

PERSONALIZED LEARNING PATH RECOMMENDATION BASED ON LEARNER PROFILE AND KNOWLEDGE GRAPH

Xin XIE, Xiangyang FENG

School of Computer Science and Technology

Donghua University

Shanghai, China

e-mail: xiexin_0920@163.com, fengxy@dhu.edu.cn

Abstract. E-learning is increasingly popular because it allows learners to freely choose their class times and locations. However, traditional E-learning platforms face issues of information overload and fragmented resources. The proposition of the concept of personalized learning has effectively alleviated these problems. However, current personalized learning recommendation methods fail to comprehensively and systematically address learners' needs. To solve this issue, this paper proposes a learning path recommendation method based on learner profiles. First, by collecting learners' personal information, learning history, and behavior data, a learner profile is established considering multiple aspects. Then, generating a path evaluation function for learners from the profile. Using the Ant Colony Optimization algorithm, the most suitable personalized learning path for the learner's needs is searched within the knowledge graph. Experimental results demonstrate that the personalized learning path recommendations generated by our algorithm meet expectations and achieve the best overall performance in comparative experiments.

Keywords: Personalized learning, knowledge graph, learner profile, e-learning, learning path recommendation

Mathematics Subject Classification 2010: 05C90

1 INTRODUCTION

With the widespread adoption of the “Internet + Education” model and the close integration of big data and artificial intelligence technologies, the online education industry has experienced significant growth, accumulating a substantial amount of valuable data in the process. MOOC, which stands for Massive Open Online Courses, saw a pivotal year in 2011 when the Stanford University offered a free course titled “Introduction to Artificial Intelligence”, attracting 160 000 registrants from around the globe. Shortly thereafter, Udacity, Coursera, and edX, the launch of three major online course platforms which marked the participation of over a dozen world-renowned universities, establishing MOOCs as a globally popular educational model. In this educational context, online learning platforms, unrestricted by time and space, have become increasingly favored by learners for their convenience, freedom, and openness [1]. During the isolation periods of the pandemic, many universities adopted MOOC-based online teaching, allowing students to choose their study time and location independently [2]. Additionally, students could repeatedly review knowledge points they found challenging, thus addressing the limitation of single time teaching in traditional offline classrooms.

Despite the numerous advantages of online learning, it also presents new challenges. In most online education platforms, only a single learning path is provided to learners. This has resulted in less than 10 % of learners persisting through to the end of their courses [3]. A survey conducted on 50 students using MOOCs indicated that over 80 % of them felt that the course content did not vary according to their different learning styles, leading to a lack of content that suited their individual learning preferences [4]. Another study, which surveyed 94 staff members at Poland’s largest national teacher-training university, found that over 50 % of the respondents were either uncertain or disagreed with the notion that current online learning facilitates personalized education [5].

Firstly, learners need to spend considerable time and effort selecting appropriate educational resources, leading to information overload [6]. Secondly, most online education platforms lack personalized guidance and feedback for learners, resulting in feelings of isolation and confusion, and thus making it difficult for learners to stay motivated. Furthermore, the fragmented, unsystematic, and generalized nature of online learning resources [7] makes it challenging for learners to build a coherent knowledge system, thereby compromising optimal learning outcomes. Additionally, the current design of educational programs often centers around the teacher [8], relying heavily on the individual teaching experience of educators. This reliance leads to significant disparities in teaching effectiveness among teachers with varying levels of experience. Therefore, it is crucial to consider the personalized needs of students and recommend suitable learning materials and resources [9].

The concept of digital education has proposed new solutions to the challenges of online learning. It emphasizes a student-centered approach, utilizing technologies such as data mining and artificial intelligence to effectively analyze students’ learning characteristics. By identifying content that meets learners’ personalized needs from

a vast array of learning resources, digital education provides students with tailored learning plans. This approach can enhance learning efficiency and the utilization rate of online platforms [10], thereby overcoming the barriers and limitations of traditional online education.

Regarding personalized learning, some researchers have already considered different learner characteristics to customize personalized learning plans. Joseph et al. [11] and Klačnja-Milićević et al. [12] recommend resources based on the highest similarity scores derived from FSLSM learning styles. Kaur et al. [13] described learner profiles from three aspects: personal details, educational details, and knowledge level. Based on learners' self-assessments, they can upload their resume PDF files to the website, which are then parsed using the Affinda API and stored in a database. However, relying solely on learners' self-assessments to obtain their knowledge background is one-sided. Zhang et al. [14] argue that besides considering the relationship between learners and resources, factors such as social trust also influence personalized recommendation results. Wang et al. [15] administer tests to new users upon registration to gather information about their learning ability and interests, but this method does not consider the accuracy of learners' self-perceptions or the variability of their interests over time. Lin et al. [16] proposed a self-supervised reinforcement learning method based on semantic relationships in knowledge graphs and learners' knowledge structures to guide course recommendations, enhancing the accuracy and interpretability of the recommendations. Zhang [17] designed a student interest model, collecting student behavior data and using collaborative filtering to analyze their knowledge mastery, with feedback provided to teachers. However, the recommendation results of collaborative filtering depend on historical data, and sparse data can degrade recommendation quality. Liu [18] improved the traditional collaborative filtering's unidirectional influence problem by enhancing data density through increasing the influence sets within online learning communities.

This paper proposes a personalized learning path recommendation method based on learner profiles. The learner profile modeling is completed from three aspects: knowledge background, learning needs, and learning styles, using data analysis and mining techniques to deeply understand the characteristics of learners. Subsequently, different weighted coefficients are generated based on the profiles of different learners to obtain path evaluation criteria. Using heuristic algorithms, the optimal current path is iteratively identified in the knowledge graph to achieve personalized recommendations. The goal is to enhance learners' learning effectiveness and motivation, helping them better select and acquire learning paths that meet their needs and interests, thus fulfilling their personalized learning requirements. Experiments demonstrate that learners expressed high satisfaction with our algorithm, and among other baseline algorithms, the comprehensive performance of our algorithm was the best.

The rest of the paper is organized as follows. Section 2 introduces related research on learner profiles and personalized learning path recommendations. Section 3 describes the relevant terminology definitions and system framework of this paper. Section 4 presents the modeling process of learner profiles and the details of the

path recommendation algorithm. Section 5 conducts experiments and evaluations on the proposed recommendation algorithm. Section 6 provides conclusions and future work directions.

2 RELATED WORKS

Studies have shown that knowledge graphs can enhance the interpretability of the recommendation process [19]. In previous research on personalized learning path recommendations, researchers' perspectives can be summarized into the following three aspects:

2.1 Learner Preferences

Zhu et al. [20] proposed a multi-constraint learning path recommendation model based on a linear weighting formula. This study addressed the key factors often overlooked by existing learning path recommendation algorithms, such as learning scenarios, fragmented learning time, and fine-grained learning resources. Therefore, based on the analysis of the characteristics of learners and resources, the algorithm proposed eight learning paths corresponding to four different learning scenarios, considering constraints such as learning frequency, learning intervals, attention to knowledge units, and centrality, to provide the optimal learning path for learners in specific scenarios. Shi et al. [21], aiming at the semantic singularity of existing knowledge graph-based learning path recommendations, designed a multi-dimensional knowledge graph framework. Based on this framework, learning objects are categorized into three types: fundamental knowledge, algorithms, and tasks, with six constraints on the relationships between learning objects. From the input queries of learners, target learning objects are extracted, and all possible learning paths are identified using greedy search starting from the target learning objects. Then, considering the novelty, authority, and popularity of the research, a weighted method is used to constrain features, taking into account different learners' preferences for different paths. Liang et al. [22] focused on user-centric reasoning by utilizing reinforcement learning and graph convolutional networks to recommend videos, courses, and learning directions, deriving interpretive paths from learners to the recommended entities.

2.2 Learning Ability and Learning Effectiveness

Zhou et al. [23] identified suitable learning path categories for learners through similarity clustering and then used LSTM networks to predict learning outcomes, recommending paths with the best learning effects. This approach requires a large amount of prior learning data to mitigate the cold start problem. Wang et al. [15] clustered learners to form a repository of learner models, recommending learning paths with the fewest knowledge nodes for users with strong learning abilities. Yun

et al. [24] proposed an offline reinforcement learning algorithm called Doubly Constrained Deep Q-learning Network, which effectively enhances learning outcomes. Han et al. [25] improved the traditional collaborative filtering algorithm by incorporating the correlation of learning skills, recommending courses of appropriate levels to learners. Lang and Wang [26], addressing the shortcomings of existing classroom education in student learning, proposed a personalized knowledge point recommendation system model named Knowledge Graph-based Personalized Knowledge Points (KG-PKP). This model innovatively constructs an evaluation equation using accuracy rates, response times, and question types from answer records to assess students' mastery of knowledge points. Jiang et al. [27] described a knowledge point difficulty model and established a knowledge point mastery model based on learning behaviors and exam scores. These two models were used as input parameters to recommend dynamically personalized learning paths.

2.3 Balancing Preferences and Effectiveness

Xie et al. [28] proposed an algorithm that designed two model strategies. The first is a group analysis strategy, where a prototype is generated for each member within the group, then different learning paths are customized and integrated to form an optimized group learning path. The second is a prototype aggregation strategy, which first generates an overall group prototype, i.e., an overall learning path, and then allocates paths based on the preferences and knowledge of group members. Additionally, the algorithm defines upper and lower time boundary parameters, considering whether the group has sufficient time to complete the task. If time is ample, the algorithm prioritizes individual learning preferences of group members. If time is limited, it considers the pre-knowledge levels of each group member to maximize learning efficiency. If the time available is between the upper and lower bounds, multiple learning methods are considered. However, this algorithm requires a large amount of user interaction data, and the cold start problem cannot be avoided. Furthermore, due to the complexity of the model, it cannot support systems that require rapid responses. Ma et al. [29] proposed a multi-behavior learner modeling approach that considers aspects such as academic background, learning scenarios, learning styles, and preferences. They designed a cascading deep Q-network with a two-level reward function to enhance the accuracy of recommendation behaviors.

Some researches have considered learning path recommendations from other perspectives. Schwab et al. [30] proposed a visual learning platform supporting both linear and nonlinear learning plans from the perspectives of teachers and students. This platform utilizes concept maps to accommodate different learning styles. Zhang et al. [31] addressed the sparsity of interaction data in MOOC courses by iterating user historical data into a course graph, thereby obtaining course recommendation results. Diao et al. [32] used association rule mining to uncover connections between erroneous concepts. They then employed topological sorting to identify complete learning paths and subsequently derived weak concept paths. Saito and

Watanobe [33] generated paths based on learners' submission history logs. They created ability graphs for each learner and calculated similarities, using LSTM networks to predict path sequences. Xu and Wong [34] proposed a multi-feature LDA recommendation model based on educational image content, using Transformer and LSTM to deeply understand the relationships between various features in educational images, providing path recommendations for learners.

3 PROBLEM STATEMENT

This section primarily involves defining specific terms mentioned in the paper and explaining several sub-problems addressed in the research.

3.1 Definition of Terminology

To better explain the content of the article, the following definitions are provided for key terms:

Learner Profile (LP): A noun composed of a series of attributes used to describe learner information.

Knowledge Element (KE): The basic unit of knowledge learning.

Knowledge Graph (KG): A heterogeneous graph consisting of knowledge elements and relationships. In this paper, it is used as the basis for finding personalized learning paths that fit the learner profile.

Knowledge Exercises Set (KES): A collection of exercises that a learner has answered, where each exercise corresponds to a knowledge element. Multiple exercises can belong to one knowledge element. The collection of these exercises grouped by knowledge elements for a learner is termed as *KES*. The percentage of correctly answered exercises in *KES* is called Knowledge Exercises Set Accuracy as KES_{acc} . Specifically, the knowledge exercises set of student a for knowledge point k_1 can be denoted as $KES(a, k_1)$ and its accuracy is represented as $KES_{acc}(a, k_1)$.

Mastered Knowledge (MK): The set of knowledge elements that a learner has proficiently mastered. For any set of exercises corresponding to a knowledge element, if a learner's accuracy exceeds 80 %, that knowledge element is considered mastered by the learner.

Weak Knowledge (WK): The set of knowledge elements that a learner has not fully mastered or has not mastered at all. Similarly, for any set of exercises corresponding to a knowledge element, if a learner's accuracy is below 50 %, that knowledge element is considered weak for the learner.

Learning Path (LP): A sequence of knowledge elements studied to master WK.

Learning Need (LN): Refers to how a learner chooses to complete the learning of MK. Learners have different learning needs in different learning scenarios; for

example, during regular review, learners may prefer to thoroughly learn content, while during exam preparation, they may want to review more content in less time.

Learning Style (LS): In this paper, information perception and comprehension dimensions extracted from the Felder-Silverman learning style model [35] are considered. Specifically, if a student *S* prefers learning factual or real-life scenario courses and is good at grasping the framework of learning content, it can be preliminarily determined that they belong to the sensing type in the information perception dimension and the global type in the information comprehension dimension.

3.2 Main Work

Through constructing learner profiles and analyzing learners' characteristics, and based on these profiles, providing learners with personalized learning paths that best fit their needs and preferences, thereby enhancing their learning effectiveness and motivation, is the primary focus of this paper. The following briefly outlines several key components of the system described in this paper.

1. **Education Knowledge Graph Framework Design:** Here, we define the entity types, relationship types, and relevant attributes of each knowledge element for the knowledge graph. These will be used for subsequent path finding and path evaluation calculations as detailed in Table 1 and Table 2.
2. **Identification of Learner's Mastered Knowledge (MK) and Weak Knowledge (WK):** Utilizing the exercise-knowledge element association matrix and the learner's historical exercise response records, we derive the learner-knowledge element mastery matrix. This allows us to determine the learner's MK and WK, facilitating the identification of starting and ending points for path recommendation optimization.
3. **Learning Style Computation:** From the learner's online interaction record data, we extract useful information for experimental behavior pattern definition and abstraction into learning styles.
4. **Determination of Path Feature Weights and Path Evaluation Function:** The learner profile consists of MK and WK, LN, and LS. Based on this profile, different weights are assigned to path nodes, forming diverse path evaluation functions to establish the foundation for finding optimal scoring paths.
5. **Path Generation Algorithm:** This paper employs a depth-first search strategy based on the education knowledge graph. Starting from the path's origin, the algorithm sequentially explores neighboring nodes, maintaining a visited node set, and selecting an unvisited neighboring node based on pheromone matrices and roulette wheel selection as the next path node. This process continues until reaching the path's endpoint.

- 6. **Personalized Learning Path Recommendation:** Using ant colony algorithm, the learner’s WK serves as the target knowledge points for path planning. We conduct path searches based on the knowledge graph, evaluate the current path score according to the learner’s path evaluation function, update pheromone matrices, and aim to recommend the most suitable path in a short timeframe to achieve optimal recommendation outcomes.
- 7. **Experimental Evaluation:** On one hand, we adopt an external evaluation method where learners rate the quality of our paths and validate the accuracy of learning styles. On the other hand, we compare the practical effects of our algorithm with other algorithms based on path scores and time expenditures.

Entity Type	Description
Concept (CON)	Some basic concepts and definitions
Structure (STR)	Specific types or organizational forms of data
Algorithm (ALG)	Methods and procedures designed based on specific structural operations
Apply (APP)	Practical application problems addressed by certain algorithms or structures

Table 1. Knowledge graph entity types

Entity Type	Description
PageRank	Importance of entities in KG
Difficulty	Difficulty level of KE, ranging from 1 to 5
Type	Entity type of KE
TimeCost	Learning time of KE
Level	The hierarchy level of KE in the tree diagram

Table 2. Attributes of knowledge element entities

4 SYSTEM ALGORITHM DESIGN

In this section, two main tasks are undertaken. Firstly, modeling of learner profiles is conducted based on collected student data, considering LS, LN, MK and WK. Secondly, personalized learning path recommendations are formulated based on different learner profiles, using knowledge graphs to establish varied path evaluation criteria.

4.1 Learner Profile Modeling

4.1.1 Learning Style

The Field-Silverman [35] learning style scale was chosen as the research tool in this study. By analyzing learners’ behaviors across different dimensions, we inferred

their distinct LS. As shown in Table 3, this study selected the information perception and information understanding dimensions from the Field-Silverman learning style scale. The dimensions of information input and information processing were excluded because this study focuses solely on path recommendation for knowledge content, without considering the form of knowledge acquisition (learning from books or videos). Building upon the behavior pattern descriptions in literature [36], we refined these descriptions to better align with the recommendation algorithm used in this study. In Table 3, the “+” and “-” symbols indicate that the behavior characteristic makes the learner’s result lean more towards the left or right side of the dimension, respectively. For instance, in the Information Perception dimension, if a learner engages more frequently in studying algorithms and applying KE, they are more inclined towards the sensing style. Conversely, if they prefer studying conceptual types of KE, their results lean towards the intuitive style.

In Table 3, we can determine that each behavioral characteristic of learners falls into a high (H), medium (M), or low (L) range. For a given behavioral pattern P_i of a learner l , we make the following quantifications for ease of calculation:

$$P_i^l = \begin{cases} 1, & \text{if } P_i^l = H, \\ 0, & \text{if } P_i^l = M, \\ -1, & \text{if } P_i^l = L. \end{cases} \quad (1)$$

The quantification of LS for learner l is calculated as follows, where n represents the total sum of behavioral patterns in that dimension. If the result for learner l in the information perception dimension is the right style, then according to Table 3, the learner is classified as an intuitive type:

$$V_l(LS) = \frac{\sum_{i=1}^n P_i^l}{n}, \quad (2)$$

$$LS = \begin{cases} \text{right style,} & \text{if } V_l(LS) \in [-1, -1/3], \\ \text{balance style,} & \text{if } V_l(LS) \in (-1/3, 1/3), \\ \text{left style,} & \text{if } V_l(LS) \in [1/3, 1]. \end{cases} \quad (3)$$

4.1.2 Learning Need

For different learning scenarios, such as studying in a library or during fragmented time, learners might have varying preferences. For instance, during regular review sessions, learners may prefer a thorough understanding of the content, while before exams, they might aim to review as much knowledge as possible in less time. Based on this consideration, this study proposes three different LN: highest learning efficiency, key path (including more nodes with higher degrees), and minimum learning

content (the smallest number of knowledge points). More detailed calculation rules will be provided in Section 4.2.

4.1.3 Mastered Knowledge and Weak Knowledge

Through basic natural language processing methods and expert proofreading, we can obtain the exercise-KE association matrix M . Each row represents which KE is associated with exercise e , and each column represents all exercises associated with knowledge element k . Each row can have only one '1', while each column can have multiple '1's. This is represented as follows:

$$\begin{matrix} & k_1 & k_2 & \dots & k_m \\ \begin{matrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{matrix} & \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \dots & 0 \end{bmatrix} \end{matrix} \quad (4)$$

Next, based on the answer record E_a of learner a , we can generate $KES(a, k_t)$ for each knowledge element k_t . This allows us to derive the learner's MK and WK. The detailed process is provided in Algorithm 1.

Algorithm 1 Generate learner's MK and WK set.

$MK, WK = getMKandWK(M, E_a)$

Require: Exercises-Knowledge Point Matrix M ; Learner a 's exercise records E_a ;

Ensure: Learner a 's MK ; Learner a 's WK ;

```

1: for all  $e_i \in E_a$  do
2:    $k_j = getKnowledgePoint(M, e_i)$ 
   // Get the knowledge points  $k_j$  associated with exercise  $e_i$ 
3:    $KES(a, k_j).add(e_i)$ 
   // add exercise  $e_i$  to the knowledge exercise set of knowledge  $k_j$ 
4: end for
5: for all  $KES(a, k_t) \in a$  do
6:   if  $KES_{acc}(a, k_t) \geq 0.8$  then
7:      $MK.add(k_t)$  // add  $k_j$  to  $MK$ 
8:   end if
9:   if  $KES_{acc}(a, k_t) < 0.5$  then
10:     $WK.add(k_t)$  // add  $k_j$  to  $WK$ 
11:  end if
12: end for
13: return  $MK, WK$ 

```

Dimension	Category	Behavioral Features	Pattern Description	Threshold	
				L-M	M-H
Perceive Information	Sensing/Intuitive	Frequency of learning algorithm application knowledge (+)	Frequency of learning algorithm and application knowledge/total learning knowledge	< 50 %	> 75 %
		Duration of exercise study (+)	Duration of exercise study/total learning time	< 5 %	> 15 %
		Number of times learning conceptual knowledge (-)	Frequency of learning conceptual knowledge/total learning knowledge	< 50 %	> 75 %
		Frequency of clicks on course navigation (+)	Personal click count/average click count	< 0.5	> 1.2
Organize Information	Global/Sequential	Frequency of views of the course outline (+)	Personal click count/average click count	< 0.5	> 1.2
		Frequency of learning leaf nodes (-)	Learning leaf nodes/learning total nodes	< 50 %	> 75 %

Table 3. Learning style behavior pattern descriptions

Feature	Related Parameters	Weight
Number of Algorithm and Apply KE	$f_1 = \text{count}(K_{\text{Type}=\text{ALG or APP}})$	w_1
Number of Concept KE	$f_2 = \text{count}(K_{\text{Type}=\text{CON}})$	w_2
Number of High Level KE	$f_3 = \text{count}(K_{\text{Level}=\text{High}})$	w_3
Number of Low Level KE	$f_4 = \text{count}(K_{\text{Level}=\text{Low}})$	w_4
Reciprocal of KE's Difficulty	$f_5 = \sum \left(\frac{1}{K_{\text{Difficulty}}} \right)$	w_5
Reciprocal of KE's Time Cost	$f_6 = \sum \left(\frac{1}{K_{\text{TimeCost}}} \right)$	w_6
PageRank	$f_7 = \sum (K_{\text{PageRank}})$	w_7
Number of KE in the LP	$f_8 = \sum (K)$	w_8

Table 4. Path feature

4.2 Personalized Learning Path Recommendation

The previous section explained the process of profile modeling. This section relies on these profiles to customize personalized learning paths for different learners. The specific work will be described in two parts: defining the path evaluation function and outlining the path recommendation algorithm.

4.2.1 Path Evaluation Function

Once MK and WK are determined, we can establish the starting and ending points of a path. However, within a complex knowledge graph structure, there may be multiple reachable paths between two entities. To achieve personalized paths based on learner profiles modeled earlier, we quantify this using a path evaluation function tailored to different learners.

Building upon weighted methods outlined in [21] for quantifying different features, this study selects eight features from both learning style (objective) and learning needs (subjective) aspects to implement the evaluation function. For any given path P , its features are detailed as shown in Table 4.

Considering learning styles, the combination of learning styles derived from the two dimensions in Table 3 results in 9 possible combinations. The allocation of weights for these combinations is detailed in Table 5, where each row represents the weight distribution for the respective combination, with all unmentioned weights set to zero.

Considering learning needs, we provide learners with 3 different learning scenarios to choose from, as shown in Table 6. Each row specifies the weight distribution for the respective scenario, with all unmentioned weights set to zero.

According to the weighted algorithm, the scoring formula for any path score P_i is represented as:

$$P_i = f_{\text{Gaussian}(\text{path_length})} \sum_{j=1}^n (w_j * f_j(P_i)). \quad (5)$$

Where w_j represents the values corresponding to Table 5 and 6, f_j denotes the relevant parameters from Table 4, and $f_{\text{Gaussian}}(\text{path_length})$ signifies the Gaussian decay function, which attenuates the path score as the path length increases beyond a threshold. This is because longer learning paths are not necessarily better. It is as follows specifically:

$$f_{\text{Gaussian}}(\text{path_length}) = e^{-\frac{(\text{path_length} - \text{max_length})^2}{2\sigma^2}}. \quad (6)$$

4.2.2 Learning Path Recommendation Algorithm

The algorithmic model structure of this paper, as shown in Figure 1, requires inputting the learner's LS , LN , MK and WK , along with the knowledge graph G . The model outputs the learning path most compatible with the learner.

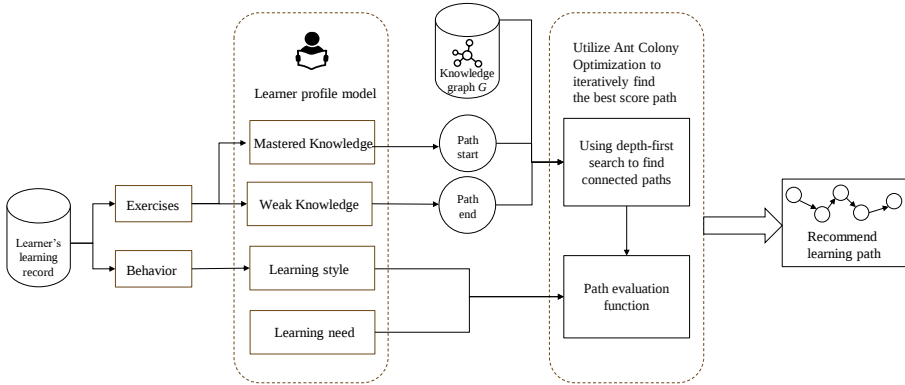


Figure 1. The structure of the learning path recommendation model

Learning Style	Weight
Sensing + Global	$w_1 = 0.5, w_3 = 0.5$
Sensing + Sequential	$w_1 = 0.5, w_4 = 0.5$
Intuitive + Global	$w_2 = 0.5, w_3 = 0.5$
Intuitive + Sequential	$w_2 = 0.5, w_4 = 0.5$
Balanced + Global	$w_1 = 0.25, w_2 = 0.25, w_3 = 0.5$
Balanced + Sequential	$w_1 = 0.25, w_2 = 0.25, w_4 = 0.5$
Sensing + Balanced	$w_1 = 0.5, w_3 = 0.25, w_4 = 0.25$
Intuitive + Balanced	$w_2 = 0.5, w_3 = 0.25, w_4 = 0.25$
Balanced + Balanced	$w_1 = 0.25, w_2 = 0.25, w_3 = 0.25, w_4 = 0.25$

Table 5. Learning styles and weights correlation

Here are the detailed algorithm descriptions: In Section 4.1, we have already determined the learner's LS , LN , MK , and WK . Within MK and WK , the learner

Algorithm 2 Random learning path generation.

$p = \text{randomPathGeneration}(G, n_{\text{start}}, n_{\text{end}})$

Require: Knowledge graph G ; The path start node n_{start} ; The path end node n_{end} ;
Pheromone array of the Graph A_p ;

Ensure: Random path p ;

```

1:  $p.add(n_{\text{start}})$  // Add the path start to  $p$ 
2:  $n_{\text{current}} = n_{\text{start}}$  // Set  $n_{\text{start}}$  to the current node
3: while  $n_{\text{current}} \neq n_{\text{end}}$  do
4:    $N = \text{getNeighbours}(n_{\text{current}})$  // Select a set of neighbor nodes that are not in
   the path and have not been visited
5:    $n_{\text{next}} = \text{choiceNextNode}(N, A_p)$  // According to  $A_p$  and roulette wheel
   method selecting a node from  $N$  as the next node to be visited
6:    $p.add(n_{\text{next}})$  // Add  $n_{\text{next}}$  to  $p$ 
7:    $n_{\text{current}} = n_{\text{next}}$  // Set  $n_{\text{next}}$  as  $n_{\text{current}}$  of the next loop
8: end while
9: return  $p$ 

```

Algorithm 3 The highest score learning path caculation p_b .

$p_b = \text{highestScorePathCaculation}(G, LS, LN, MK, WK, A_p)$

Require: Knowledge graph G ; Learner's learning style LS ; Learner's learning need
 LN ; Learner's mastered knowledge set MK ; Learner's weak knowledge set WK ;
Pheromone array of the Graph A_p ;

Ensure: The highest score learning path p_b ;

```

1:  $\text{initParams}()$  // Initializes  $N_{\text{ants}}$ ,  $A_p$ ,  $\text{evaporationRate}$ ,  $\text{iterationTime}$  of the ant
   colony optimization algorithm
2:  $n_{\text{start}}, n_{\text{end}} = \text{choiceStartAndEnd}(MK, WK)$ ;
   // Choice start and end node in  $MK$  and  $WK$ 
3:  $p_{\text{best}}, s_{\text{best}} = \text{None}, 0$  // Initializes  $p_{\text{best}}$  and  $s_{\text{best}}$ 
4: for  $i$  in  $\text{range}(\text{iterationTime})$  do
5:   for  $j$  in  $\text{range}(N_{\text{ants}})$  do
6:      $p_{\text{current}} = \text{generateRandomPath}(G, n_{\text{start}}, n_{\text{end}}, A_p)$ 
     // Generate a new path between  $n_{\text{start}}$  and  $n_{\text{end}}$ 
7:      $s_{\text{current}} = \text{caculatePathScore}(p_{\text{current}}, LS, LN)$ ;
     // Caculate the score of  $p_{\text{new}}$ 
8:     if  $s_{\text{current}} > s_{\text{best}}$  then
9:        $p_{\text{best}} = p_{\text{current}}, s_{\text{best}} = s_{\text{current}}$ ;
10:    end if
11:     $\text{updatePheromoneEvaporation}(A_p)$ 
12:     $\text{updatePheromoneDeposition}(A_p)$ 
13:  end for
14: end for
15: return  $p_{\text{best}}$ 

```

Learning Need	Weight
Highest Learning Efficiency	$w_5 = 0.5, w_6 = 0.5$
Main Path	$w_7 = 1$
Minimal Learning Time	$w_8 = 1$

Table 6. Learning needs and weights correlation

can freely select two nodes as the starting and ending points for path recommendation, denoted in Algorithm 3 by the function *choiceStartAndEnd*(*MK*, *WK*). Subsequently, based on the learner's *LS* and *LN*, referencing Tables 4, 5 and 6, we derive the learner's feature set and weights. Finally, this process yields the personalized path evaluation criteria for the learner.

Starting from the node n_{start} and using depth-first search, we explore connected paths to the endpoint n_{end} . Beginning at n_{start} , we sequentially search neighboring nodes, maintaining a set of visited nodes, and filter out unvisited neighbors not yet on the path. These neighbors form set N . For each node in N , we calculate the ratio of its pheromone in the list A_p to the total pheromone sum in N . This ratio represents the probability of selecting the node, where nodes with higher pheromone levels have a greater chance of being chosen as the next node in the path. Using a roulette wheel selection method, one node is chosen iteratively as the next node in the path until reaching n_{end} . For detailed algorithm flow, refer to Algorithm 2.

Algorithm 3 provides the overall framework of the algorithm. Using ant colony optimization, parameters such as the number of ants N_{ants} , pheromone lists A_p for each node, *evaporationRate*, and *iterationTime* are initialized. During each iteration, paths accessible to all ants are generated, evaluated based on the learner's path evaluation criteria (5), and continually update the local optimal path. After all ants have completed their traversal in each round, the pheromone evaporation and deposition are updated. Upon completion of iterations, the current optimal solution p_b is outputted.

5 EXPERIMENTAL EVALUATION

In this section, we employed a series of experimental schemes to validate the feasibility and effectiveness of the system. Firstly, we introduced the datasets and preparatory work required for the experiments. Secondly, a survey questionnaire was distributed to investigate the effectiveness of path recommendations. Finally, we compared our proposed algorithm with other algorithms.

5.1 Dataset and Experimental Procedure Description

In this paper, taking the Data Structures course for Computer Science as an example, we collected a total of 418 entities and 487 relationships by combining data from Wikipedia and textbook materials. If learning knowledge node B requires knowledge

node A as a prerequisite, there will be a relationship from A to B. The local structure of the knowledge graph is shown in Figure 2.

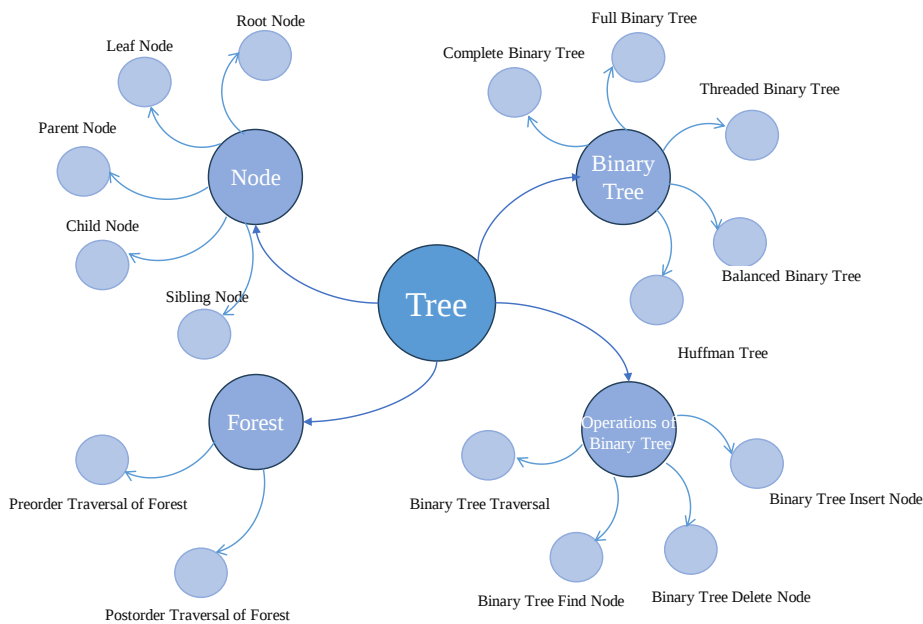


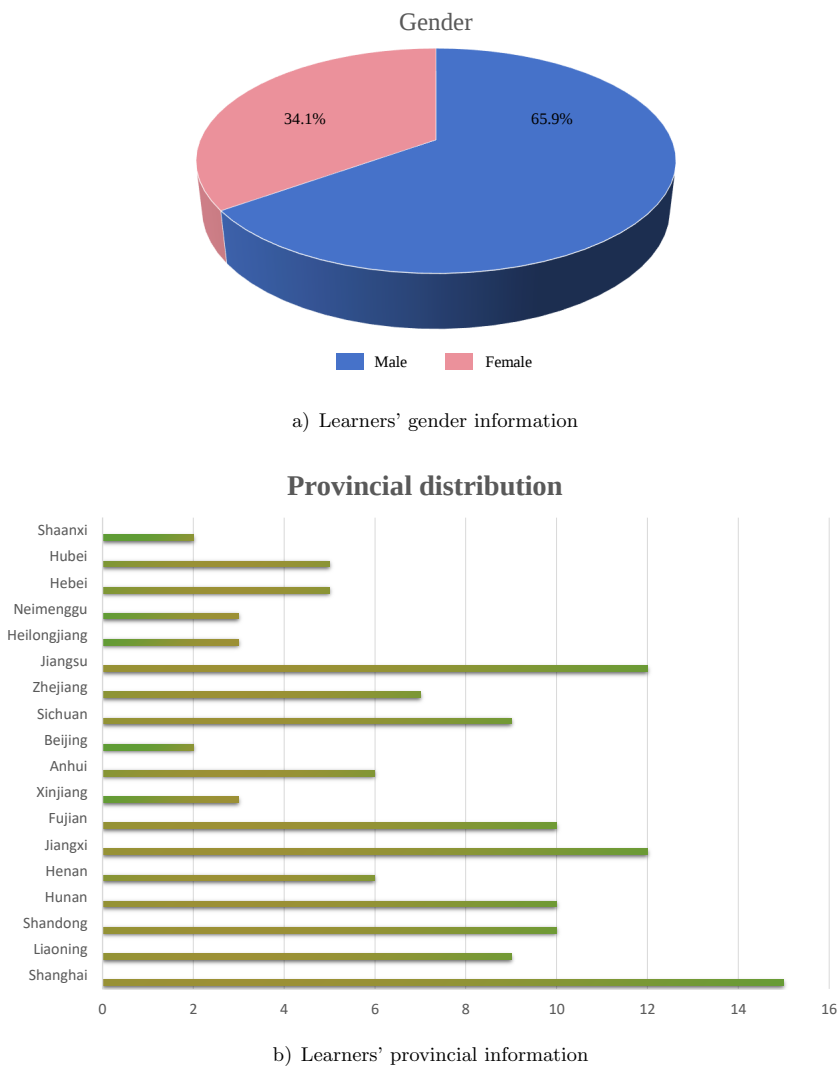
Figure 2. The part of the data structure knowledge graph

First, we designed a survey questionnaire consisting of three main parts. The first part gathers basic information about the learners, the second part assesses the learners' learning styles, and the third part presents learning paths generated by the system for different profiles, which the learners then evaluate. We set five levels of satisfaction for the path recommendations: 1 represents very dissatisfied, 2 represents dissatisfied, 3 represents neutral, 4 represents satisfied, and 5 represents very satisfied.

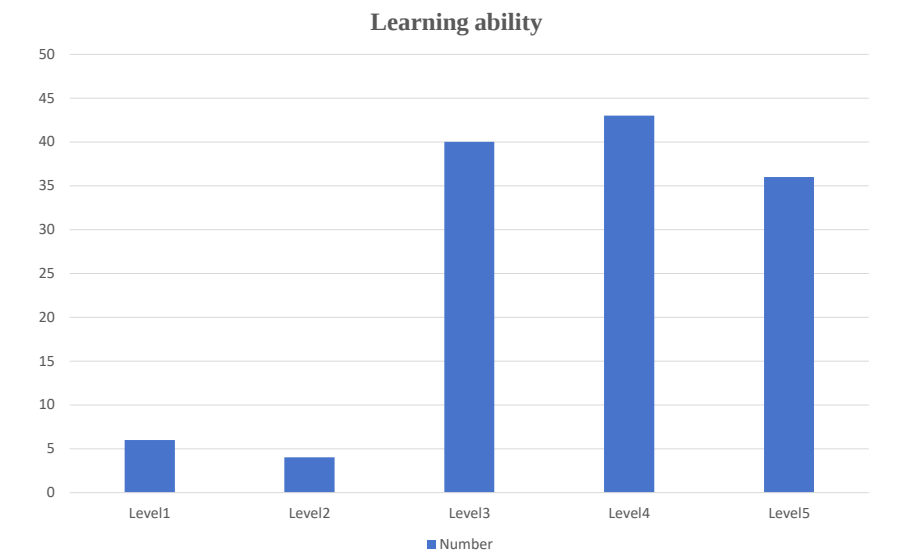
For the controlled experiment, we used both the Depth-First Search algorithm and Dijkstra's algorithm to compare the path generation scores within the same knowledge graph.

5.2 Analysis of Experimental Results

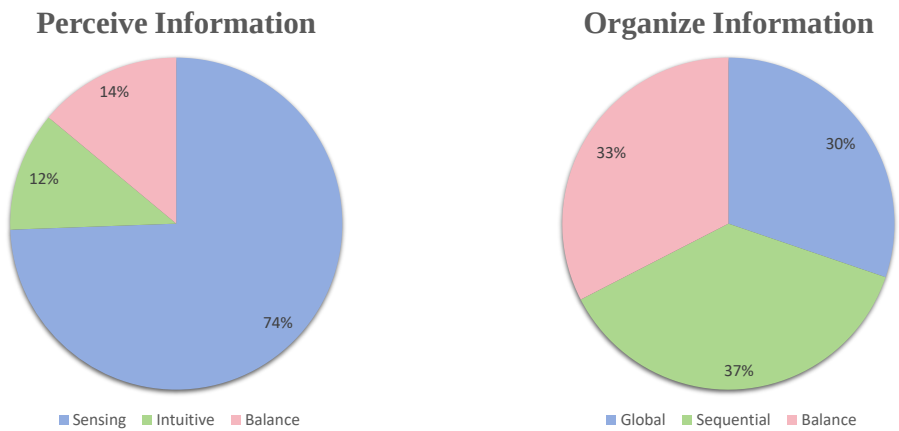
The experimental subjects were students majoring in computer science who had taken a data structures course. We received a total of 129 valid questionnaires. Figure 3 a) and Figure 3 b) present the basic information of these learners. From Figure 3 a), we can see that males accounted for two-thirds of the total number of participants, while females made up one-third. And these students came from 18 different provinces.



Additionally, we surveyed the learners' exam scores in the data structures course and mapped these scores to learning abilities. The full score is 100 points, with the learning ability categorized as follows: scores below 60 are mapped to learning ability 1, scores from 61–70 to learning ability 2, scores from 71–80 to learning ability 3, scores from 81–90 to learning ability 4, and scores from 91–100 to learning ability 5. The statistical results are shown in Figure 3 c). As can be seen, except for a very small number of students with lower learning abilities, the learning abilities are evenly distributed between 3 and 5.



c) Learners' learning ability information



d) Learners' learning style information

Figure 3. Overview of learners' information

In the second part of the questionnaire, we used 12 different multiple-choice questions to test the learners' self-perception of their learning styles. The first 6 questions were designed to test the information perception dimension, while the latter 6 questions were aimed at the information understanding dimension. Figure 3d) presents the distribution results of learning styles. From the figure, in the information perception dimension, the majority of students identified themselves as sensing types. This indicates that most students believe they are better at learning factual content but are less adept at innovative thinking. In the information understanding dimension, the data for global, sequential, and balanced types are evenly distributed, indicating a relatively balanced distribution.

```

{"id": "1784532", "exercise_time": [{"exercise_id": "52849367", "73918420", "16274895", "89456231", "34782159", "61529384", "48291763", "75360418"}, {"result": [true, false, true, true, false, true, true, true], "start_time": "2024-06-29 21:57:39", "end_time": "2024-06-29 22:12:25"}, {"exercise_id": "39582641", "47296183", "83647295", "19483726", "56473821", "72839145"}, {"result": [true, true, false, true, true, false], "start_time": "2024-06-30 08:15:20", "end_time": "2024-06-30 08:30:45"}, {"exercise_id": "81726354", "26495873", "59387126", "38274619", "91738264", "48572693", "73619485", "12947368"}, {"result": [true, false, true, true, false, true, true], "start_time": "2024-07-01 14:25:00", "end_time": "2024-07-01 14:43:12"}], "activity": [{"course_id": "035", "video_id": "a7c9d3f6b2x8y1z", "watching_time": 886, "start_time": "2024-05-31 21:57:39", "end_time": "2024-05-31 22:12:25"}, {"course_id": "035", "video_id": "p4k7m2n9l3q8w5r", "watching_time": 8455, "start_time": "2024-06-27 21:16:19", "end_time": "2024-06-27 23:37:14"}], "online_time": [{"start_time": "2024-06-29 21:48:44", "end_time": "2024-06-29 23:47:20"}, {"start_time": "2024-06-30 14:15:03", "end_time": "2024-06-30 15:28:15"}, {"start_time": "2024-06-29 21:20:56", "end_time": "2024-07-01 23:47:20"}], "navigation_clicks": ["2024-06-29 22:10:07", "2024-06-29 22:36:40", "2024-06-29 22:49:19", "2024-06-29 23:12:10", "2024-06-29 23:40:52", "2024-06-30 14:41:49", "2024-06-30 14:48:43", "2024-06-30 15:22:58", "2024-06-30 15:28:03", "2024-06-30 17:11:05", "2024-06-30 18:36:51", "2024-07-01 12:38:06", "2024-07-01 20:45:02"], "read_outline_times": ["2024-06-29 22:57:13", "2024-06-29 23:22:16", "2024-06-30 15:19:10"]}

```

Figure 4. The part of the learners' behavior dataset

We collected behavioral data from students on the OJ website and used the pandas library in Python for data cleaning. We filtered out learners with low activity levels on the website and irrelevant attributes (part of the data is shown in the Figure 4), obtaining the necessary behavioral features for our study. The final dataset includes online behavior data from 108 learners. Based on the problem-knowledge element association matrix and the video-knowledge element association matrix, we assessed the learners' mastery of knowledge points. From this, we identified the attributes required for calculating learning styles, which include: *learningFrequencyOfALG*, *learningFrequencyOfAPP*, *learningFrequencyOfCON*, *totalLearningFrequency*, *onlineTime*, *timeOfExercise*, *navigationClicks*, *outlineViews*, and *learningFrequencyOfLeaves*.

Using the data in Table 3, we converted these into the actual learning styles and, based on the learners' self-reported learning needs, generated personalized learning paths for them. According to the learners' self-perceived learning styles, Table 8 presents the accuracy of learning styles calculated from actual behavior and the average satisfaction with the generated paths. Over two-thirds of the learners' self-perceived learning styles align with their actual performance styles, and the majority of learners show at least one dimension of consistency between self-perception and actual performance. The average satisfaction score is 3.65, indicating that most

learners are satisfied with the path recommendations. Table 7 lists three examples of the recommended learning paths.

	Learning Style	Learning Need	Start Node	End Node	Path Result
Path 1	Sensing Global	High efficiency	Singly Linked List	Circular Queue	Singly Linked List, Linked List, Linear List, Queue, Sequential Queue, Circular Queue
Path 2	Intuitive Sequential	Critical path	Tree	Huffman Coding	Tree, Forest, Conversion between Forest and Binary Tree, Binary Tree, Huffman Tree, Huffman Coding
Path 3	Balance Balance	Minimal learning time	Graph	Dijkstra's Algorithm	Graph, Weight, Network, Shortest Path, Dijkstra's Algorithm

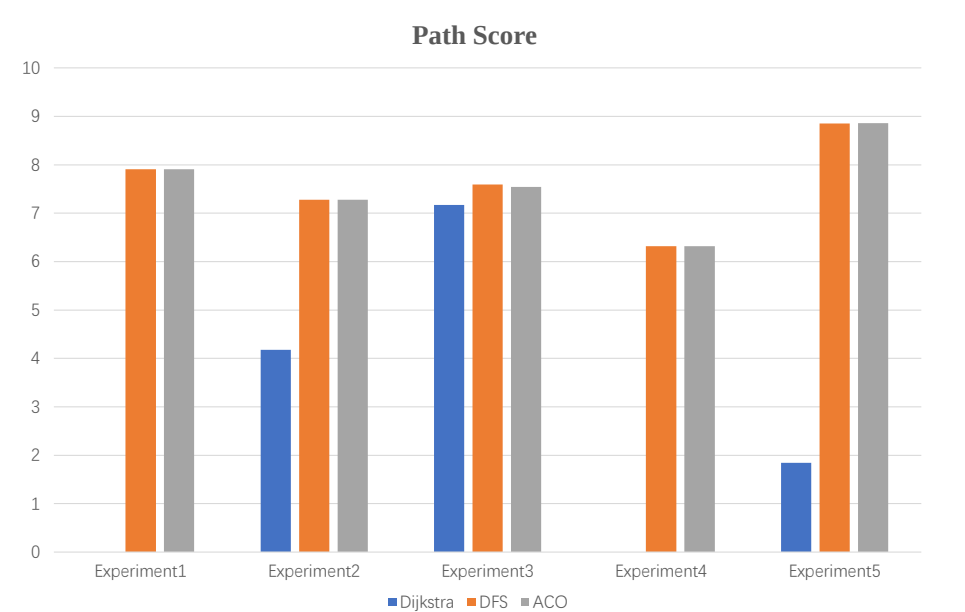
Table 7. Some examples of path recommend result

Perceive Information Accuracy	Organize Information Accuracy	Total Accuracy	Average Score
91.16 %	86.51 %	77.21 %	3.65

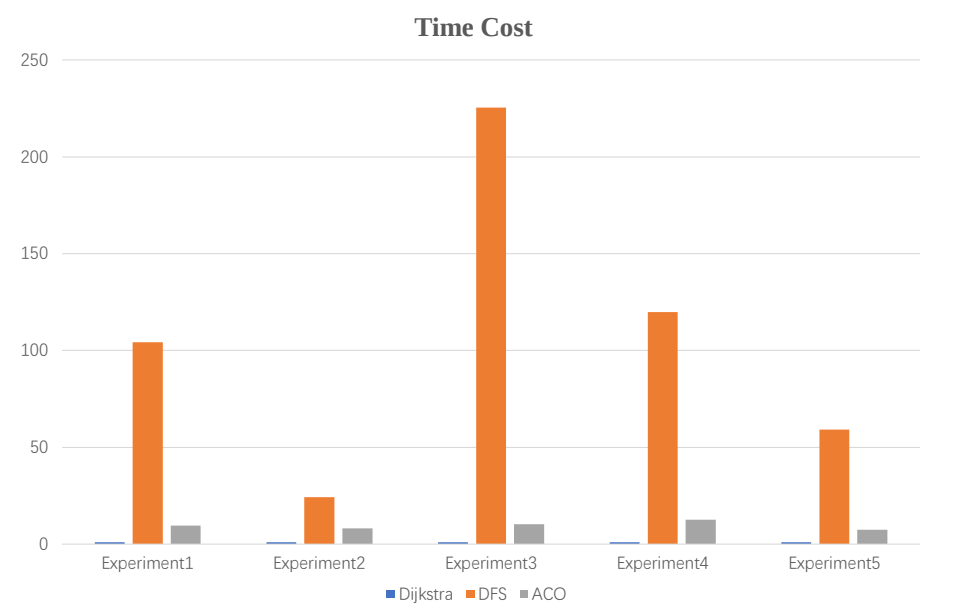
Table 8. Learning style accuracy

In the controlled experiments, we compared our proposed ACO algorithm with DFS and Dijkstra algorithms over 5 trials, as illustrated in Figure 5 a) and Figure 5 b). From Figure 5 a), it is evident that DFS and ACO achieve similar path scores, where higher scores indicate better alignment with learner profiles. In contrast, Dijkstra algorithm exhibits instability, failing to find paths in two out of the five experiments, and yielding lower scores in the remaining three due to generating overly lengthy paths leading to score degradation.

Figure 5 b) presents the time costs for these algorithms to find paths. Dijkstra generates paths in a single pass, allowing minimal time expenditure but with suboptimal effectiveness as it cannot iteratively update based on path evaluations. DFS, while capable of identifying the optimal path similar to ACO, incurs variable time costs due to its undirected search, necessitating repeated traversal of reachable paths to find higher-scoring alternatives. ACO consistently finds optimal paths within approximately 10 seconds, demonstrating significantly higher efficiency compared to DFS.

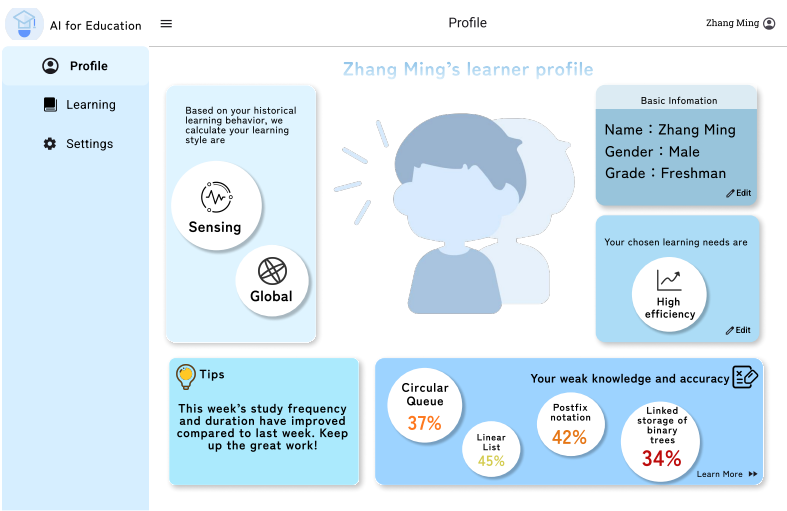


a) Path score of recommend experiments



b) Time cost of recommend experiments

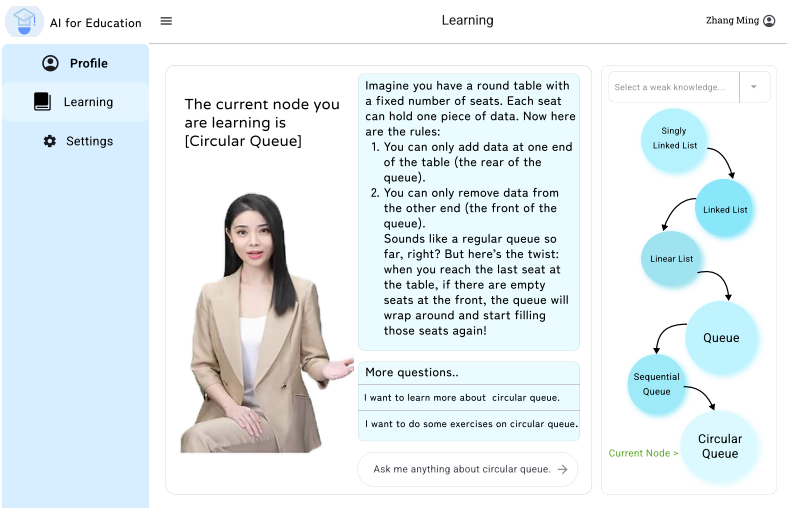
Figure 5. Recommend experiments result



a) Profile page of the system

5.3 System Presentation

Figure 6 shows the visualization pages of our research system. In the profile page, students can view the latest updates on their learner profile, including options to



b) Learning page of the system

Figure 6. Partial display of system pages

modify certain information. In the learning page, students can independently choose weak knowledge points to study. The system incorporates a digital human instructor to explain the knowledge points, and students can also ask questions.

6 CONCLUSION

This paper proposes a learning path recommendation algorithm based on student profiles, modeling profiles across learning style, learning needs, and knowledge mastery. It defines distinct evaluation criteria for different profiles and utilizes ACO on a subject knowledge graph to discover the highest scoring paths for recommendation. Our algorithm demonstrated superior comprehensive capabilities in experiments, receiving positive feedback from learners.

In the future, we will focus on finer-grained profile design, considering dynamic updates to profiles. Additionally, it aims to gather more learner data to enhance evaluation systems and system development. Exploring additional algorithms will be pursued to optimize recommendation success rates and efficiency.

REFERENCES

- [1] ZHANG, X.—LUO, H.—CHEN, B.—GUO, G.: Multi-View Visual Bayesian Personalized Ranking for Restaurant Recommendation. *Applied Intelligence*, Vol. 50, 2020, No. 9, pp. 2901–2915, doi: 10.1007/s10489-020-01703-6.
- [2] ALMAIAH, M. A.—AL-KHASAWNEH, A.—ALTHUNIBAT, A.: Exploring the Critical Challenges and Factors Influencing the E-Learning System Usage During COVID-19 Pandemic. *Education and Information Technologies*, Vol. 25, 2020, No. 6, pp. 5261–5280, doi: 10.1007/s10639-020-10219-y.
- [3] DHAIQUIR, I.—EZZIYYANI, M.—KHALDI, M.: The Personalization of Learners' Educational Paths E-Learning. In: Ben Ahmed, M., Teodorescu, H.N.L., Mazri, T., Subashini, P., Boudhir, A. A. (Eds.): *Networking, Intelligent Systems and Security*. Springer, Singapore, Smart Innovation, Systems and Technologies, Vol. 237, 2022, pp. 521–534, doi: 10.1007/978-981-16-3637-0_37.
- [4] EL-SABAGH, H. A.: Adaptive E-Learning Environment Based on Learning Styles and Its Impact on Development Students' Engagement. *International Journal of Educational Technology in Higher Education*, Vol. 18, 2021, No. 1, Art.No. 53, doi: 10.1186/s41239-021-00289-4.
- [5] TOMCZYK, L.—POTYRAŁA, K.—DEMESHKANT, N.—CZERWIEC, K.: University Teachers and Crisis E-Learning: Results of a Polish Pilot Study On: Attitudes Towards E-Learning, Experiences with E-Learning and Anticipation of Using E-Learning Solutions After the Pandemic. 2021 16th Iberian Conference on Information Systems and Technologies (CISTI), 2021, pp. 1–6, doi: 10.23919/CISTI52073.2021.9476521.
- [6] LIU, X.—DENG, M.—MENG, Q.—YAN, Y.: History and Pedagogy Mathematics in Mathematics Education: On the Causes of Learning Difficulties in Advanced

- Mathematics. 2020 International Conference on Modern Education and Information Management (ICMEIM), 2020, pp. 730–733, doi: 10.1109/ICMEIM51375.2020.00163.
- [7] SEGAL, A.—GAL, K.—SHANI, G.—SHAPIRA, B.: A Difficulty Ranking Approach to Personalization in E-Learning. *International Journal of Human-Computer Studies*, Vol. 130, 2019, pp. 261–272, doi: 10.1016/j.ijhcs.2019.07.002.
 - [8] LI, C. C.—DONG, Y.—HERRERA, F.: A Consensus Model for Large-Scale Linguistic Group Decision Making with a Feedback Recommendation Based on Clustered Personalized Individual Semantics and Opposing Consensus Groups. *IEEE Transactions on Fuzzy Systems*, Vol. 27, 2019, No. 2, pp. 221–233, doi: 10.1109/TFUZZ.2018.2857720.
 - [9] RAHMAN, M. M.—ABDULLAH, N. A.: A Personalised Group-Based Recommendation Approach for Web Search in E-Learning. *IEEE Access*, Vol. 6, 2018, pp. 34166–34178, doi: 10.1109/ACCESS.2018.2850376.
 - [10] LU, H.: Research on Personalized Course Selection Platform for College Students Based on Hybrid Recommendation. *C E Ca*, Vol. 42, 2017, pp. 1948–1952.
 - [11] JOSEPH, L.—ABRAHAM, S.—MANI, B. P.—RAJESH, N.: Exploring the Effectiveness of Learning Path Recommendation Based on Felder-Silverman Learning Style Model: A Learning Analytics Intervention Approach. *Journal of Educational Computing Research*, Vol. 60, 2022, No. 6, pp. 1464–1489, doi: 10.1177/07356331211057816.
 - [12] KLAŠNJA-MILIĆEVIĆ, A.—IVANOVIĆ, M.—VESIN, B.—BUDIMAC, Z.: Enhancing E-Learning Systems with Personalized Recommendation Based on Collaborative Tagging Techniques. *Applied Intelligence*, Vol. 48, 2018, No. 6, pp. 1519–1535, doi: 10.1007/s10489-017-1051-8.
 - [13] KAUR, R.—GUPTA, D.—MADHUKAR, M.—SINGH, A.—ABDELHAQ, M.—ALSAQOUR, R.—BREÑOSA, J.—GOYAL, N.: E-Learning Environment Based Intelligent Profiling System for Enhancing User Adaptation. *Electronics*, Vol. 11, 2022, No. 20, Art.No. 3354, doi: 10.3390/electronics11203354.
 - [14] ZHANG, X.—LI, M.—DEWEN, S.—CHEN, X.—CHEN, X.: A Novel Precise Personalized Learning Recommendation Model Regularized with Trust and Influence. *Scientific Programming*, Vol. 2022, 2022, Art.No. 8479423, doi: 10.1155/2022/8479423.
 - [15] WANG, F.—ZHANG, L.—CHEN, X.—WANG, Z.—XU, X.: A Personalized Self-learning System Based on Knowledge Graph and Differential Evolution Algorithm. *Concurrency and Computation: Practice and Experience*, Vol. 34, 2022, Art.No. e6190, doi: 10.1002/cpe.6190.
 - [16] LIN, Y.—ZHANG, W.—LIN, F.—ZENG, W.—ZHOU, X.—WU, P.: Knowledge-Aware Reasoning with Self-Supervised Reinforcement Learning for Explainable Recommendation in MOOCs. *Neural Computing and Applications*, Vol. 36, 2024, No. 8, pp. 4115–4132, doi: 10.1007/s00521-023-09257-7.
 - [17] ZHANG, Q.: Construction of Personalized Learning Platform Based on Collaborative Filtering Algorithm. *Wireless Communications and Mobile Computing*, Vol. 2022, 2022, Art.No. 5878344, doi: 10.1155/2022/5878344.
 - [18] LIU, X.: A Collaborative Filtering Recommendation Algorithm Based on the Influence Sets of E-Learning Group's Behavior. *Cluster Computing*, Vol. 22, 2019,

- No. Suppl 2, pp. 2823–2833, doi: 10.1007/s10586-017-1560-6.
- [19] LIU, D.—LIAN, J.—LIU, Z.—WANG, X.—SUN, G.—XIE, X.: Reinforced Anchor Knowledge Graph Generation for News Recommendation Reasoning. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD '21), 2021, pp. 1055–1065, doi: 10.1145/3447548.3467315.
 - [20] ZHU, H.—TIAN, F.—WU, K.—SHAH, N.—CHEN, Y.—NI, Y.—ZHANG, X.—CHAO, K. M.—ZHENG, Q.: A Multi-Constraint Learning Path Recommendation Algorithm Based on Knowledge Map. Knowledge-Based Systems, Vol. 143, 2017, pp. 102–114, doi: 10.1016/j.knosys.2017.12.011.
 - [21] SHI, D.—WANG, T.—XING, H.—XU, H.: A Learning Path Recommendation Model Based on a Multidimensional Knowledge Graph Framework for E-Learning. Knowledge-Based Systems, Vol. 195, 2020, Art.No. 105618, doi: 10.1016/j.knosys.2020.105618.
 - [22] LIANG, Z.—MU, L.—CHEN, J.—XIE, Q.: Graph Path Fusion and Reinforcement Reasoning for Recommendation in MOOCs. Education and Information Technologies, Vol. 28, 2023, No. 1, pp. 525–545, doi: 10.1007/s10639-022-11178-2.
 - [23] ZHOU, Y.—HUANG, C.—HU, Q.—ZHU, J.—TANG, Y.: Personalized Learning Full-Path Recommendation Model Based on LSTM Neural Networks. Information Sciences, Vol. 444, 2018, pp. 135–152, doi: 10.1016/j.ins.2018.02.053.
 - [24] YUN, Y.—DAI, H.—AN, R.—ZHANG, Y.—SHANG, X.: Doubly Constrained Offline Reinforcement Learning for Learning Path Recommendation. Knowledge-Based Systems, Vol. 284, 2024, Art.No. 111242, doi: 10.1016/j.knosys.2023.111242.
 - [25] HAN, J.—JO, J.—JI, H.—LIM, H.: A Collaborative Recommender System for Learning Courses Considering the Relevance of a Learner's Learning Skills. Cluster Computing, Vol. 19, 2016, No. 4, pp. 2273–2284, doi: 10.1007/s10586-016-0670-x.
 - [26] LANG, Y.—WANG, G.: Personalized Knowledge Point Recommendation System Based on Course Knowledge Graph. Journal of Physics: Conference Series, Vol. 1634, 2020, Art.No. 012073, doi: 10.1088/1742-6596/1634/1/012073.
 - [27] JIANG, B.—LI, X.—YANG, S.—KONG, Y.—CHENG, W.—HAO, C.—LIN, Q.: Data-Driven Personalized Learning Path Planning Based on Cognitive Diagnostic Assessments in MOOCs. Applied Sciences, Vol. 12, 2022, No. 8, Art.No. 3982, doi: 10.3390/app12083982.
 - [28] XIE, H.—ZOU, D.—WANG, F. L.—WONG, T. L.—RAO, Y.—WANG, H.: Discover Learning Path for Group Users: A Profile-Based Approach. Neurocomputing, Vol. 254, 2017, pp. 59–70, doi: 10.1016/j.neucom.2016.08.133.
 - [29] MA, D.—ZHU, H.—LIAO, S.—CHEN, Y.—LIU, J.—TIAN, F.—CHEN, P.: Learning Path Recommendation with Multi-Behavior User Modeling and Cascading Deep Q Networks. Knowledge-Based Systems, Vol. 294, 2024, Art.No. 111743, doi: 10.1016/j.knosys.2024.111743.
 - [30] SCHWAB, M.—STROBELT, H.—TOMPKIN, J.—FREDERICKS, C.—HUFF, C.—HIGGINS, D.—STREZHNEV, A.—KOMISARCHIK, M.—KING, G.—PFISTER, H.: booc.io: An Education System with Hierarchical Concept Maps and Dynamic Non-Linear Learning Plans. IEEE Transactions on Visualization and Computer Graphics, Vol. 23, 2017, No. 1, pp. 571–580, doi: 10.1109/TVCG.2016.2598518.

- [31] ZHANG, H.—SHEN, X.—YI, B.—WANG, W.—FENG, Y.: KGAN: Knowledge Grouping Aggregation Network for Course Recommendation in MOOCs. *Expert Systems with Applications*, Vol. 211, 2023, Art.No. 118344, doi: 10.1016/j.eswa.2022.118344.
- [32] DIAO, X.—ZENG, Q.—LI, L.—DUAN, H.—ZHAO, H.—SONG, Z.: Personalized Learning Path Recommendation Based on Weak Concept Mining. *Mobile Information Systems*, Vol. 2022, 2022, Art.No. 2944268, doi: 10.1155/2022/2944268.
- [33] SAITO, T.—WATANOBE, Y.: Learning Path Recommender System Based on Recurrent Neural Network. 2018 9th International Conference on Awareness Science and Technology (iCAST), IEEE, 2018, pp. 324–329, doi: 10.1109/ICAwST.2018.8517231.
- [34] XU, G.—WONG, C. U. I.: Deep Learning-Based Educational Image Content Understanding and Personalized Learning Path Recommendation. *Traitement du Signal*, Vol. 41, 2024, No. 1, pp. 459–467, doi: 10.18280/ts.410140.
- [35] VAN ZWANENBERG, N.—WILKINSON, L. J.—ANDERSON, A.: Felder and Silverman's Index of Learning Styles and Honey and Mumford's Learning Styles Questionnaire: How Do They Compare and Do They Predict Academic Performance? *Educational Psychology*, Vol. 20, 2000, No. 3, pp. 365–380, doi: 10.1080/713663743.
- [36] JIANG, Q.—ZHAO, W.—DU, X.: Study on the Users Learning Style Model of Correction Under Felder-Silverman Questionnaire. *Modern Distance Education*, Vol. 27, 2010, No. 1, pp. 62–66, <https://api.semanticscholar.org/CorpusID:148386202> (in Chinese).



Xin XIE is a Master's student at the School of Computer Science and Technology, Donghua University. Her research interests include knowledge graphs and personalized recommendation.



Xiangyang FENG is Associate Professor and Master's Supervisor in the Department of Computer Applications, School of Computer Science and Technology, Donghua University. He received his Ph.D. in intelligent information systems from the Saga University, Japan, in 2003. His research interests include knowledge graphs, digital supply chains, blockchain, e-commerce, human-computer interaction, and intelligent computer education.