not set	0.8,0.003	Not set
Size of population	1000	1000	1000	Not set

The **population** of PDP-EG refers to the set of exam paper generated at the beginning.

reflects the difficulty level of the exam paper. If most students get the higher scores, that means the exam paper is too easy; to the opposite, if most students get the lower score, that means the exam paper is too hard. The average score is generally used to indicate the difficulty of the exam paper, which can be calculated as:

$$Difficulty = \frac{Average_{score}}{100}, \tag{13}$$

where $Average_{score}$ represents the average score of students of a group. When the value of the *Difficulty* index is larger, the exam paper is easier. As stated in the literature of [38,18], the *Difficulty* of exam paper should be controlled near 0.7. If *Difficulty* is more than 0.75, it indicates that the exam paper is quite easy. While *Difficulty* is less than 0.45, it indicates the exam paper is rather difficult.

The *Validity* index refers to the skill coverage of the exam paper. In the evaluation plan, we upgrade this index to the distance between the skill occurrence probability of the exam paper and the skill weight of the course. The distance here is the Euclidean distance. The *Validity* can be expressed as:

$$Validity = 1 - Dis_{Euclidean}\left(E^C, W^C\right), \tag{14}$$

where $E^C$is the skill occurrence probability of the exam paper, and $W^C$is the skill weight of the course C.

Furthermore, the literature [18] introduced a reasonable exam paper should discriminate between the students who score high on the exam and those who score low. Therefore, they proposed a discrimination indicator to evaluate an exam paper. The discrimination index can be calculated by following formula:

$$discrimination = \frac{P_H - P_L}{100}, \tag{15}$$

where P_H is the average score for the 27% of those with highest marks, and P_L is the average score for the 27% of those with lowest marks. According to the literature [18], if the discrimination indicator in the range of (0.30, 0.39), the exam paper is qualified, if the discrimination indicator larger than 0.39, the exam paper is excellent.

We can estimate the standard deviation σ of Z based on this discrimination indicator, where Z represents a list of expected exam scores meets normal distribution, whose elements is the random integer in range $[0, 100]$. For simplify the calculation, we denote the distance between P_H and P_L by $Dis_{P_H P_L}$, where

$$Dis_{P_H P_L} = discrimination \times 100 = P_H - P_L.$$

The center point of the low score area of 27% is denoted by $\hat{P_L}$, and the center point of the high score area of 27% is denoted by $\hat{P_H}$. Then, $\hat{P_L}$ and $\hat{P_H}$ are the two-sided 27%-quantiles of the distribution(That is, 13.5% of the regions are divided on both sides of the distribution). In the case where the values of P_H and P_L cannot be clarified, we use the distance between $\hat{P_L}$ and $\hat{P_H}$ as an approximation of the distance between P_H and P_L.

We according to the table of standard normal distribution get the coefficient of quantile, and then the normal distribution transform is used to obtain the expressions of $\hat{P_L}$ and $\hat{P_H}$:

$$\hat{P_L} = -1.103063 \times \sigma + \mu,$$

and

$$\hat{P_H} = 1.103063 \times \sigma + \mu,$$

where μ is the mean of Z. The coefficient -1.103063 and 1.103063 is the two-sided 27%-quantile of the standard normal distribution. The distance between $\hat{P_H}$ and $\hat{P_L}$ is denoted by $Dis_{\hat{P_H}\hat{P_L}}$, where $Dis_{\hat{P_H}\hat{P_L}} = \hat{P_H} - \hat{P_L}$. Then,

$$Dis_{P_HP_L} \approx Dis_{\hat{P_H}\hat{P_L}} = 2.206126 \times \sigma.$$

If the $Dis_{P_HP_L} \in (30, 39)$, then the $\sigma \in (13.59, 17.67)$. We choose some different σ in range of $(13.59, 17.67)$ to verify this result. Three columns on the left of the Table 5 shows the average of $Dis_{P_HP_L}$ and the average of the maximum with 10,000 normal distributions, with different σ. These normal distributions are composed of 50 random numbers, and with a mean of 70. We set the target value of $Dis_{P_HP_L}$ is 39, the target value of maximum is 100 (The highest score of the exam), and use Euclidean metric to evaluate the target difference of each σ value, which are shown on the rightest column.

Therefore, as mentioned in Section 6.3, the most suitable standard deviation of Z is 15, and our target distribution of exam scores is $Z \sim N\left(70, 15^2\right)$. We set the difference between the distribution of exam scores and Z as the evaluation metric and named it *Rationality*, which is obtained by calculating the Kullback–Leibler divergence, which can be expressed as:

$$Rationality = 1 - Div_{kl}\left(P\left(R_E^S\right), P(Z)\right), \tag{16}$$

where $R_E^S$is the prediction scores of students on the exam paper E, whose skill mastery level matrix is S. In the next section we will show the results of the experiment, and analyze the reasons.

6.5. Results and discussions

In this section, we will first show the result on the indicators *Difficulty* and *Validity* separately, and then analyze the potential relationship between these two indicators. Next, we will show the indicator *Rationality* for each method, and compare them more intuitively by showing the probability distribution figure of prediction score.

6.5.1. Difficulty and validity

In our experiment, the expected value of indicator *Difficulty* is 0.70, which was calculated by simulating the use of exam paper. This process was based on skill predictions of 50 students by using DKT models. Table 6 shows the *Difficulty* of exam papers generated by using PDP-EG and PGA-EG are more advantage than the GA and RSF. Moreover, PGA-EG has a more stable effect on the dataset *ASSISTments2015*. Fig. 6 shows the comparison of the indicator *Difficulty* for 100 exam paper samples generated on three data sets using four methods, which also illustrates that PDP-EG and PGA-EG both have the advantage effect than GA and RSF. In addition, as can be seen from the figure that the value of *Difficulty* indicator on the dataset *OLIES2011* is far from 0.7. When we increase the penalty weight of the indicator *Difficulty* in methods, the indicator *Validity* will be greatly reduced, which indicates that under the small size question bank, in order to approach the expected difficulty of the exam paper, some skills may appear repeatedly in many questions, which leads to the decline of the exam paper *Validity* indicator.

The indicator *Validity* reflects the exam paper samples effectiveness by distance between the skill occurrence probability of the exam paper samples and the skill weight of the course. Fig. 7 shows the comparison of the indicator *Validity* for 100 exam paper samples generated on three data sets by using each method. As shown in Fig. 7, PGA-EG has significant advantages over other methods in datasets *ASSISTments0910* and *ASSISTments2015*. In dataset *OLIES2011*, the curves for indicator *Validity* for the four methods are similar, but Table 7 shows their subtle differences in the mean of the indicators, and PDP-EG has a slight advantage over other methods.

In order to better compare the effects of these methods, Fig. 8 shows the effect of the indicator *Difficulty* and *Validity* in an exam paper sample generated by four methods on three datasets. As shown in Fig. 8, all of the samples generated by PDP-EG and PGA-EG are concentrated at the target area. It is furthermore indicate that PDP-EG and PGA-EG have the better effect.

Table 5
Comparison of Standard Deviation of Z.

Standard Deviation	Average of $Dis_{P_HP_L}$	Average of maximum	Target difference
σ = 13	30.698	99.275	8.333
σ = 14	33.114	101.548	6.085
σ = **15**	35.440	103.814	**5.217**
σ = 16	37.843	105.999	6.110
σ = 17	40.118	108.149	8.226
σ = 18	42.537	110.557	11.134

Table 6
Comparisons of indicator *Difficulty*.

	GA		PDP-EG		PGA-EG		RSF	
	Mean	Std.D	Mean	Std.D	Mean	Std.D	Mean	Std.D
ASSISTments0910	0.721	0.006	**0.716**	0.004	0.718	0.004	0.722	0.009
ASSISTments2015	0.573	0.034	0.671	0.018	**0.689**	0.013	0.580	0.048
OLIES2011	0.391	0.041	**0.478**	0.036	0.475	0.027	0.393	0.041

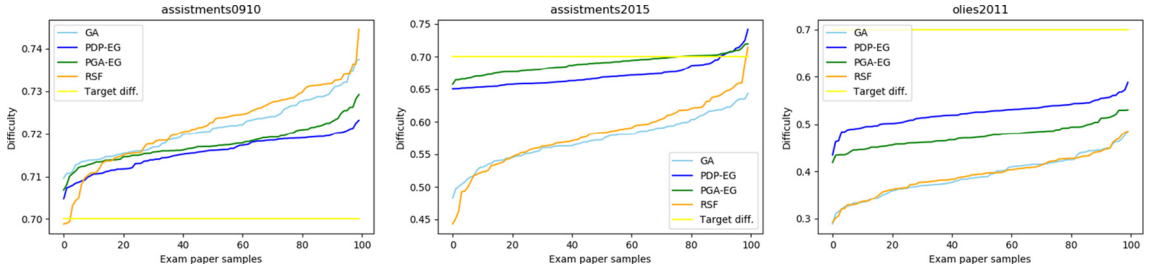

Fig. 6. Indicator *Difficulty* Comparison.

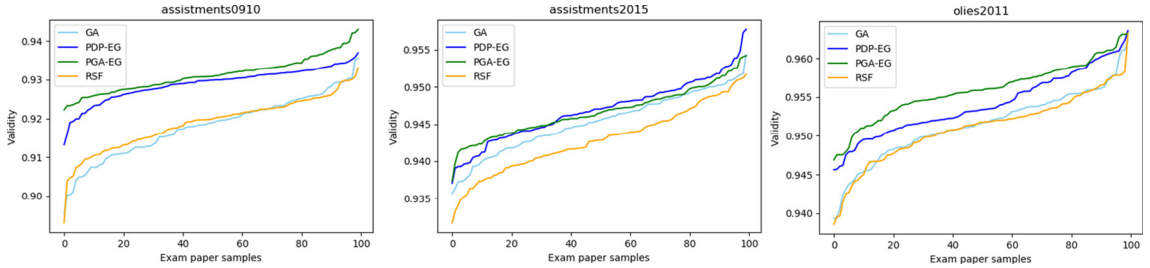

Fig. 7. Indicator *Validity* Comparison.

Experiments show that in the GA and RSF methods, although we replace the difficulty label with the skill answer correct rate, the subjective influence of the artificial setting difficulty is reduced to some extent, but it does not achieve the expected effect for a specific student group.

6.5.2. Score distribution

The *Rationality* index reflects the reasonableness of score distribution on exam paper samples. We also use Eq. (2) and Eq. (3) to predict the students' score on each exam paper sample. Fig. 9 shows the comparison of the indicator *Rationality* for 100 exam paper samples generated by using four methods on three datasets. As shown in Fig. 9, the PDP-EG and PGA-EG has significant advantage in datasets *ASSISTments0910* and *ASSISTments2015*.

In dataset *OLIES2011*, the curves for indicator *Rationality* for the four methods are similar. However, we can find that PGA-EG and PDP-EG are more advantageous from the minimum value of this indicator, In addition, the Table 8 shows their subtle differences in the mean and standard deviation that shows PGA-EG and PDP-EG have advantages over the GA and RSF.

Fig. 10 shows the exam paper sample with max value of *Rationality* which generated by all methods in dataset *ASSISTments0910*. As shown in Fig. 10, the distribution of score on exam paper samples generated by PDP-EG and PGA-EG are more consistent. In the distribution of score of the exam paper sample generated by GA and RSF, the segmentation at 30–40 was collapsed. This shows that we use the divergence between the preset random normal distribution and the predicted score

Table 7
Comparisons of indicator *Validity*

	GA		PDP-EG		PGA-EG		RSF	
	Mean	Std.D	Mean	Std.D	Mean	Std.D	Mean	Std.D
ASSISTments0910	0.918	0.008	0.929	0.004	**0.931**	0.004	0.919	0.007
ASSISTments2015	0.945	0.004	0.947	0.004	**0.947**	0.003	0.943	0.004
OLIES2011	0.951	0.004	**0.959**	0.005	0.956	0.004	0.951	0.004

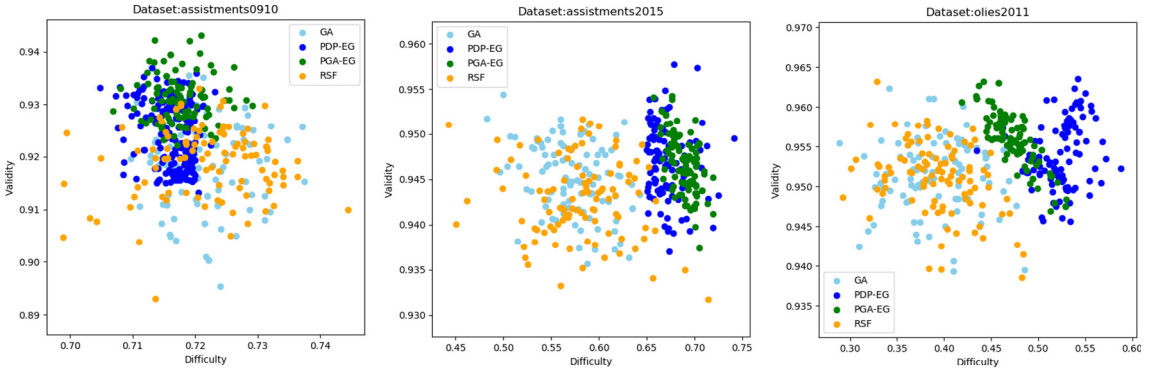

Fig. 8. *Difficulty* and *Validity* Comparison.

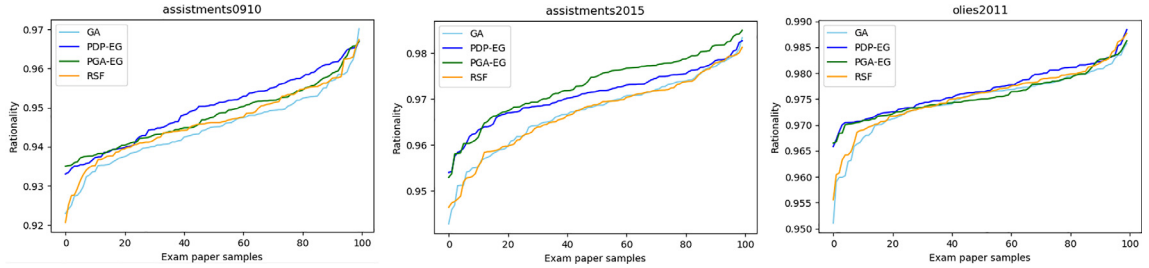

Fig. 9. Indicator *Rationality* Comparison.

distribution as the optimization condition of the EPG, which can achieve the purpose of getting the reasonable exam scores for a group students.

6.6. Experiment in a real teaching scenario

In order to experiment with real teaching scenarios, we provided online exercise for a real course and organize mock exams for this course. The name of the course is *Higher Education*, which is an optional course for undergraduate and graduate students in education. The learning system of this course is available on the site:http://gdsz.zymreal.com/resource/. The course includes 70 skills and has 494 questions in its exercise bank (including 406 Multiple-Choice questions and 88 True-or-False questions). More than 1,000 students accessed this course online by using their mobile phones or PCs. The learning system records the results of each student exercise answers.

Students in the two classes are instructed to take a mock examination after studying and exercising online, and each class consisted of 20 students. The questions in the mock exam papers used by the two classes are extracted from two different question sets. The first set is an exercise bank, and the second set is a mixed question bank with exercise and 150 new questions.

We export the data of 1,313 students in the learning system and process it into a sequence dataset, and then use this dataset to train the DKT model (AUC = 0.78). This DKT model is used to predict the performance of students in two classes.

Then, we use PDP-EG and RSF to generate two mock exam papers for class A based on the mixed question bank, and used PGA-EG and GA to generate two mock exam papers for class B based on the exercise bank. Each mock exam paper has 50 questions, and each question has a score of 2. The target of difficulty is 0.7. The target of distribution of exam scores is

Table 8
Comparisons of indicator *Rationality*

	GA		PDP-EG		PGA-EG		RSF	
	Mean	Std.D	Mean	Std.D	Mean	Std.D	Mean	Std.D
ASSISTments0910	0.945	0.010	**0.950**	0.009	0.949	0.008	0.944	0.010
ASSISTments2015	0.968	0.008	0.971	0.007	**0.973**	0.006	0.967	0.008
OLIES2011	0.971	0.006	**0.976**	0.004	0.975	0.004	0.970	0.006

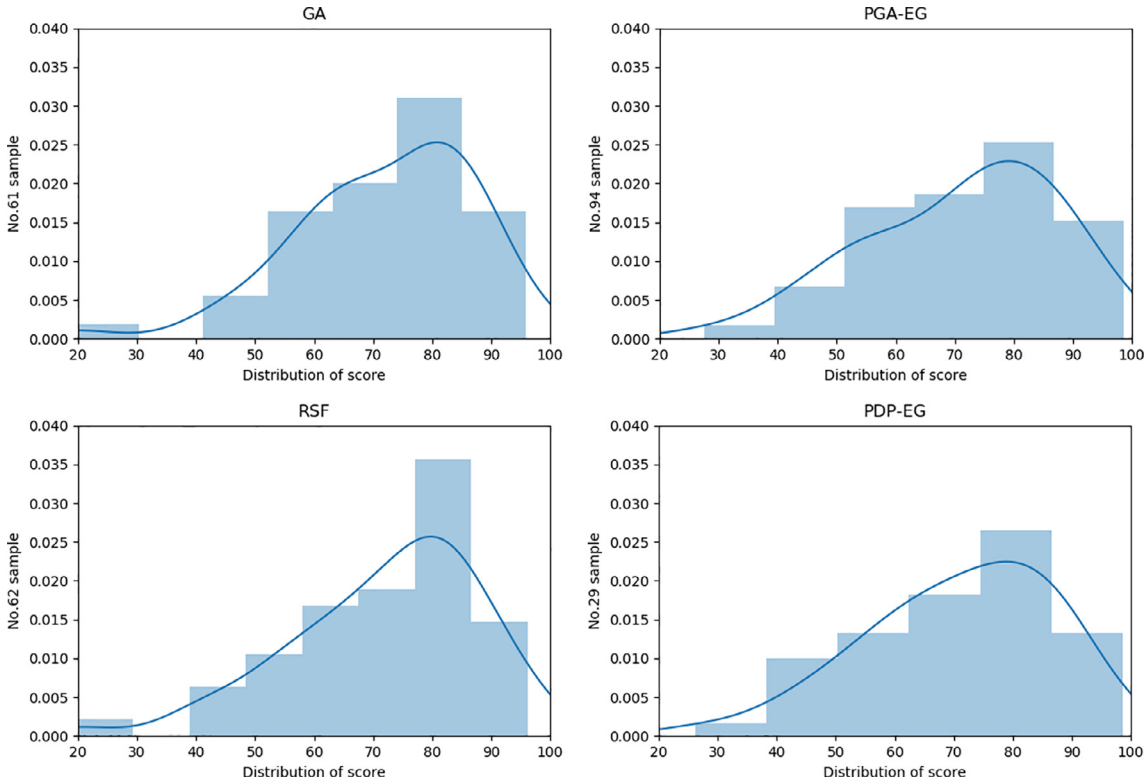

Fig. 10. Distribution of score on exam paper samples.

the parameter for PDP-EG and PGA-EG methods, which satisfy normal distribution $N\left(70, 15^2\right)$, that is, the mean is 70 and the standard deviation is 15.

The exam paper evaluation indicators reflected in the exam results of Class A and Class B are shown in Tables 9 and 10. From the results we can see that our approach has advantages over the baselines. The indicator Difficulty reflects the average score of the students in the exam. Obviously, Class A gets lower scores than Class B. The reason is that new questions are added to the exam paper used by Class A, which makes the exam paper more difficult. However, PDP-EG and PGA-EG show good results and bring the indicator Difficulty close to the required difficulty, and the other two indicators are also better than the baseline methods.

7. Conclusions and future work

As an essential part of teaching, exam paper generation has to face the challenge of manual labeling. In this study, we propose a novel EPG approach, which applies the DKT model to predict learning performance, and use dynamic programming and genetic algorithm to optimize the quality of exam paper. The achieved AUC scores of the DKT model are not the same for different datasets, which measures the accuracy difference in the level of prediction of students' skill mastery level in practice. With the higher accuracy of the students performance prediction our approach can better ensure the difficulty of the exam paper and reasonable distribution of student achieved scores in the exam. Our approach is summarized as follows. First, omitting pre-setting the difficulty level of each question in the question bank, PDP-EG and PGA-EG can effectively overcome the limitations of the method of manually labeling exam paper parameters. Second, our approach ensure that the distribution of the skill weights of an exam paper generated could be close to that of the corresponding course. Third, our approach can generate exam papers for different groups of students to ensure a relatively reasonable distribution

Table 9
Result of the experiments on Class A.

EPG Method	PDP-EG	RSF
Difficulty	**0.704**	0.645
Validity	**0.952**	0.939
Rationality	**0.970**	0.958

Table 10
Result of the experiments on Class B.

EPG Method	PGA-EG	GA
Difficulty	**0.709**	0.735
Validity	**0.958**	0.931
Rationality	**0.959**	0.945

of exam scores. Our future work will focus on skill data mining, and use deep learning to improve methods to solve related educational knowledge mining problems. Of course, our approach still allows space for improvement. The first direction is that we use random sampling in this study, and there should be other sampling algorithms can replace it to improve performance. Another direction for future work is that we will try to develop this approach into a deep learning-based generation model, to improve the coupling and performance of the EPG.

CRediT authorship contribution statement

Zhengyang Wu: Conceptualization, Methodology, Writing - original draft. **Tao He:** Software, Investigation, Visualization. **Chenjie Mao:** Data curation, Validation. **Changqin Huang:** Supervision, Formal analysis, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (No. U1811263, 61877020), and the Foundation of China Scholarship Council (No. 201808440652), and the Key-Area Research and Development Program of Guangdong Province, China (No. 2018B010109002).

References

[1] Charu C Aggarwal, S. Yu Philip, An effective and efficient algorithm for high-dimensional outlier detection, The VLDB Journal 14 (2) (2005) 211–221.
[2] Nabeela Altrabsheh, Mihaela Cocea, and Sanaz Fallahkhair. Predicting students' emotions using machine learning techniques. In Proceedings of 17th International Conference on Artificial Intelligence in Education, pages 537–540, 2015..
[3] Gema Bello-Orgaz, Sancho Salcedo-Sanz, David Camacho, A multi-objective genetic algorithm for overlapping community detection based on edge encoding, Information Sciences 462 (1) (2018) 290–314.
[4] Abi Brooker, Linda Corrin, Paula De Barba, Jason Lodge, Gregor Kennedy, A tale of two moocs: How student motivation and participation predict learning outcomes in different moocs, Australasian Journal of Educational Technology 34 (1) (2018) 73–87.
[5] Claudio Carpineto, Renato De Mori, Giovanni Romano, Brigitte Bigi, An information-theoretic approach to automatic query expansion, ACM Transactions on Information Systems 19 (1) (2001) 1–27.
[6] Dhawaleswar Rao Ch and Sujan Kumar Saha. Automatic multiple choice question generation from text: A survey. IEEE Transactions on Learning Technologies, pages 1–13, 2018..
[7] Aishwarya Chavan, Mojitha Mohandas, Rasika Manjarekar, Divya Karekar, Supriya Mandhare, Automated question paper generator system using apriori algorithm and fuzzy logic, International Journal for Innovative Research in Science & Technology 2 (11) (2016) 707–710.
[8] Xiang Cheng, Shuguang Zhu, Su. Sen, Gang Chen, A multi-objective optimization approach for question routing in community question answering services, IEEE Transactions on Knowledge and Data Engineering 29 (9) (2017) 1779–1792.
[9] Albert T Corbett, John R Anderson, Knowledge tracing: Modeling the acquisition of procedural knowledge, User modeling and user-adapted interaction 4 (4) (1994) 253–278.
[10] Linda Crocker and James, Algina. Introduction to classical and modern test theory, ERIC (1986).
[11] P.G. De Barba, Gregor E Kennedy, M.D. Ainley, The role of students' motivation and participation in predicting performance in a mooc, Journal of Computer Assisted Learning 32 (3) (2016) 218–231.
[12] Sahar Abd El-Rahman and Ali Hussein Zolait, Automated test paper generation using utility based agent and shuffling algorithm, International Journal of Web-Based Learning and Teaching Technologies 14 (1) (2019) 69–83.
[13] Lanting Fang, Luu Anh Tuan, Siu Cheung Hui, Lenan Wu, Personalized question recommendation for english grammar learning, Expert Systems 35 (2) (2018) 35–50.
[14] Kenneth D Hopkins, Educational and psychological measurement and evaluation, ERIC (1998).
[15] Yu. Zhenya Huang, Enhong Chen Yin, Yu.Su. Hui Xiong, Hu. Guoping, et al, Ekt: Exercise-aware knowledge tracing for student performance prediction, IEEE Transactions on Knowledge and Data Engineering (2019).
[16] Suraj Kamya, Madhuri Sachdeva, Navdeep Dhaliwal, Sonit Singh, Fuzzy logic based intelligent question paper generator, in: In Proceedings of 4th IEEE International Advance Computing Conference, 2014, pp. 1179–1183.
[17] Derong Liu, Ding Wang, Xiong Yang, An iterative adaptive dynamic programming algorithm for optimal control of unknown discrete-time nonlinear systems with constrained inputs, Information Sciences 220 (1) (2013) 331–342.
[18] Xinping Liu, Yunliang Zhang, Introduction to educational statistics and evaluation, Science Press, Beijing, 2013.
[19] Kritphong Mongkhonvanit, Klint Kanopka, and David Lang. Deep knowledge tracing and engagement with moocs. In Proceedings of 9th International Conference on Learning Analytics & Knowledge, pages 340–342, 2019..
[20] Kapil Naik, Shreyas Sule, Shruti Jadhav, Surya Pandey, Automatic question paper generation system using randomization algorithm, International Journal of Engineering and Technical Research 2 (12) (2014) 192–194.

[21] N. Efendi, Nasibov and A Övgü Kinay. An iterative approach for estimation of student performances based on linguistic evaluations, Information Sciences 179 (5) (2009) 688–698.
[22] Minh Luan Nguyen, Siu Cheung Hui, and ALvis CM Fong. An efficient multi-objective optimization approach for online test paper generation. In Proceedings of IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making, pages 182–189, 2011..
[23] N.T.Heffernan. Assistments. Website, 2014. Accessed Nov. 29, 2019..
[24] SS. Dandge, PV. Bobade, M.A.M.A. Pund, A study of different algorithm for automatic generation of question paper, International Science and Technology Journal 7 (5) (2018) 27–32.
[25] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. Deep knowledge tracing. In Proceedings of the Advances in Neural Information Processing Systems, pages 505–513, 2015..
[26] Ramkumar Rajendran, Anurag Kumar, Kelly E Carter, Daniel T Levin, and Gautam Biswas. Predicting learning by analyzing eye-gaze data of reading behavior. In Proceedings of 11th International Conference on Educational Data Mining, pages 455–461, 2018..
[27] Meghna Saikia, Saini Chakraborty, Suranjan Barman, and Sarat Kr Chettri. Aptitude question paper generator and answer verification system. In Recent Developments in Machine Learning and Data Analytics, Springer, 2019, pp. 129–136.
[28] Nina Stancheva Stancheva, Ivan Popchev, Asya Stoyanova-Doycheva, and Stanimir Stoyanov. Automatic generation of test questions by software agents using ontologies. In Proceedings of 8th IEEE International Conference on Intelligent Systems, pages 741–746, 2016..
[29] Paul Steif. Oli engineering statics. Website, 2011. Accessed Nov. 29, 2019..
[30] Su. Yu, Qingwen Liu, Qi Liu, Yu. Zhenya Huang, Enhong Chen Yin, Chris Ding, Si Wei, Guoping Hu, Exercise-enhanced sequential modeling for student performance prediction, in: In Proceedings of 32nd AAAI Conference on Artificial Intelligence, 2018, pp. 2435–2443.
[31] Abu Bakar Md Sultan, Shuffling algorithms for automatic generator question paper system, Computer and Information Science 3 (2) (2010) 244–248.
[32] Thomas James Tiam-Lee and Kaoru Sumi. Analysis and prediction of student emotions while doing programming exercises. In Proceedings of 15th International Conference on Intelligent Tutoring Systems, pages 24–33, 2019..
[33] Lisa Wang, Angela Sy, Larry Liu, and Chris Piech. Deep knowledge tracing on programming exercises. In Proceedings of 4th ACM Conference on Learning at Scale, pages 201–204, 2017..
[34] Kai Xiong and Xiaojun Huang. Research on auto-generating test paper system based on lda and genetic algorithm. In Proceedings of 9th IEEE International Conference on Software Engineering and Service Science, pages 416–421, 2018..
[35] Chun-Kit Yeung and Dit-Yan Yeung. Incorporating features learned by an enhanced deep knowledge tracing model for stem/non-stem job prediction. International Journal of Artificial Intelligence in Education, 29(3):317–244, 2019..
[36] Mehmet Yildirim, A genetic algorithm for generating test from a question bank, Computer Applications in Engineering Education 18 (2) (2010) 298–305.
[37] Xiaomei Yu, Daibin Wei, Qian Chu, and Hong Wang. The personalized recommendation algorithms in educational application. In Proceedings of 9th IEEE International Conference on Information Technology in Medicine and Education, pages 664–668, 2018..
[38] Wenjie Yuan, Chengji Deng, Hongxi Zhu, Jun Li, The statistical analysis and evaluation of examination results of materials research methods course, Creative Education 3 (7) (2012) 162–164.
[39] Kai Zhang, Yiyu Yao, A three learning states bayesian knowledge tracing model, Knowledge-Based Systems 148 (1) (2018) 189–201.
[40] Yuwen Zhou, Changqin Huang, Hu. Qintai, Jia Zhu, Yong Tang, Personalized learning full-path recommendation model based on lstm neural networks, Information Sciences 444 (1) (2018) 135–152.