J. COMBIN. MATH. COMBIN. COMPUT. 127a (2025) 5357--5377

Journal of Combinatorial Mathematics and Combinatorial Computing
www.combinatorialpress.com/jcmcc

A study on the shortest path algorithm for vocal skill mastery path in practical teaching innovation

Yan Liu[1,✉]

[1] *School of Music, Henan Polytechnic University, Jiaozuo, Henan, 454003, China*

ABSTRACT

How to form a personalized shortest learning path for vocal skills based on learners' individual characteristics is the key to improve the efficiency of vocal music teaching. In this paper, on the basis of dynamic key-value memory network, a gating mechanism is used to update students' knowledge mastery status, and a knowledge tracking model based on dynamic key-value gated recurrent network is proposed to realize the accurate assessment of students' vocal music level. On this basis, after searching the suboptimal path using the particle swarm algorithm, the shortest path is searched using the ant colony algorithm, which solves the shortcoming of the blindness of the initial search direction of the single ant colony algorithm, and constructs a recommendation model for optimization of the learning path of vocal skills. The results of simulation experiments show that the model AUC and ACC on the ASSIST2015 dataset are 0.7468 and 0.7654, respectively, which are much higher than the highest 0.7281 and 0.7528 in the baseline model. Path optimization was achieved for both ordinary and excellent vocal students, and the average optimization was 4.297 and 3.242 on ASSIST2009, and 3.819 and 3.044 on ASSIST2015.This paper makes an innovative exploration to improve the quality of vocal music teaching.

Keywords: key-value memory network; gating mechanism; ant colony algorithm; vocal music teaching

1. Introduction

Vocal music teaching is one of the most important aspects of cultivating students' musical literacy and performance ability [1]. The cultivation of basic skills in vocal music teaching is the cornerstone of students' growth and development, and is the key for students to build a good foundation in vocal music [2]. Good foundation skills can not only lay the foundation for students' future in-depth study and singing, but also improve their performance ability and musical literacy [3-4]. However, due to the differences and individual differences of students, teachers need to flexibly use a variety of methods to provide targeted guidance for different students' characteristics [5]. For students learning vocal music, they need to master a variety of skills including breathing training, occurrence exercises,

✉ Corresponding author.
E-mail address: liuchang830302@163.com (Y. Liu).
Received 03 February 2024; Revised 12 April 2024; Accepted 10 November 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-301

including practical learning methods [6-7]. This is not only related to the students' own professional development, but also will have a direct impact on whether the discipline teaching education can achieve comprehensive development [8-9]. Therefore, it is crucial to investigate an effective strategy that can improve the teaching of specialized music skills to students in higher education.

The goal of the shortest path problem is to find a path that minimizes the sum of total weights between two given nodes in a network [10]. As one of the most classical and common problems in graph theory, the shortest path problem is the theoretical basis for research in computer science, geographic information science, operations research, management science, transportation engineering, industrial engineering and complex systems science, and its core idea is closely related to the fields of education and management in daily life [11-14]. Combining it with the teaching of vocal skills in music education helps students find the shortest path to skill mastery and brings an innovative teaching approach to the practical teaching of the course [15-17].

In this paper, based on the knowledge tracking of vocal level determination, we design the optimization recommendation model mentioning the learning path optimization of vocal skills based on ant colony-particle swarm hybrid algorithm. Firstly, a knowledge tracking model DKVGRU based on dynamic key-value gated recurrent network is produced. By retaining the external key-value memory module and memory read-write mechanism of the DKVMN model, the DKVGRU model not only digs out the relationship between exercises and individual knowledge points, but also reads and writes knowledge mastery status of individual knowledge points. Then based on the ant colony-particle swarm hybrid algorithm, the particle swarm algorithm is utilized in the early stage to carry out a coarse search with its rapidity and global nature, and after a certain number of iterations to find a suboptimal solution to the problem, and then the pheromone matrix of the ant colony algorithm is initially distributed with the obtained suboptimal solution to overcome the defect of the ant colony algorithm's searching having a blindness, and thus to determine the shortest path of the vocals skill mastery path The shortest path for vocal skill mastery is determined by overcoming the blindness of the ACO search. Finally, simulation experiments are carried out on two datasets, ASSIST2009 and ASSIST2015, to verify the model performance.

2. Measurement of students' vocal level based on DKVGRU knowledge tracking

2.1. Definition of basic concepts and symbols

Commonly used concepts in the knowledge tracking model are exercise interaction, knowledge mastery state and knowledge growth, and their concepts are explained as follows:

1) Exercise interaction: Exercise interaction refers to the process in which the online learning platform judges whether the answers to the exercises submitted by students are correct or not. Therefore, the exercise interaction data consists of two parts: the exercise label and the student's answer, i.e., $x_t = (q_t, a_t)$.

2) Knowledge mastery state: students' mastery of each knowledge point is collectively referred to as the knowledge mastery state, which is a dynamically changing value in the model of this paper.

3) Knowledge growth: the new knowledge acquired by students in one answer is called knowledge growth. The model obtains knowledge growth information through the exercise interaction data.

4) Knowledge update: Through a question-answering activity, the change of students' own knowledge mastery state is called knowledge update, and knowledge update includes two parts: knowledge forgetting and knowledge growth.

The DKVGRU model stores N knowledge point of the exercise set in the static knowledge matrix $K \in \mathbb{R}^{N \times d_k}$, d_k represents the dimension of the knowledge point vector, and the knowledge vector k_i in the knowledge matrix $K = (k_1, k_2, \dots, k_N)$ represents the knowledge point i. The knowledge mastery states of the N knowledge points are stored in the dynamic knowledge mastery state matrix $V \in \mathbb{R}^{N \times d_v}$, d_v represents the dimension of the knowledge point mastery state vector, and the knowledge mastery vector v_i in the knowledge mastery state matrix $V_t = (v_1, v_2, \ldots, v_N)$ represents the mastery degree of the corresponding knowledge point i.

2.2. Knowledge tracking model based on dynamic key-value gated recurrent networks

The DKVGRU model is designed to gradually update students' knowledge mastery state based on their historical sequences of exercise interactions, and to predict the probability of students' correctly answering exercises through their knowledge mastery state. Among them, the construction of the DKVGRU model is divided into the answer prediction network and the knowledge update network based on the gating mechanism. The answer prediction network is used to predict the correctness of students' answers to Exercise q_t, which consists of three parts: the knowledge weight of the exercise, the mastery level of the exercise, and the prediction of the answer to the exercise: 1) The calculation of the knowledge weight of the exercise is to obtain the relevance degree of Exercise q_t to various knowledge points. 2) The calculation of the degree of knowledge mastery of the exercises is a combination of the degree of mastery of the knowledge contained in Exercise q_t and the difficulty of the exercises, to obtain the overall degree of mastery of Exercise q_t by the students. 3) Exercise Answer Prediction predicts the probability of students answering Exercise q_t correctly based on their overall mastery of Exercise q_t. The knowledge update network based on the gating mechanism updates the students' knowledge mastery status through the exercise interaction data $x_t = (q_t, a_t)$ to get the students' new knowledge mastery status [18].

2.2.1. Knowledge weighting of exercises. For predicting the probability of a student answering Exercise q_t correctly, the model first calculates the correlation between Exercise q_t and each knowledge point. Exercise q_t is mapped into exercise vector $e_t \in \mathbb{R}^{d_k}$ by Embedding matrix $A \in \mathbb{R}^{E \times d_k}$. Exercise vector e_t is inner-producted with each knowledge point vector k_i in knowledge matrix $K = (k_1, k_2, \ldots, k_N)$ for similarity calculation. Then, softmax function normalizes the similarity between the exercises and all the knowledge points to obtain the weight $w_t \in \mathbb{R}^N$ of exercise q_t with each knowledge point, i.e.:

$$w_t = softmax\left(e_t \cdot K^T\right) \tag{1}$$

where, $softmax(z_i) = e^{z_i} \Big/ \sum_j e^{z_j}$. Each element of the knowledge point weight vector w_t, represents the degree of relevance of Exercise q_t to each knowledge point.

2.2.2. Mastery of exercises. The knowledge weights w_t of Exercise q_t are weighted and summed for each knowledge point in the students' knowledge mastery state $V_t = (v_1, v_2, \ldots, v_N)$ of each knowledge point v_i to obtain the students' overall mastery level $r_t \in \mathbb{R}^{d_v}$ of the knowledge points contained in Exercise q_t, i.e:

$$r_t = w_t \cdot V_t \tag{2}$$

In addition to the students' mastery of the knowledge points contained in the exercises will affect the correct rate of answering the questions, the difficulty of the exercises themselves will also affect the correct rate of answering the questions. The model takes the exercise vector e_t, which represents the information of the exercise itself, and goes through the fully connected layer and activation function to get the exercise difficulty vector $d_t \in \mathbb{R}^{d_k}$, which represents the information of the difficulty of the exercise, i.e.:

$$d_t = \tanh\left(e_t \cdot W_1 + b_1\right) \tag{3}$$

Among them, the activation function formula is $\tanh(z_i) = \left(e^{z_i} - e^{-z_i}\right) \Big/ \left(e^{z_i} + e^{-z_i}\right)$, $W_1 \in \mathbb{R}^{d_k \times d_k}$ and $b_1 \in \mathbb{R}^{d_k}$ are the weights and bias terms of the fully connected layer, respectively.

Taking into account the overall mastery level of students on the exercises and the difficulty of the exercises, two factors that affect the correctness of students' answers, the model splices the vectors of r_t and d_t, through the fully connected layer and the activation function, to get the vector of overall mastery level of students on the exercises, $f_t \in \mathbb{R}^{d_k}$, that is:

$$f_t = \tanh\left(\left[r_t; d_t\right] \cdot W_2 + b_2\right) \tag{4}$$

2.2.3. Exercise Answer Prediction. The student's overall mastery of the exercises vector f_t is passed through the fully connected layer and the activation function, respectively, to obtain the student's predicted performance of doing the exercises p_t, i.e:

$$p_t = sigmoid\left(f_t \cdot W_3 + b_3\right) \tag{5}$$

where the activation function is given by $sigmoid(z) = 1/\left(1+e^{-z}\right)$, p_t represents the predicted probability of a student answering Exercise q_t correctly.

2.2.4. Knowledge updating based on gating mechanisms. Utilizing the external memory module's readable and writable knowledge mastery status of individual knowledge points, the DKVGRU model uses the knowledge application control gate to calculate the weights of students' actual application of knowledge points and the knowledge forgetting control gate to calculate the weights of students' forgetting of successively learned knowledge, respectively [19].

The DKVGRU model updates the student's knowledge mastery state matrix based on the Exercise Interaction Data $\left(q_t, a_t\right)$. V_t. The Exercise Interaction Data $\left(q_t, a_t\right)$ is used to obtain the Exercise Interaction Information y_t after the student answers Exercise q_t through Eq. (6). The Exercise Interaction Information y_t is mapped into an Exercise Interaction Vector $c_t \in \mathbb{R}^{d_v}$ through the Embedding Matrix $B \in \mathbb{R}^{E \times d_v}$:

$$y_t = q_t + a_t * E \tag{6}$$

In the actual answering of the questions, students will apply the skillful knowledge points to answer the exercise questions.The DKVGRU model adds the interaction vector c_t of the exercise questions with the knowledge mastery state vector v_i of each knowledge point in the knowledge mastery state matrix $V_t = \left(v_1, v_2, \ldots, v_N\right)$. The result of adding this vector represents the extent to which the student actually applies each knowledge point to answer the question according to his/her own knowledge mastery state, and then go through the fully connected layer and activation function to obtain the exercise knowledge application control gate $Z_t \in \mathbb{R}^{N \times d_v}$, i.e.:

$$C_t = Concat\left(c_t, c_t, \ldots, c_t\right) \tag{7}$$

$$Z_t = sigmoid\left(\left[V_t + C_t\right] \cdot W_z + b_z\right) \tag{8}$$

where the exercise interaction matrix $C_t \in \mathbb{R}^{N \times d_v}$ is obtained by splicing N identical exercise interaction vectors c_t. The knowledge application control gate Z_t represents the weight of each knowledge point applied by the student to answer the exercise q_t according to his/her knowledge mastery status.

The DKVGRU model uses the knowledge application control gate Z_t to weight the student's knowledge mastery state matrix V_t to obtain the applied knowledge state $D_t \in \mathbb{R}^{N \times d_v}$. The exercise interaction matrix C_t is then spliced with the applied knowledge state D_t to obtain the student's knowledge growth matrix $\tilde{V}_t \in \mathbb{R}^{N \times d_v}$ after answering this question, i.e.:

$$D_t = Z_t * V_t \tag{9}$$

$$\tilde{V}_t = \tanh\left(\left[D_t; C_t\right] \cdot W_r + b_r\right) \tag{10}$$

Where the weight of the fully connected layer is $W_r \in \mathbb{R}^{2d_v \times d_v}$ and the bias term is $b_r \in \mathbb{R}^{d_v}$.

While students learn knowledge, they also forget knowledge, and the degree of forgetting of students' successively mastered knowledge is different.The DKVGRU model adds up the exercise interaction vector c_t and each knowledge point state v_i in the knowledge mastery state matrix $V_t = (v_1, v_2, \ldots, v_N)$, and then obtains the knowledge forgetting control gate $U_t \in \mathbb{R}^{N \times d_v}$ through the fully connected layer and the activation function, that is:

$$U_t = sigmoid\left(\left[V_t + C_t\right] \cdot W_u + b_u\right) \tag{11}$$

Knowledge forgetting control gate U_t can distinguish the degree of students' forgetting of successively learned knowledge.DKVGRU model updates students' knowledge mastery state through knowledge forgetting control gate U_t to get students' new knowledge mastery state V_{t+1}, i.e.:

$$V_{t+1} = (1 - U_t) * V_t + U_t * \tilde{V}_t \tag{12}$$

3. Vocal skill learning path recommendation based on ant colony-particle swarm hybrid algorithm

3.1. Mathematical model of the personalized learning path recommendation problem for vocal music

The problem of personalized learning path recommendation for vocal music focuses on learners' individual attribute characteristics and learning object characteristics, and provides learners with a sequence of learning objects that match their individual learning abilities through the analysis of differences between learners and learning objects. Learner attribute characteristics include cognitive level, desired goal and learning style, etc. Learning object characteristics include the difficulty coefficient of the learning object, the knowledge points covered and the constraint relationship between the objects. Vocal personalized learning path recommendation is a learning object recommendation and path planning problem modeling, the problem model is a formulaic expression of matching learner characteristics and learning object characteristics.

3.1.1. Construction of the learning object model. Let $K = \{k_1, k_2, \cdots, k_M\}$, denote the set of M knowledge points that learners need to learn; $R = \{r_1, r_2, \cdots, r_N\}$ is the set of N learning objects in the course; let v_{mn} denote the correlation coefficient between the n th learning object r_n and the m th knowledge point k_n, $0 \le v_m \le 1$; g_{ij} denote the order of precedence between r; and r, as in equation (13).

$$g_{ij} = \begin{cases} 1 & \text{Learn } r_i \text{ before you learn } r_j \\ A & \text{Learning } r_j \text{ before learning } r_i \end{cases} \tag{13}$$

Where A is the penalty parameter.

Difficulty coefficient $D = \{d_1, d_2, \cdots, d_N\}$ is used to weigh the difficulty level of the learning object, $0 \le d_n \le 1$, d_n is closer to 0 means that the learning object is more suitable for beginners, and closer to 1 means that the learning object is more suitable for experts.

The types of learning objects are categorized into four types: text, picture, audio/video, and interactive software, and many learning objects adopt multiple modes of expression at the same time, so the vector $Q_n = \{q_{n1}, q_{n2}, q_{n3}, q_{n1}\}$ is used to describe the degree to which the learning object r_n adopts the various modes of knowledge expression, $0 \le q_{n1}, q_{n2}, q_{n3}, q_{n6} \le 1$, and $\sum_{j=1}^{4} q_{nj} = 1$.

3.1.2. Construction of Learner Characterization Models. Define the learner U, and express the learner characteristic model as: $U = \{C, S\}$. Where, $C = \{c_1, c_2, \cdots, c_N\}$ indicates the learner U's cognitive level of the N learning objects in the course, whose specific value is calculated by the knowledge tracking model based on DKVGRU designed in the previous section, $0 \le c_N \le 1$, c_n the closer to 0 indicates that the learner's level is closer to that of an expert.

The learning process is calculated based on the learner's pre-test, learning behavior data and other aspects, 6 and 7 the closer to 0 indicates that the learner's cognitive level is closer to that of a beginner, and the closer to 1 indicates that the learner's level is closer to that of an expert.

$S = \{s_1, s_2, s_3, s_4\}$ indicates the learning style of the learner U. From Kolb's learning style types, we know that there are four types of learning styles: dispersive, convergent, assimilative and moderative, and different learning styles will have an impact on the selection of learning objects. Learners with dispersive styles prefer learning objects rich in vivid symbols such as diagrams, tables, animations, etc.; those with convergent styles prefer text-based learning objects; those with assimilative styles prefer learning objects rich in language explanations such as audio and video; and those with assimilative styles prefer learning objects rich in language explanations such as audio and video. Assimilative learners prefer audio, video, and other objects rich in verbal explanations, while moderated learners prefer to learn through experience, such as manipulating simulation software. The same learner will show a tendency to exhibit multiple learning types, s_1, s_2, s_3, s_4 indicating the extent to which the learner U tends to belong to each of the four learning styles, $0 \le s_1, s_2, s_3, s_4 \le 1$ and $\sum_{j=1}^{4} s_j = 1$ respectively.

3.1.3. Construction of the objective function. The learning path recommendation problem can be regarded as a single-objective optimization problem transformed from multi-objective, the objectives include: whether the difficulty of the learning object matches the cognitive level of the learner, whether the type of the learning object matches the learning style of the learner, whether the order of the learning objects on the learning path is reasonable, etc., and ultimately, we find the optimal path to enable the learner to complete the learning of all the learning objects on the learning path.

Define the learning path selection matrix $X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1N} \\ x_{21} & x_{22} & \cdots & x_{2N} \\ \vdots & \vdots & \vdots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NN} \end{bmatrix}$, $x_{ij} = 1$ if the learning path is learning object r_i to learning object r_j; otherwise $x_{ij} = 0$.

1) The difference between the cognitive level of the learner and the difficulty of the learning object on the learning path x_{ij} as in equation (14).

$$f_1(x) = x_{ij}\left(\left|d_i - c_i\right| + \left|d_j - c_j\right|\right) \tag{14}$$

2) The relevance of the learning object to the knowledge point on learning path x_{ij}, as in equation (15).

$$f_2(x) = x_{ij}\left(\sum_{p=1}^{M} v_{ip} + \sum_{p=1}^{M} v_{jp}\right) \tag{15}$$

3) Learning cost on learning path x_{ij}, as in equation (16).

$$f_3(x) = x_{ij} g_{ij} \tag{16}$$

4) Difference between learner's learning style and the type of learning object on learning path x_{ij} as in equation (17).

$$f_4(x) = x_{ij} \times \left(\sqrt{\sum_{p=1}^{4} \left(s_p - q_{ip} \right)^2} + \sqrt{\sum_{p=1}^{4} \left(s_p - q_{jp} \right)^2} \right) \tag{17}$$

The four functions constructed above are assigned with corresponding weight values, and the objective function of the learning path is constructed using the linear weighting method, as in equation (18).

$$\min F(x) = \omega_1 f_1(x) + \omega_2 \log_a f_2(x) + \omega_3 f_3(x) + \omega_4 f_4(x) \tag{18}$$

where $\omega_1 + \omega_2 + \omega_3 + \omega_4 = 1$; $0 < a < 1$.

3.2. Ant colony particle swarm hybrid algorithm design

Ant Colony Algorithm (ACO) has strong blindness and slow search speed in the early stage of the algorithm, while Particle Swarm (PSO) algorithm has strong parallel search capability and faster search speed. The idea of Ant Colony Particle Swarm hybrid algorithm is to utilize the fast and global nature of Particle Swarm algorithm to conduct coarse search in the early stage of the algorithm, and find the suboptimal solution of the problem after a certain number of iterations, and then use the suboptimal solution to distribute the pheromone matrix of the Ant Colony algorithm initially, so as to overcome the defects of Ant Colony algorithm's blindness in searching, and to make the searching space smaller, so as to find the optimal solution of the problem. The flowchart of the hybrid algorithm is shown in Fig. 1.

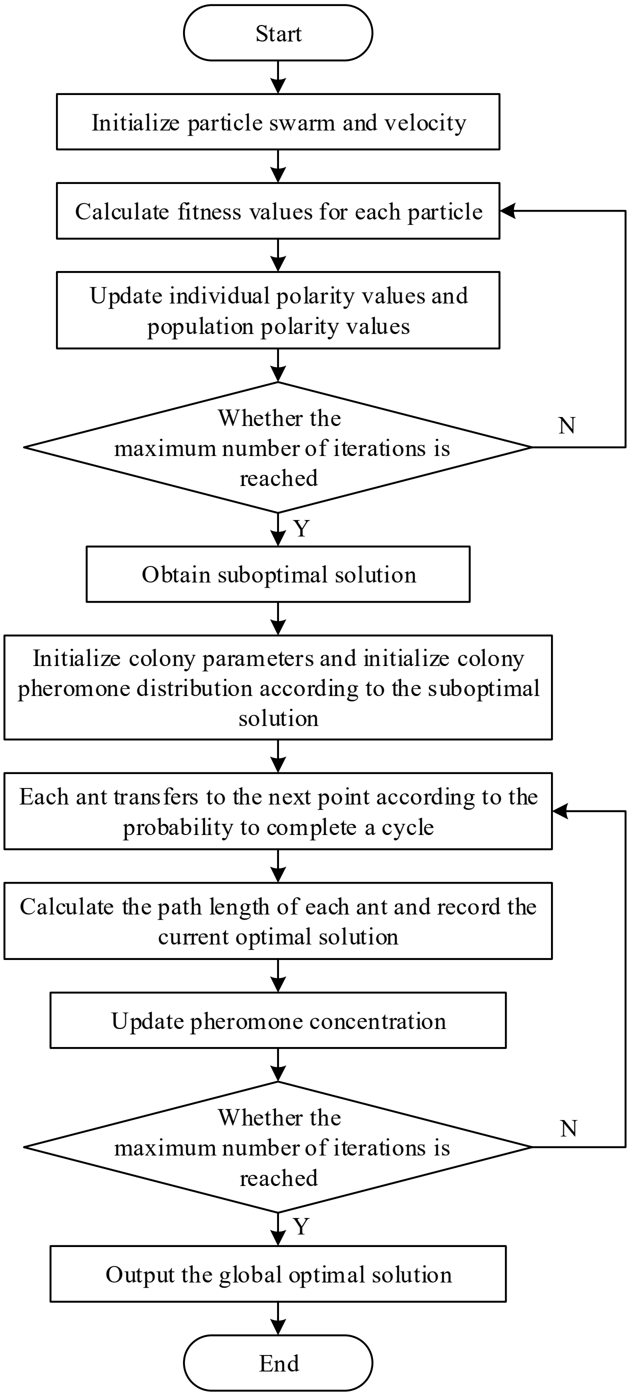


Fig. 1. Ant swarm hybrid algorithm process

The main parametric variables in the ACO particle swarm algorithm are the fitness function, the velocity and position update formulas for each particle, the heuristic information, the pheromone, and the selection probability of the path.

1) Fitness function. The objective function of the learning path is taken as the fitness function.

2) Velocity and position update formula. In each iteration, the particle i at time $(t+1)$ velocity and position are updated as in Eq. (19), Eq. (20).

$$v_{i,j}(t+1)=\omega v_{i,j}(t)+c_1 r_1\left[p_{i,j}-x_{i,j}(t)\right] \\ +c_2 r_2\left[p_{k,j}-x_{i,j}(t)\right] \tag{19}$$

$$x_{i,j}(t+1)=x_{i,j}(t)+v_{i,j}(t+1), j=1,\cdots,d \tag{20}$$

Where $v_{i,j}(t)$ is the current velocity of particle i in the j rd dimension; $x_{i,j}(t)$ is the current position of particle i in the j th dimension; $p_{i,j}$ is the individual extreme value; $p_{g,j}$ is the global extreme value; c_1, c_2 are acceleration constants; r_1, r_2 are random numbers uniformly distributed between 0 and 1; ω is the inertia weight.

3) Heuristic information. The objective function of the learning path is used as the heuristic information [20].

4) Pheromone. Initialize the pheromone τ_{ij} according to the suboptimal solution obtained by the particle swarm algorithm, and when after n moments, the learner completes the learning of the whole path L, the pheromone on each road section is globally updated as in equation (21).

$$\tau_{ij}(t+n)=\begin{cases}(1-\rho)\times\tau_{ij}(t)+\Delta\tau_L^U & (r_i\to r_j)\in L\\ (1-\rho)\times\tau_{ij}(t) & Other\end{cases} \tag{21}$$

where ρ denotes the pheromone volatilization factor, $0\le\rho\le 1$; $\Delta\tau_L^U$ denotes the pheromone increment left by the user U on the path L, as in equation (22).

$$\Delta\tau_L^U=\begin{cases}\dfrac{Q}{Len} & (r_i\to r_j)\in L\\ 0 & Other\end{cases} \tag{22}$$

where Len denotes the length of path L and Q is a constant.

5) Selection probability of path. The probability of choosing r_j as the next learning object when learner U completes the learning of learning object r_i at the moment t, as in equation (23).

$$p_{ij}(t)=\begin{cases}\dfrac{\left[\eta_{ij}(t)\right]^a\times\left[\tau_{ij}(t)\right]^\beta}{\sum\left(\eta_{ij}(t)^a\times\left[\tau_{ij}(t)\right]^\beta\right)} & (r_i\to r_j)\in L(i)\\ 0 & Other\end{cases} \tag{23}$$

where η_{ij} is the heuristic information transferred from r_i to r_j; τ_{ij} denotes the pheromone concentration on the path from r_i to r_j; α denotes the parameter of the influence of the heuristic information η_{ij} on the path selection probability, respectively; and β is the parameter of the influence of the information τ_{ij} on the path selection probability.

4. Knowledge tracing performance testing and path recommendation experiments

4.1. Accuracy Test for Vocal Level Measurement

4.1.1. Data sets. This chapter uses 2 real-world datasets for the experiments, these datasets are ASSIST2009, ASSIST2015, they contain different number of students, skills and records, and the number of problem records per student. The statistical information of these four datasets is shown in Table 1.

Table 1. The summary of datasets

Data set	Student number	Skill number	Record number	Questions for each student
ASSIST2009	4151	124	325637	79
ASSIST2015	19840	100	683801	35

4.1.2. Contrasting models. DKT: The LSTM reported in the original DKT paper was used as the subject network. For the implementation of the LSTM, a single-layer LSTM with 200 hidden units was used in this chapter. The initial value of the learning rate was set to 0.001 and the model was optimized using Adam's algorithm and paradigm trimming, with the threshold for paradigm trimming set to 50. The batch size was set to 30 for all datasets.

DKVMN: For experiments on the ASSIST2009 dataset, the memory size was set to 20; the initial value of the learning rate was set to 0.05; and the batch size was set to 32.

DKT-DSC In implementing the LSTM, this chapter uses a single-layer LSTM with 200 hidden units and $keep\_pr\ ob$ is set to 0.6.

In this chapter, 80% of the data from each dataset is used for model training and the other 20% is used for testing. For each dataset, the training process was repeated 100 times with the aim of reducing the randomness of the results of a single experiment and improving the stability and reliability of the evaluation results.

4.1.3. Prediction of student performance. The main purpose of this section is to compare the performance of the models in this paper and the baseline model in predicting student performance. Table 2 shows the average test AUC and average test ACC of all models on the datasets.

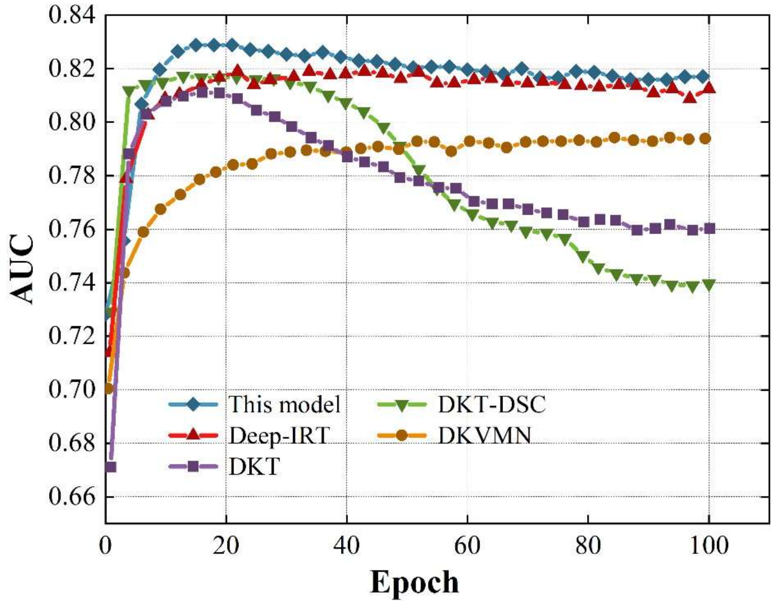
The model in this paper outperforms the other four state-of-the-art KT models based on DKT or DKVMN on all datasets. Specifically, on the ASSIST2009 dataset, the average test AUC of this paper's model is 0.8183 and the average test ACC is 0.7675, both of which are more than 0.015 higher than the baseline model.

On the ASSIST2015 dataset, the model AUC and ACC are 0.7468 and 0.7654, respectively, which are much higher than the highest 0.7281 and 0.7528 in the baseline model.

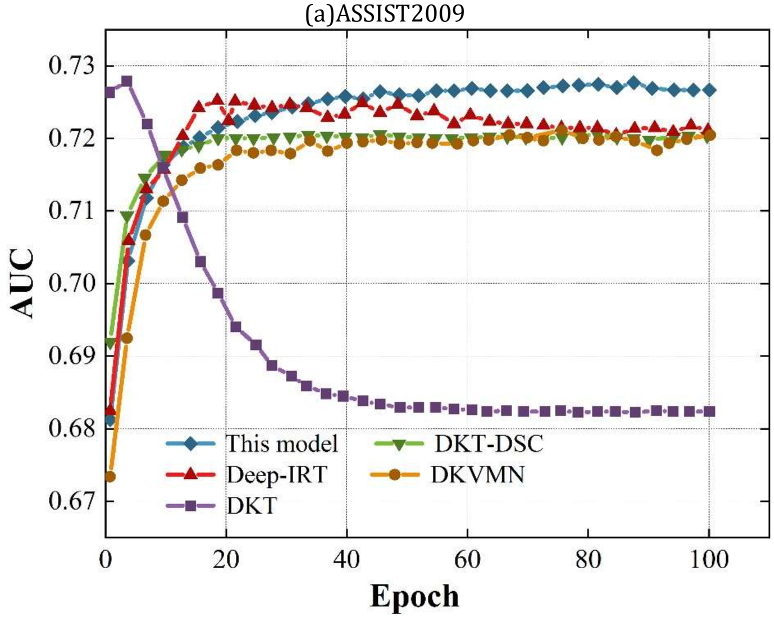
Table 2. Experimental results on student performance prediction

Model	ASSIST2009		ASSIST2015	
	AUC	ACC	AUC	ACC
DKT	0.7855	0.7465	0.6941	0.7439
DKVMN	0.7917	0.7498	0.7223	0.7441
Deep-IRT	0.8024	0.7508	0.7281	0.7528
DKT-DSC	0.7842	0.7389	0.718	0.7481
This model	0.8183	0.7675	0. 7468	0.7654

From Table 2, it is found that the performance of the models in this paper is quite close in predicting student performance. It is worth noting that the average test AUC and the average test ACC are used as the evaluation metrics in this chapter during the model evaluation process. A deeper look into the robustness of the model reveals that the model performs much better in this regard. To validate this finding, the results of the AUC curves on all test datasets are visualized as shown in Figure 2. Where (a) and (b) are ASSIST2009 and ASSIST2015 datasets, respectively.



(a)ASSIST2009



(b)ASSIST2015

Fig. 2. The AUC curve results

In order to show the robustness of the models in this paper more comprehensively, the test AUC and test ACC results of all models for the initial 40 epochs on different datasets are analyzed. Table 3 shows the results for the ASSIST2009 dataset and Table 4 shows the results for the ASSIST2015 dataset.

The choice of presenting only the result data for the first 40 epochs is based on two main considerations: first, according to the experimental results, the test performance of these models has stabilized after 40 epochs; second, such a choice helps to keep the information consistent and concise.

It can be clearly observed that the models in this paper show more robustness compared to the mechanical parts models. In addition, according to the maximum test AUC and maximum test ACC achieved by each model in the first 40 epochs, this paper's model achieves higher maximum test AUC and maximum test ACC than the Deep-IRT model on the ASSIST2009 dataset, which further proves the robustness and performance advantages of this paper's model.

Table 3. Test AUC and test ACC of all models on the ASSIST2009 dataset

ASSIST2009-AUC	ASSIST2009-ACC

This model	DKTDSC	Deep-IRT	DKVMN	DKT	This model	DKTDSC	Deep-IRT	DKVMN	DKT
0.7316	0.7303	0.7193	0.7001	0.6763	0.6947	0.7221	0.7037	0.6973	0.7207
0.7437	0.8082	0.7206	0.7181	0.7177	0.7159	0.7562	0.7034	0.7109	0.7309
0.7484	0.8128	0.7405	0.7331	0.7828	0.7202	0.7557	0.7113	0.7174	0.735
0.7547	0.819	0.7799	0.7494	0.7872	0.7255	0.7565	0.7441	0.7272	0.745
0.7867	0.8166	0.8003	0.7463	0.7913	0.7426	0.7544	0.7537	0.731	0.7432
0.7895	0.819	0.8018	0.754	0.8072	0.7471	0.7568	0.7613	0.7335	0.7537
0.8107	0.8218	0.8067	0.7611	0.8113	0.7421	0.7583	0.7593	0.7393	0.7371
0.8136	0.8164	0.8065	0.7636	0.8086	0.7472	0.7629	0.7654	0.7318	0.7432
0.8211	0.8224	0.8137	0.7634	0.8137	0.7515	0.7676	0.7629	0.7347	0.7484
0.8249	0.8139	0.814	0.7707	0.8098	0.7531	0.7652	0.7659	0.7365	0.7523
0.8252	0.8146	0.816	0.7751	0.8096	0.7592	0.7603	0.7653	0.7379	0.7555
0.8253	0.8158	0.8162	0.7688	0.8149	0.7558	0.7619	0.768	0.7473	0.7525
0.827	0.8203	0.8124	0.7761	0.8114	0.7571	0.7647	0.765	0.7432	0.7498
0.8357	0.8156	0.8095	0.7798	0.8138	0.7594	0.7623	0.7656	0.7489	0.7428
0.8336	0.8236	0.8128	0.776	0.8175	0.7628	0.7609	0.7706	0.7477	0.7489
0.8284	0.8157	0.8163	0.7791	0.8155	0.7637	0.764	0.7724	0.751	0.7449
0.8311	0.8163	0.8186	0.7783	0.8127	0.765	0.7614	0.771	0.7453	0.7511
0.8359	0.819	0.8218	0.7822	0.8137	0.7697	0.7669	0.7718	0.7504	0.756
0.8286	0.822	0.8174	0.7882	0.8103	0.7666	0.7598	0.7636	0.7472	0.7491
0.8294	0.8244	0.8153	0.7822	0.8173	0.764	0.7632	0.7687	0.7502	0.7479
0.8349	0.8159	0.8195	0.7874	0.8106	0.7732	0.7622	0.7709	0.753	0.7491
0.8305	0.8178	0.826	0.7886	0.8162	0.772	0.7632	0.7676	0.7526	0.7525
0.8332	0.8204	0.8228	0.7872	0.8072	0.776	0.7635	0.7742	0.7505	0.7544
0.8263	0.8228	0.8205	0.7939	0.8135	0.7762	0.7648	0.77	0.7494	0.7504
0.835	0.8162	0.8197	0.7904	0.8093	0.7782	0.7625	0.7716	0.7484	0.748
0.8294	0.8182	0.8174	0.7877	0.808	0.777	0.7652	0.7723	0.753	0.7499
0.8254	0.8181	0.816	0.791	0.8038	0.7677	0.7656	0.7669	0.7517	0.7479
0.8274	0.8237	0.8177	0.7882	0.8096	0.7708	0.7601	0.7722	0.7508	0.7544
0.8336	0.8133	0.8159	0.7942	0.8037	0.7689	0.764	0.7733	0.7589	0.7514
0.834	0.8201	0.8187	0.7926	0.8054	0.7769	0.761	0.7706	0.7527	0.7557
0.8312	0.8221	0.8237	0.7961	0.8031	0.7759	0.7624	0.7653	0.7594	0.7541
0.83	0.8154	0.8174	0.7901	0.797	0.7748	0.7587	0.7726	0.7525	0.744
0.828	0.8163	0.8225	0.7928	0.8004	0.7689	0.7637	0.7713	0.7561	0.757
0.8237	0.8147	0.8207	0.7958	0.799	0.7712	0.7645	0.7667	0.7565	0.7565
0.8337	0.8145	0.8263	0.7921	0.7941	0.7774	0.7561	0.77	0.7597	0.7558

0.8311	0.8152	0.8258	0.7881	0.7928	0.7711	0.7656	0.7713	0.7532	0.7508
0.8313	0.8108	0.8241	0.7958	0.7925	0.7721	0.7609	0.7719	0.7505	0.7517
0.8246	0.8165	0.8224	0.7956	0.7897	0.7702	0.764	0.7638	0.7515	0.7515
0.8289	0.8083	0.8229	0.792	0.7908	0.7764	0.7569	0.7677	0.7511	0.7508
0.8311	0.809	0.8187	0.7881	0.7886	0.7767	0.7595	0.7667	0.7493	0.7546

Table 4. Test AUC and test ACC of all models on the ASSIST2015 dataset

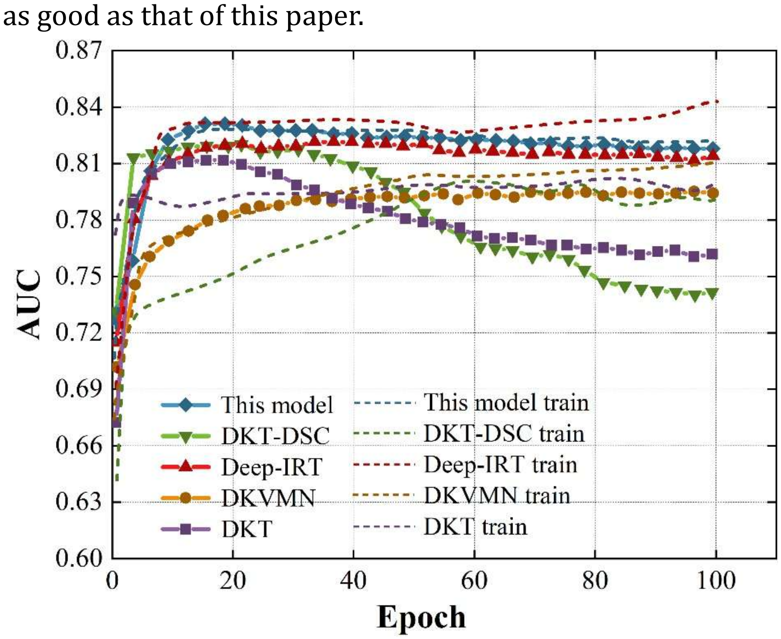
ASSIST2015-AUC					ASSIST2015-ACC				
This model	DKTDSC	Deep-IRT	DKVMN	DKT	This model	DKTDSC	Deep-IRT	DKVMN	DKT
0.64	0.661	0.6449	0.6207	0.6291	0.6637	0.6738	0.6644	0.6442	0.6406
0.6553	0.6711	0.6613	0.6242	0.6566	0.6704	0.6749	0.6716	0.6502	0.6523
0.6663	0.6762	0.6563	0.6274	0.6513	0.6732	0.6762	0.6789	0.6603	0.6702
0.6669	0.694	0.6688	0.6336	0.6746	0.6806	0.6813	0.6849	0.6617	0.668
0.6766	0.7	0.67	0.6344	0.6832	0.6864	0.6798	0.68	0.6603	0.6746
0.676	0.6973	0.6778	0.641	0.6922	0.6886	0.6828	0.6783	0.6565	0.6743
0.6847	0.708	0.6806	0.6483	0.6957	0.6819	0.6871	0.6805	0.6566	0.6804
0.6965	0.7068	0.6892	0.6467	0.7053	0.6866	0.6852	0.6832	0.6628	0.6753
0.6973	0.7103	0.6928	0.6429	0.7054	0.6846	0.6916	0.6822	0.6618	0.6839
0.7001	0.7137	0.6971	0.6489	0.7101	0.6889	0.6911	0.6845	0.6608	0.6789
0.7062	0.7181	0.698	0.6526	0.7118	0.6915	0.6935	0.6825	0.6601	0.6725
0.7106	0.7173	0.705	0.649	0.7118	0.6899	0.6926	0.6883	0.6649	0.6792
0.7116	0.7168	0.7007	0.6516	0.7163	0.6897	0.6904	0.6875	0.6651	0.6765
0.7176	0.7158	0.701	0.6563	0.7146	0.6924	0.6881	0.6847	0.6663	0.6805
0.7186	0.7247	0.7052	0.6521	0.7226	0.6856	0.6905	0.6839	0.6678	0.6829
0.7189	0.7213	0.7037	0.6535	0.7108	0.6885	0.6908	0.6865	0.6634	0.6751
0.721	0.7264	0.7101	0.6529	0.7175	0.6936	0.6933	0.6859	0.6589	0.6806
0.7164	0.7272	0.7066	0.6604	0.7153	0.6897	0.6937	0.6822	0.6634	0.6824
0.7209	0.7224	0.7122	0.6594	0.7142	0.6928	0.6925	0.6908	0.6597	0.6817
0.7247	0.7207	0.7052	0.6579	0.7137	0.6925	0.6906	0.6862	0.6657	0.6757
0.7258	0.7269	0.7084	0.6601	0.708	0.6925	0.6919	0.6856	0.6648	0.6793
0.7229	0.7275	0.7096	0.6529	0.7177	0.6924	0.6972	0.6933	0.6592	0.6765
0.7212	0.7237	0.7108	0.66	0.7151	0.697	0.6947	0.6811	0.6653	0.6843
0.7276	0.7257	0.7131	0.6543	0.713	0.6948	0.691	0.6912	0.6596	0.68
0.7209	0.7217	0.716	0.6579	0.709	0.6941	0.697	0.6922	0.6596	0.6842
0.7186	0.7227	0.7078	0.6573	0.7097	0.6959	0.6937	0.6894	0.6598	0.682
0.7243	0.7185	0.7069	0.6595	0.7066	0.6966	0.6881	0.6829	0.6655	0.6816

0.7255	0.7168	0.71	0.657	0.7048	0.689	0.6922	0.6833	0.6608	0.6851
0.7261	0.7163	0.7157	0.6619	0.7072	0.6931	0.6948	0.6911	0.6678	0.6807
0.7198	0.7171	0.7092	0.657	0.7067	0.6914	0.6927	0.6931	0.6628	0.6796
0.7212	0.7159	0.716	0.6637	0.7041	0.6983	0.6891	0.6867	0.6637	0.6783
0.724	0.7127	0.7139	0.6543	0.7029	0.6917	0.6908	0.6928	0.6684	0.678
0.7242	0.7166	0.7139	0.6614	0.699	0.6974	0.6845	0.6867	0.6661	0.685
0.7236	0.7148	0.7123	0.6573	0.703	0.6973	0.6884	0.6893	0.6614	0.6851
0.7189	0.7091	0.7109	0.6621	0.6992	0.6976	0.6839	0.6856	0.6689	0.6837
0.7212	0.7075	0.7193	0.6597	0.6937	0.6909	0.6885	0.6951	0.6669	0.6825
0.7244	0.7023	0.7143	0.6611	0.7017	0.6969	0.6822	0.691	0.6626	0.6832
0.7247	0.7032	0.714	0.6559	0.7021	0.6932	0.6831	0.6943	0.6664	0.6837
0.7182	0.7033	0.7165	0.6563	0.7004	0.6936	0.6863	0.6924	0.6682	0.6868
0.7191	0.6999	0.7099	0.6582	0.6973	0.6913	0.6792	0.6876	0.6661	0.6869

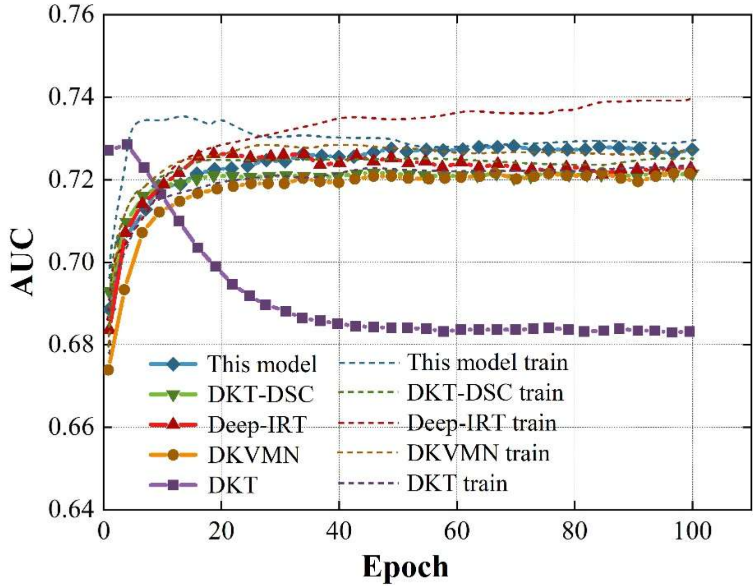
4.1.4. Mitigating overfitting. Figures 3 and 4 show the AUC curves and ACC curves of the model on the dataset, respectively, demonstrating the variation of AUC as well as ACC with training rounds for the model and other comparison models on all datasets. For any given dataset, this chapter uses colors or symbols to distinguish different models. For the performance curves of the same model on the training and test sets, this chapter distinguishes them by the voxels of the lines.

Specifically, the difference between the AUC or ACC curves of DKT-DSC increases rapidly after 20 epochs, and the difference between the AUC or ACC curves of DKVMN and Deep-IRT both increase rapidly after only 6 epochs. The model in this paper performs well in mitigating overfitting, and the AUC or ACC of the training and test sets are always close to each other, and the AUC and ACC of the test set steadily increase.

It is worth noting that both the model in this paper and the DKVMN model are effective in mitigating overfitting. However, since DKVMN only predicts students based on their current responses and does not take into account their skill mastery level prior to their responses, its AUC performance and ACC performance are not as good as that of this paper.

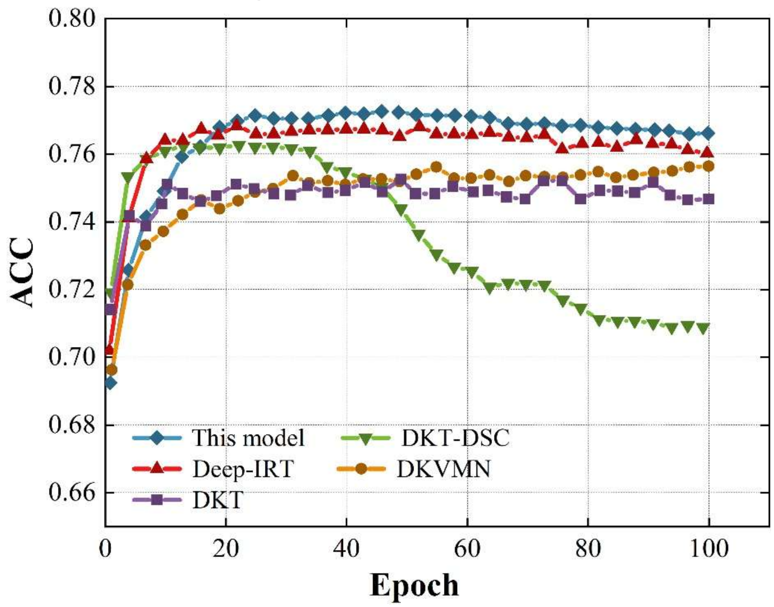


(a)ASSIST2009

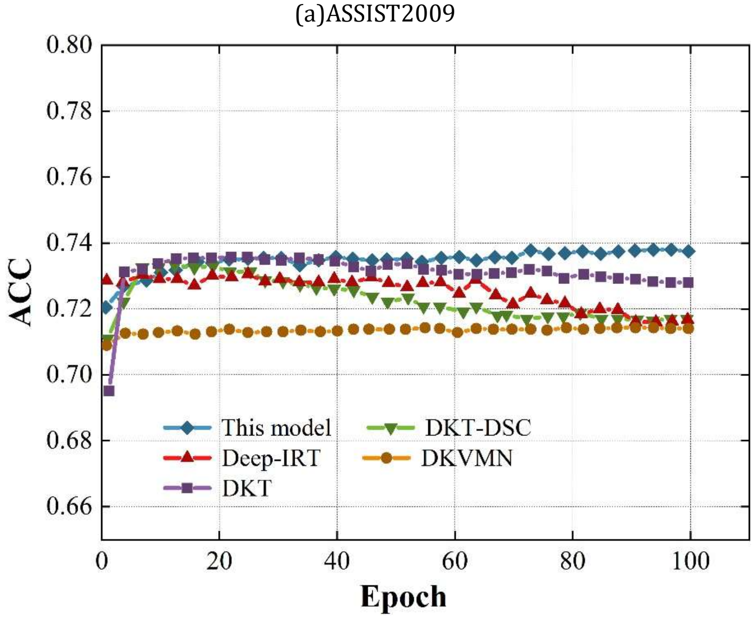


(b)ASSIST2015

Fig. 3. The AUC curve results



(a)ASSIST2009



(b)ASSIST2015

Fig. 4. The ACC curve results

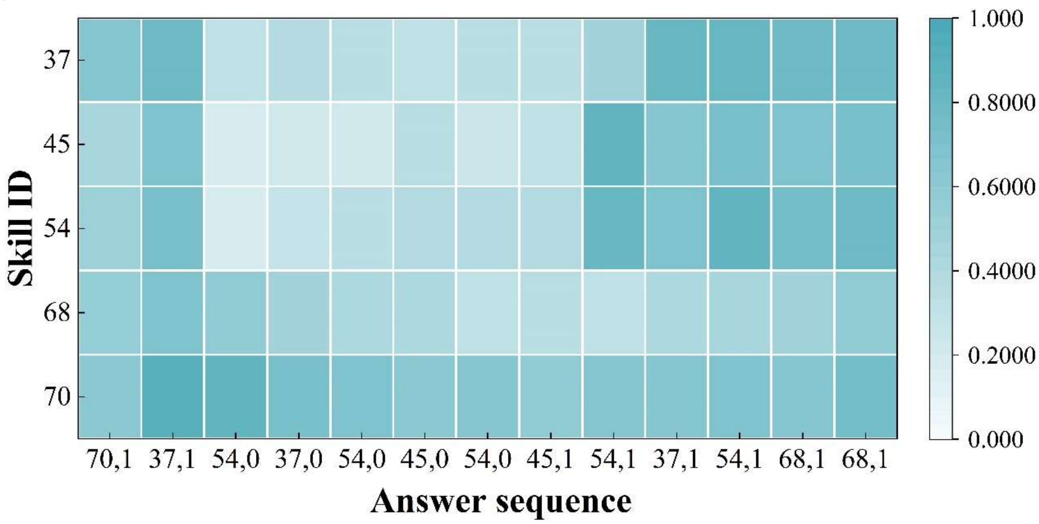
4.1.5. Knowledge tracking results. The main objective of this section is to validate the effectiveness of this paper's model presented in this chapter in tracking changes in students' skill mastery levels. To this end, this chapter randomly selects a student's answer records from the ASSIST2009 dataset over a period of time and uses this paper's model to track the change in the student's skill mastery level over time. The tracking results of the model are shown in Figure 5. Where (a) and (b) are time series performance plots and first and last mastery level radar plots, respectively.

In subplot (a), the horizontal axis represents the student's answer sequence, and each record in the sequence is presented as a binary group. The answer sequence provides a clear view of the skills and corresponding answer performance that the student was exposed to at different time steps. The vertical axis represents the five skills tracked by the model. The color shade of each grid in the graph visually reflects the student's mastery level of each skill at the corresponding time step. The darker the color, the higher the student's mastery level of the corresponding skill at that time step; conversely, the lighter the color, the lower the mastery level of the corresponding skill.

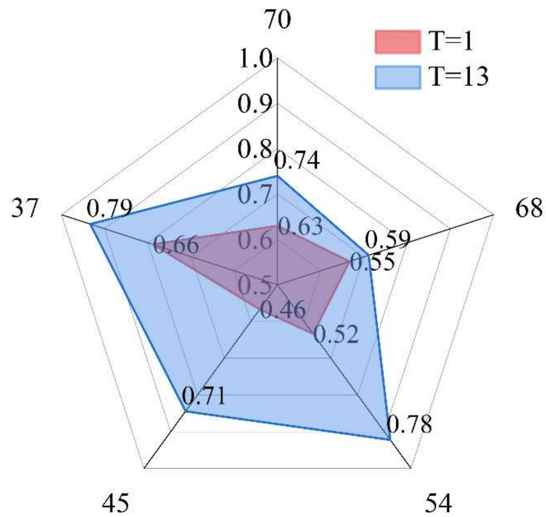
Subfigure (b) is a radar plot that shows the student's mastery level of the five skills at the beginning (T=1) and end (T=13) time steps. Each axis of the radar plot represents a skill, and the scale on the axis represents the student's mastery level of the corresponding skill (ranging from 0 to 1).

Taking the mastery level of vocal skill 37 as an example, the tracking results show that on the 3rd time step, the student answered skill 37 incorrectly, at which time the model tracking results show that the student's mastery level of skill 37 shows a decreasing trend, i.e., the color changes from dark to light. On the 6th time step, the student again answered Skill 37 incorrectly, and the results showed that the student's mastery level of Skill 37 further decreased and the color became lighter. On the 10th time step, the student answered Skill 37 correctly, and the results showed that the student's mastery level of Skill 37 increased, i.e., the color became lighter to darker. This series of results shows that the model in this paper can accurately capture the trend of students' skill mastery level at different time steps and visualize it through the color shade change.

At T = 1, when students first attempt to answer the exercises, their mastery levels of all skills are relatively low, which reflects the general state of students at the beginning of learning. Over time, at T = 13, students showed good skill growth as their mastery levels of skills 37, 45, and 54 increased significantly.



(a)Time sequence performance diagram



(b)Master horizontal radar

Fig. 5. Model tracking results

4.2. Learning path recommendation experiment

The ASSIST2009 and ASSIST2015 datasets will continue to be applied, divided into 80% training set,20% test set, comparing DKT-CF, ANN-ACS, DKT-ACS algorithms, to measure the effectiveness of this paper's shortest path recommendation algorithm from the perspective of different evaluation indexes.

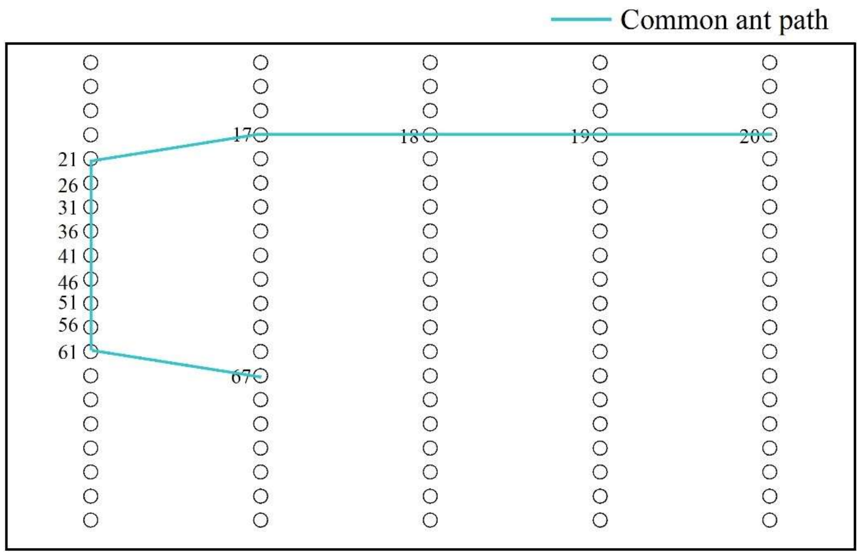
Mapping each knowledge point according to the relationship between knowledge points as a raster map location point of ant colony algorithm path planning. Combined with the syllabus and the opinions of experts in the field, the difficulty of the knowledge points is divided into three levels: simple, general and difficult after the formula calculation. Different difficulty of knowledge points in the map corresponding to the path distance is different, simple knowledge points of the path length is set to 1, general knowledge points of the path length of 2, difficult knowledge points of 3. In the raster map there are knowledge points of the location of the passable points, no knowledge points of the location is set as an obstacle point, if B for the advancement of knowledge A, set from A to B can be passable, the distance for the difficulty of the knowledge of the B corresponding to the length of the path, and vice versa. The distance is infinite and impassable. The resulting raster map is populated with rectangular maps as input for ACO path planning. The details of the knowledge points are shown in Table 5.

Table 5. Partial knowledge node information

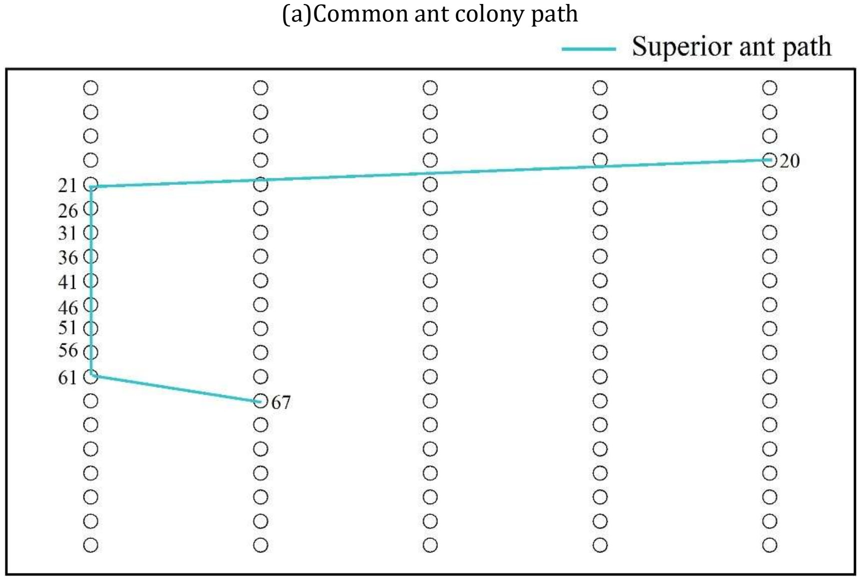
· Knowledge point number	Difficulty in knowledge	Corresponding path distance	Node number
1	Simplicity	1	14
2	Simplicity	1	78
3	Simplicity	1	36
4	Simplicity	1	85
5	General	2	21
6	Difficulty	3	9
7	General	2	66
8	Difficulty	3	43
9	Simplicity	1	57

10	Simplicity	1	38
11	Simplicity	1	72
12	Difficulty	3	15
13	Simplicity	1	43
14	General	2	25
15	Simplicity	1	63
16	Simplicity	1	46
17	Simplicity	1	33
18	Simplicity	1	73

Set any learning path: start and end points, so as to get different path recommendation results. If the learner wants to start learning from “Song Analysis” to the end of “Flower Cavity Technique”, input map nodes 20 and 67, and the path results of different ant colonies are shown in Figure 6.



(a)Common ant colony path



(b)Superior colony path

Fig. 6. Ant colony path selection

Calculate the data of each index between this algorithm and the other three algorithms respectively, in addition, calculate the complexity of the algorithm, and the final comparison results are shown in Table 6.

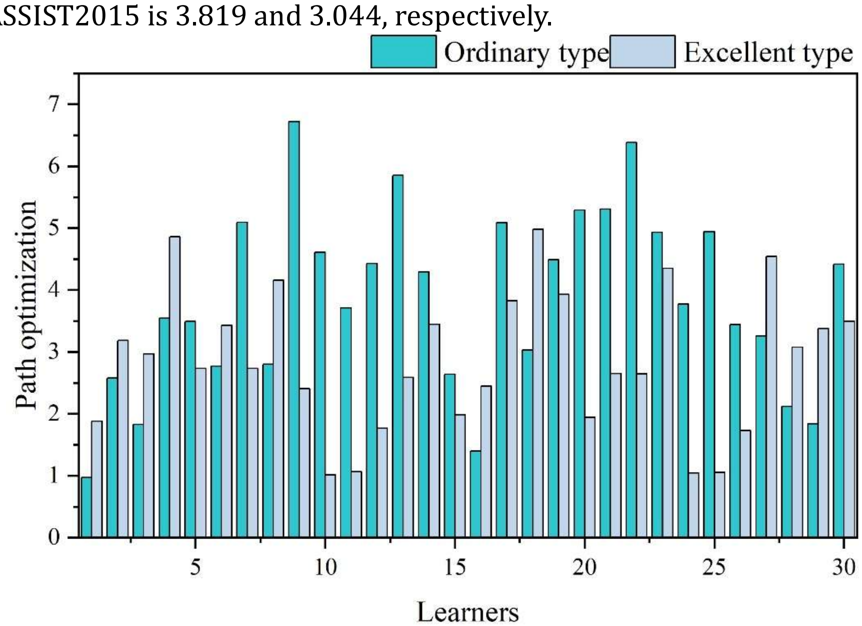
It can be seen that the present algorithm achieves optimal training results. On the ASSIST2009 dataset, the model accuracy and F1 of this paper are 0.904 and 0.691, respectively, which are the first among all models. On the ASSIST2015 dataset, the model coverage, precision, recall, and F1 of this paper are 0.813, 0.845, 0.533, and 0.657, respectively, which are still ahead of the remaining baseline models.

Table 6. Experimental data contrast

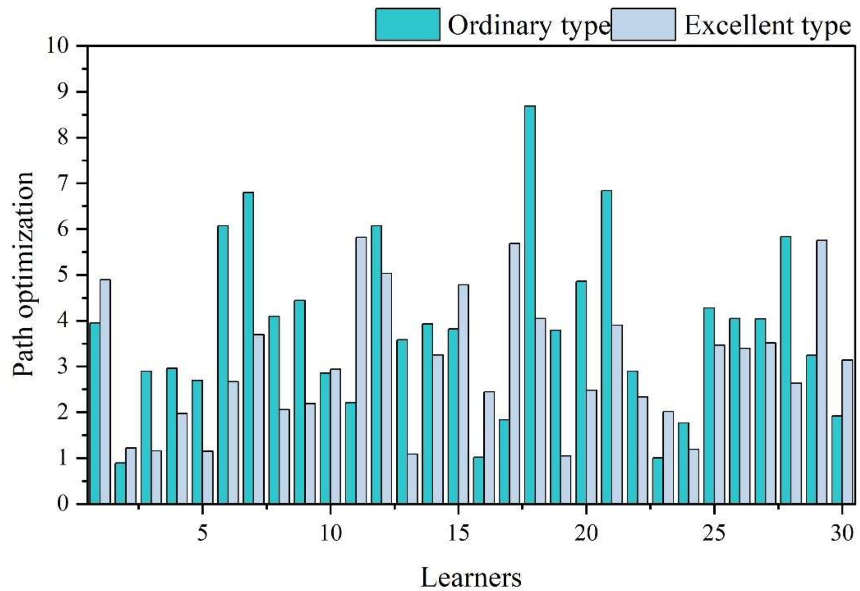
	ASSIST2009				ASSIST2015			
Parameter	DKT-CF	ANN-ACS	DKT-ACS	This model	DKT-CF	ANN-ACS	DKT-ACS	This model
Coverage	0.697	0.786	0.803	0.878	0.666	0.736	0.773	0.813
Precision	0.873	0.829	0.885	0.904	0.807	0.805	0.835	0.845
Recall	0.352	0.471	0.549	0.562	0.418	0.416	0.523	0.533
F1	0.505	0.604	0.648	0.691	0.542	0.549	0.647	0.657

Figure 7 shows 30 randomly selected samples of voice students from both datasets to calculate the amount of distance optimization between the recommended path and the original learning path. Figure 7 (a) shows ASSIST2009 and (b) shows ASSIST2015.

It can be seen that the model achieves path optimization for both average and excellent vocal students, with an average optimization of 4.297 and 3.242 on ASSIST2009, respectively. The average optimization on ASSIST2015 is 3.819 and 3.044, respectively.



(a)ASSIST2009



(b)ASSIST2015

Fig. 7. The distance optimization of the learning path

5. Conclusion

In this paper, based on the vocal music level determination by knowledge tracking, we design a learning path optimization recommendation model for vocal music skills based on ant colony-particle swarm hybrid algorithm. Experiments are designed separately to test the accuracy of vocal level determination and the effect of learning path optimization recommendation. The experiments found that on the ASSIST2009 dataset, the average test AUC of this paper's model is 0.8183, and the average test ACC is 0.7675, both of which are more than 0.015 higher than the baseline model. The model in this paper can accurately capture the trend of students' skill mastery level at different time steps and visually display it through color shade changes. The model is able to optimize the learning path of both ordinary and excellent vocal students.

References

[1] Fu, L. (2020). Research on the reform and innovation of vocal music teaching in colleges. Region-Educational Research and Reviews, 2(4), 37-40.

[2] Zhang, Y., Li, H., & Ye, P. (2020, August). Enlightenment on vocal music classroom teaching from the perspective of big data. In Journal of Physics: Conference Series (Vol. 1621, No. 1, p. 012102). IOP Publishing.

[3] Beeman, S. A. (2017). Perceptions of voice teachers regarding students' vocal behaviors during singing and speaking. Journal of Voice, 31(1), 111-e19.

[4] Zhang, Y. (2018). Enlightenment on vocal music classroom teaching from the perspective of neuroscience. NeuroQuantology, 16(6).

[5] Phillips, K. H., & Vispoel, W. P. (2021). The effects of class voice and breath-management instruction on vocal knowledge, attitudes, and vocal performance among elementary education majors. Visions of Research in Music Education, 16(1), 14.

[6] Bremmer, M., & Nijs, L. (2020, June). The role of the body in instrumental and vocal music pedagogy: a dynamical systems theory perspective on the music Teacher's bodily engagement in teaching and learning. In Frontiers in Education (Vol. 5, p. 79). Frontiers Media SA.

[7] Zhou, N. (2022). A study on the teaching of vocal music in colleges and universities from the perspective of aesthetic education. Advances in Educational Technology and Psychology, 6(1), 25-28.

[8] Li, S. (2018, April). The Status Quo and Development Ways of Vocal Music Teaching in Colleges. In 2018 4th International Conference on Education Technology, Management and Humanities Science (ETMHS 2018) (pp. 91-95). Atlantis Press.

[9] Zhang, Y., & Xing, J. (2018). Study on the Application of Pronunciation Training in Vocal Music Teaching in Colleges and Universities. Educational Sciences: Theory & Practice, 18(5).

[10] Candra, A., Budiman, M. A., & Hartanto, K. (2020, July). Dijkstra's and a-star in finding the shortest path: A tutorial. In 2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA) (pp. 28-32). IEEE.

[11] Mitchell, J. S. (2017). Shortest paths and networks. In Handbook of discrete and computational geometry (pp. 811-848). Chapman and Hall/CRC.

[12] Shanmugasundaram, N., Sushita, K., Kumar, S. P., & Ganesh, E. N. (2019). Genetic algorithm-based road network design for optimising the vehicle travel distance. International Journal of Vehicle Information and Communication Systems, 4(4), 344-354.

[13] Karur, K., Sharma, N., Dharmatti, C., & Siegel, J. E. (2021). A survey of path planning algorithms for mobile robots. Vehicles, 3(3), 448-468.

[14] Tharwat, A., Elhoseny, M., Hassanien, A. E., Gabel, T., & Kumar, A. (2019). Intelligent Bézier curve-based path planning model using Chaotic Particle Swarm Optimization algorithm. Cluster Computing, 22, 4745-4766.

[15] Ashokkumar, P., Arunkumar, N., & Don, S. (2018). Intelligent optimal route recommendation among heterogeneous objects with keywords. Computers & Electrical Engineering, 68, 526-535.

[16] De Prisco, R., Zaccagnino, G., & Zaccagnino, R. (2020). EvoComposer: an evolutionary algorithm for 4-voice music compositions. Evolutionary computation, 28(3), 489-530.

[17] Balliauw, M., Herremans, D., Palhazi Cuervo, D., & Sörensen, K. (2017). A variable neighborhood search algorithm to generate piano fingerings for polyphonic sheet music. International Transactions in Operational Research, 24(3), 509-535.

[18] Grienggrai Rajchakit,Pharunyou Chanthorn,Pramet Kaewmesri,Ramalingam Sriraman & Chee Peng Lim. (2020). Global Mittag–Leffler Stability and Stabilization Analysis of Fractional-Order Quaternion-Valued Memristive Neural Networks. Mathematics(3),422-422.

[19] Balázs Tóth,Yuefeng Jiang,Andras Szollosi,Zhe Zhang & László Csanády. (2024). A conserved mechanism couples cytosolic domain movements to pore gating in the TRPM2 channel.. Proceedings of the National Academy of Sciences of the United States of America(46),e2415548121.

[20] Shohanul Islam. (2025). Application of genetic algorithms to optimize fuel loading pattern and use of burnable absorbers to minimize power peaking and control excess reactivity in gas cooled fast reactor. Nuclear Engineering and Design113802-113802.