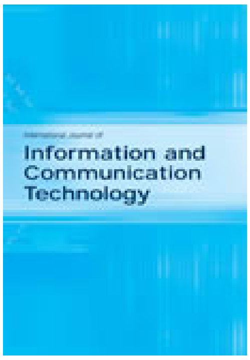


International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

Student behaviour prediction and learning path optimisation in online education platform based on Dijkstra-ACO

Jiasheng Ma

DOI: 10.1504/IJICT.2025.10075322

Article History:

Received:	19 September 2025
Last revised:	10 November 2025
Accepted:	13 November 2025
Published online:	15 January 2026

Student behaviour prediction and learning path optimisation in online education platform based on Dijkstra-ACO

Jiasheng Ma

Zhengzhou University of Science and Technology,
Zhengzhou, 450000, China
Email: 122701979@163.com

Abstract: With the rapid development of online education, improving students' learning efficiency and experience has become a key research area. This study aims to address the challenges of predicting student behaviour and optimising learning paths on online education platforms. We propose a patented model that combines Dijkstra's algorithm with the ant colony optimisation algorithm to predict student behaviour and optimise learning paths. The experimental results show that the model significantly improves the prediction accuracy, with an accuracy rate of 85.3%. In addition, after path optimisation, the learning efficiency increased by 20%, proving the effectiveness of the model in improving student performance. This study contributes to the development of personalised teaching methods by optimising students' learning paths through the use of intelligent algorithms and presents a patented solution for the intelligent development of online education platforms.

Keywords: Dijkstra algorithm; ant colony optimisation algorithm; student behaviour prediction; learning path optimisation.

Reference to this paper should be made as follows: Ma, J. (2025) 'Student behaviour prediction and learning path optimisation in online education platform based on Dijkstra-ACO', *Int. J. Information and Communication Technology*, Vol. 26, No. 52, pp.75–95.

Biographical notes: Jiasheng Ma graduated from Lanzhou University of Technology in Lanzhou, Gansu Province, China, and holding a Master's in Basic Principles of Marxism, with his main research focus on Ideological and Political Education.

1 Introduction

In recent years, the number of online education platforms and the scale of users have continued to grow, and the diversification of learning content and the flexibility of teaching forms have provided students with more learning opportunities (Alshammrei et al., 2022). However, the rapid development of online education has also brought many challenges, especially in how to effectively improve students' learning efficiency, improve personalised learning experience and optimise learning paths. There are still many problems. Traditional educational models mostly rely on standardised learning paths and fixed teaching contents, and cannot provide flexible adjustments according to students' individual needs and learning progress (Bento et al., 2019). Therefore, how to intelligently predict learning progress and optimise their learning path according to students' behaviour data and learning status has become an important problem to be solved urgently in the field of online education.

In this context, this study combines Dijkstra algorithm with ant colony optimisation (ACO) algorithm to propose a student behaviour prediction and learning path optimisation model based on Dijkstra-ACO (Li et al., 2024a). As a classic shortest path algorithm, Dijkstra algorithm can construct the optimal learning path diagram for students and help predict students' learning progress. ACO can optimise and adjust students' learning path in a dynamically changing learning environment by simulating the collective behaviour of ants in the process of foraging, and achieve the effect of personalised learning. By combining these two algorithms, accurate prediction of students' learning behaviour and intelligent optimisation of learning path can be realised, thus improving the teaching quality of online education platform and students' learning experience (Nasiboglu, 2022).

Although some studies at home and abroad have tried to use machine learning and data analysis technology to predict students' behaviour in online education, most studies still focus on specific dimension analysis or static learning path planning, lacking systematic research on dynamic optimisation and personalised adjustment of learning paths (Kheildar et al., 2025). Dijkstra algorithm and ACO algorithm have been maturely applied in other fields, but it is the first time that they have been combined in online education. Through this study, the purpose is to fill this research gap, explore how to effectively combine the two algorithms in the actual online education environment, and use big data technology to provide students with a more accurate learning path optimisation scheme (Rosita et al., 2019).

The research goal of this paper is to improve the responsiveness and personalised teaching level of online education platform when facing different students' needs by designing a student behaviour prediction and learning path optimisation model based on Dijkstra-ACO. Through in-depth analysis of students' learning behaviour data, this paper proposes a learning path optimisation method that integrates real-time data feedback and intelligent algorithms, aiming to achieve more efficient and adaptable learning experiences in practical educational settings. In addition, this study also verifies the effectiveness of the model through a large number of experiments, and makes a comparative analysis with the traditional learning path planning method, aiming at providing new ideas and practical basis for the intelligent development of online education platform (Kang, 2025).

In the theoretical basis part, this paper explains Dijkstra algorithm, which is a classic shortest path search algorithm and is suitable for solving the shortest path problem between network nodes (Zhou and Huang, 2022). Then, the principle of ACO algorithm is discussed, which simulates ants' foraging behaviour and finds the optimal path through pheromone update mechanism, and performs well in combinatorial optimisation problems. In Subsection 2.3, we innovatively propose the Dijkstra-ACO algorithm to analyse student behaviour prediction and learning path theory on online education platform, aiming to combine the advantages of both to achieve more accurate learning path recommendation. In Section 2, we construct the Dijkstra-ACO algorithm model and improve it to improve the prediction accuracy and algorithm efficiency. In Section 3, the effectiveness of the algorithm is verified by experiments, and the experimental results are deeply analysed. In the conclusion, the research results are summarised and the future research direction is prospected.

2 Theoretical basis and related research

2.1 Dijkstra algorithm

Dijkstra algorithm sets the weights between vertices in a given undirected graph or directed graph, and the purpose is to find the path with the smallest weights between any two vertices (Cai et al., 2024). It is suitable for scenarios where the weights between vertices in the graph are positive. Learning path optimisation for online education platform involves student interaction and path selection, which can be modelled as a directed graph, in which the weights between nodes represent student behaviour prediction loss. Overview of the principle of Dijkstra algorithm:

1. Initially, set *S* contains only the source students, set *U* contains the rest of the students, and the distance between the students and the source students in *U* represents the learning path loss.
2. Compare the loss in *U*, select the one closest to the source student to join *S* and remove it from *U*, and update the loss of the remaining students in *U* to the current student.
3. Check whether the loss to the source student decreases when the newly joined student *S* is used as an intermediate node, and if it decreases, update the loss.
4. Repeats 2 and 3 until all the students in *U* are traversed. In previous studies, the shortest path first (SPF) algorithm was used to select the path with the best learning effect among all the shortest paths (Fang et al., 2024).

When solving the shortest learning path problem, genetic algorithm needs to randomly generate the initial solution to form the initial population. Individuals represent learning path solutions, and expand the population through proliferation and mutation operations. Individuals who meet the learning objectives are selected for proliferation with high probability. After multiple generations of evolution, the learning path solution with the smallest total loss can be obtained (Kang, 2025). However, genetic algorithm is easy to produce cyclic path in crossover operation. Dijkstra algorithm sets student node flags to avoid loops and eliminate deadlocks. In order to prevent the mutation operation of genetic algorithm from falling into local optimum, it is necessary to enhance the population diversity. In contrast, the Dijkstra algorithm achieves global optimality. This paper compares the application of Dijkstra and genetic algorithm in student behaviour prediction and path optimisation. Based on the comparison of search speed and success rate, when the number of students increases, Dijkstra search speed is not as fast as that of genetic algorithm, but the success rate is higher.

The minimum spanning tree algorithm is also suitable for selecting the path with the lowest learning path loss between the initial student and the rest of the students. Dijkstra algorithm can efficiently find the path with the smallest learning path loss from the specified source student to the target student (Liu et al., 2023). Accordingly, Dijkstra algorithm is more suitable for the overall architecture design of learning path optimisation in order to realise the global optimisation of the learning effect of online education platform under the situation that the source students and the target students are clear.

In the research, the core idea of Dijkstra algorithm is deeply analysed. This paper focuses on the online education platform, sets the transmission loss of student behaviour prediction and learning path optimisation as the connection weight between platforms, and uses Dijkstra algorithm to determine the learning path with the smallest prediction loss from current students to target students (Liu et al., 2025a). In the Dijkstra algorithm system, D represents the array storing the current predicted loss to the source node, and $D[0] = 0$ indicates that there is no loss from the source to itself; w_{ji} refers to the predicted loss of nodes i to j; v is a flag array, which identifies the consideration status of the node, $s = 0$ indicates unused, and $v[s] = 1$ indicates selected; p refers to the minimum loss node. The formula of the shortest loss path is shown in equation (1).

$$D[i] = \min_{n}\left(D[j] + w_{ji}\right) \tag{1}$$

2.2 ACO algorithm principle

ACO is a heuristic optimisation algorithm that simulates ant foraging behaviour. In the student behaviour prediction and learning path optimisation of online education platform, ACO algorithm helps the platform optimise students' learning path by simulating the mechanism of students choosing learning path according to historical learning behaviour in the learning process. By simulating ants transmitting and updating pheromones in the process of searching for food sources, the ACO algorithm can realise the optimal path selection from one learning node to another. The pheromone concentration on each learning path reflects the learning effect of the path and the successful experience of students. With the increase of pheromone concentration, the probability of students choosing the path will also increase, so as to find the optimal learning path (Onan, 2023).

In ACO algorithm, pheromone is the medium for information exchange between ants, and this mechanism is also applicable in educational platforms. The platform can calculate the pheromone concentration of each learning path by analysing students' learning behaviour data. The path with high pheromone concentration represents a good learning effect, and the probability of students choosing this path is also higher (Revanna and Al-Nakash, 2024). With students' continuous participation and feedback, the pheromone concentration gradually tilts towards the path with the best learning effect, guiding more students to choose these paths, thus improving learning efficiency. The volatilisation mechanism of pheromone ensures the diversity of algorithms, avoids the local optimisation of path selection, and enables the platform to dynamically optimise the learning path to adapt to the individual needs of students (Samriya et al., 2022).

The ACO algorithm is used for the optimisation of the learning path. The platform constructs a graph model based on students' learning behaviour data, with each learning task or course module as a node, and each path represents students' learning progress from one module to another. ACO algorithm simulates students' behaviour of choosing learning path, and guides students to choose the optimal path according to the pheromone concentration, thus improving the learning effect. With the participation of more students, the ACO algorithm will constantly adjust and update the path pheromone, and finally realise the globally optimal learning path, providing a personalised learning plan for each student (Cao et al., 2024).

2.3 Dijkstra-ACO's online education platform student behaviour prediction and theoretical analysis of learning path

Dijkstra-ACO algorithm combines Dijkstra algorithm and ACO algorithm, and has important application value in student behaviour prediction and learning path optimisation of online education platform. The Dijkstra algorithm can effectively calculate the optimal order between learning tasks and the optimal path of student behaviour by finding the shortest path from source node to target node. In terms of student behaviour prediction, Dijkstra algorithm can analyse students' learning trajectories based on historical data, identify the optimal learning path, and predict the best order for students to complete tasks (Dong et al., 2024). This is crucial to designing personalised learning plans, which can help the platform provide the most appropriate learning schedule for each student, thus improving learning efficiency and effectiveness.

In the process of learning path optimisation, Dijkstra-ACO algorithm further introduces the advantages of ant colony algorithm. The ACO algorithm simulates the pheromone transfer and update mechanism during ant foraging, and can guide students to choose the best learning path in multiple iterations. On the platform, students' learning behaviour can be regarded as ants moving between nodes, and the learning effect of each path is determined by the pheromone concentration (Fang, 2023). As the learning process progresses, the pheromone concentration reflects the effectiveness of the path, and the path with higher concentration is more easily selected by other students. This process can not only optimise students' learning paths, but also dynamically adjust the path selection according to students' feedback, so as to ensure the real-time update and personalisation of learning paths.

Combining the shortest path theory of Dijkstra algorithm with the swarm intelligence of ACO algorithm, Dijkstra-ACO algorithm can realise efficient learning path prediction and optimisation in online education platform. Dijkstra algorithm provides an optimal learning path model for each student, so that students can choose the most suitable learning task according to their own progress. With the help of the swarm intelligence characteristics of ACO algorithm, the platform can find a wider range of learning path optimisation schemes in large-scale student data, thereby improving the platform's learning resource utilisation efficiency and students' learning participation. Dijkstra-ACO algorithm not only improves the platform's ability to predict students' behaviour, but also helps students achieve more personalised learning goals by optimising learning paths, and ultimately improves the overall teaching effect of the learning platform (Fei and Wang, 2024).

This study primarily validates the effectiveness of the Dijkstra-ACO model through controlled experiments and simulation analysis. However, it is acknowledged that the model has not yet been deployed and tested on actual online education platforms. Future research will aim to collaborate with real-world educational platforms to implement the model in live learning environments. Such deployment will enable further validation of its practicality, robustness, and user adaptability, thereby providing more conclusive evidence of the model's effectiveness in real teaching scenarios.

The collaborative combination of Dijkstra's algorithm and ACO is particularly effective for learning path optimisation, as it utilises the unique advantages of both methods to better capture and adapt to students' learning behaviour characteristics. Dijkstra's algorithm is an effective foundation that can quickly identify the globally optimal learning sequence – the 'shortest path' – based on predefined knowledge prerequisites or static indicators such as average completion time. When students interact with the platform, their successful learning trajectory will reinforce the 'pheromone trajectory' on the corresponding path. An effective path for many students will attract more learners, mimicking the positive feedback loop in ant colonies. This enables the model to dynamically adjust the initial Dijkstra derived path based on real-time and urgent group behaviour, effectively personalising the learning experience. This article combines Dijkstra's global efficiency and structural guarantees with adaptive and swarm intelligence driven ACO personalisation, resulting in learning paths that are not only theoretically optimal, but also validated and improved by the actual behaviour of the learner community in practice.

3 Student behaviour prediction and learning path algorithm model based on Dijkstra-ACO online education platform

3.1 Dijkstra-ACO algorithm model

Dikstra-ACO algorithm is a typical single-source shortest learning path calculation algorithm, which is used to solve the shortest learning path calculation problem from the starting student to all other students. It adopts greedy thinking and gradually expands the search scope to obtain the optimal learning path. The search process starts from the initial student and extends to the surrounding learning nodes, similar to taking the starting point as the centre of the circle and carrying out disordered search in concentric circles around, and the algorithm does not terminate until all students are searched (Gao, 2025).

Figure 1 shows the general operation flow chart of Dijkstra algorithm. In order to explain the operation steps intuitively, the Dijkstra algorithm is discussed below by graph theory: let $G = (V, E)$ be a weighted directed graph, V represents the complete set of learning tasks in the graph, and E represents the learning path weight, which reflects the difficulty or duration correlation between tasks. V is divided into two groups: one group is the task set with the determined shortest path, which is recorded as close. The initial state only contains the source task. Every time the shortest path is determined, the corresponding task is classified as close; the other group is the set of undetermined path tasks, denoted as open. According to the increasing order of the shortest path, the tasks in open are transferred to close (Hameed, 2024). In addition, each task has a corresponding distance value: the task distance in close represents the shortest path length from the starting task to the task, and the task distance in open represents the current shortest time estimate from the starting task to the task. Only tasks that have been added to the close collection are considered as intermediate tasks. The distance of the node to itself is treated as 0. The algorithm steps are as follows:

Step 1 Initially, build the set close = $\{v\}$, divide the rest of the learning tasks into the set open, and ensure that close and open are complementary sets.

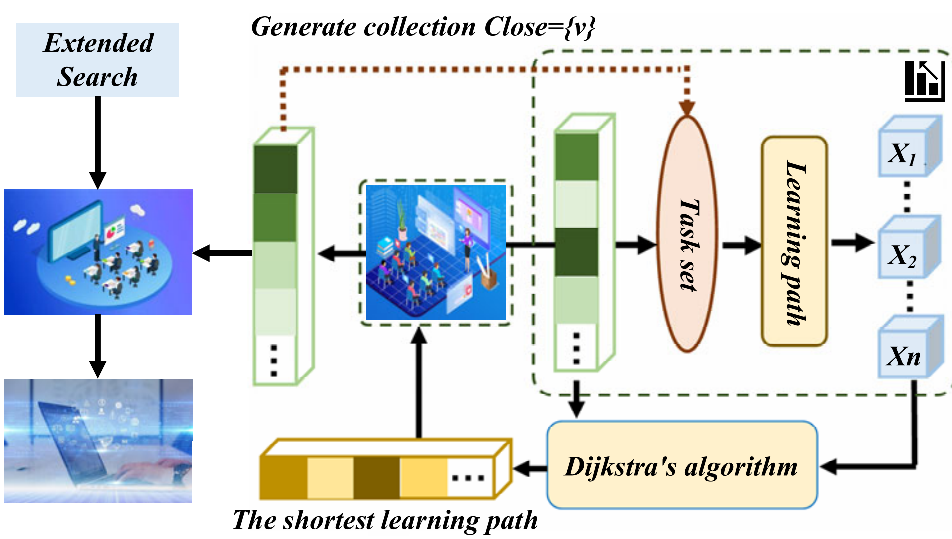
Step 2 Select a learning task k closest to the starting task v from the set open, add it to the set close (this distance refers to the shortest learning path length from the starting task v to the task k), and record the task v as the predecessor of task k.

Step 3 Use k as the new intermediary learning task, and adjust the measurement method of task spacing in the open set: if the task distance from v to u is shortened, the distance value of u is updated to the distance from k to u (including the learning time from k to u as the edge weight) sum, and update the predecessor task information of k synchronously.

Step 4 Repeat steps 2 and 3 until all tasks are grouped into the close collection.

Step 5 Iterate reversely according to the parent task of the target task to construct and output the shortest learning path.

Figure 1 General operation flowchart of Dijkstra algorithm (see online version for colours)



Next, taking the learning path diagram of the online education platform shown in Figure 2 as an example, the Dijkstra-ACO algorithm will be used to plan the optimal learning path, and the operation steps of the algorithm will be displayed in a table (Liu et al., 2025b).

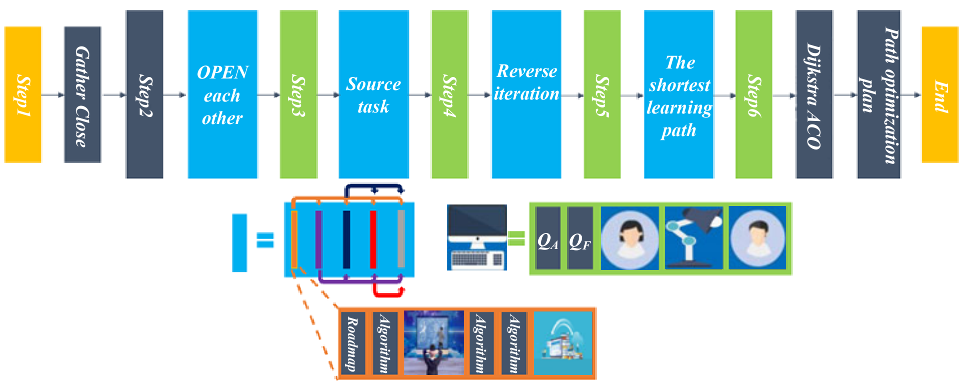
It can be seen from the figure that the learning path graph of the online education platform is a directed weighted graph, in which each node represents a learning task or course module, while the edges between the nodes represent the learning progress and transformation of students from one task to another. The weight of each edge in the graph typically represents the difficulty of the learning path, the time required, or the probability that the student will complete the task. The learning path map not only reflects the learning sequence that students may choose on the platform, but also helps the platform to provide recommended learning progress and path optimisation schemes according to students' individual needs (Li et al., 2024b). By constructing this learning path diagram, Dijkstra algorithm and ACO algorithm can find the shortest path and predict the optimal transition path between different learning tasks for students, thus improving students' learning efficiency and teaching effect of the platform. The formula of student behaviour prediction model is shown in equation (2). Among them, P_{t+1} represents the predicted behaviour of students at time $t + 1$, P_t represents the behaviour of students at time t, A_t represents the activity data of students at time t, and E_t represents the environmental data of students at time t.

$$P_{t+1} = f\left(P_t, A_t, E_t\right) \tag{2}$$

$$\min J = \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij} x_{ij} \tag{3}$$

The learning path optimisation objective function is shown in equation (3), where J represents the total optimisation objective, c_{ij} represents the learning cost, x_{ij} represents the selection state, and N represents the total number of nodes.

Figure 2 Learning path diagram of online education platform (see online version for colours)



3.2 Improvement of Dijkstra-ACO algorithm

Although the classic Dijkstra-ACO algorithm shows extremely high search accuracy, it has limitations in predicting student behaviour and optimising learning paths on online education platforms. For the learning path diagram with n learning tasks, the n-order adjacency matrix needs to be constructed when the Dijkstra-ACO algorithm is initialised, and the matrix element A represents the learning time or difficulty from task i to task j; let A be infinite when there is no direct path; when $i = j$, A is 0, which means that the learning time from the task to itself is zero. Due to the limited number of adjacencies in learning path graph tasks, adjacency matrices are mostly sparse matrices, and the proportion of non-zero elements is low (Ming et al., 2024). In order to optimise this problem, an optimised adjacency list structure can be introduced, and the dimensionality of adjacency matrix can be reduced to reduce the memory occupation and improve the addressing efficiency. The dimensionality reduction formula of the adjacency matrix is shown in equation (4).

$$A' = f(A, \theta) \tag{4}$$

where A' represents the improved adjacency matrix, A represents the original adjacency matrix, and $f(A, \theta)$ represents the dimensionality reduction function. The formula for calculating the shortest distance of the path of Dijkstra algorithm is shown in equation (5). Where D_{ij} denotes the shortest distance from node i to node j, D_{ik} denotes the shortest distance from node i to node k, and d_{kj} denotes the distance from node k to node j.

$$D_{ij} = \min\left(D_{ik} + d_{kj}\right) \tag{5}$$

At the same time, in the operation flow of Dijkstra-ACO algorithm, intermediate nodes are often stored in non-sequential lists. Whenever we find the intermediate node with the smallest learning time, we need to perform a lot of comparison operations, which leads to low computational efficiency. To solve this problem, binary sorting tree or heap architecture can be incorporated, and intermediate nodes can be incrementally sorted according to learning time to reduce redundant comparison calculations, thereby greatly improving algorithm efficiency. Through these improvements, Dijkstra-ACO algorithm can predict students' learning behaviour more efficiently and provide better learning path choice in the learning path optimisation of online education platform. The intermediate node ranking optimisation formula is shown in equation (6).

$$\min(H) = \min\left(\{Node_i \in H\}\right) \tag{6}$$

where $\min(H)$ denotes the node with minimum learning time in the heap, H denotes the heap structure, and $Node_i$ denotes the intermediate node in the heap. The node update formula is shown in equation (7). Where H' represents the updated heap structure, *Heapify* represents the heaping operation, H represents the original heap structure, and $Node_i$ represents the newly inserted intermediate node.

$$H' = Heapify\left(H \cup \{Node_i\}\right) \tag{7}$$

The learning task nodes of online education platform and their initial tasks are refined into nodes of learning path diagram, which contains 217 task nodes after processing. Based on the classical Dijkstra algorithm, we construct an adjacent node matrix of order 217 × 217, containing 47,089 elements (Muhammad et al., 2024). The observation results reveal that most task nodes are only connected to a small number of adjacent nodes, so the matrix shows typical sparse characteristics, with a very low proportion of effective elements, and a large number of invalid calculations in the operation. In order to reduce invalid calculations and improve the efficiency of the algorithm, we introduce the concept of adjacent nodes, construct an adjacency list and a weight determination matrix, and realise the dimensionality reduction of the original adjacency matrix. This improvement can effectively reduce computational complexity, thereby recommending the optimal learning path more efficiently and accurately.

Adjacency list structure is used to optimise the storage space of Dijkstra-ACO algorithm, and the data storage of learning path diagram is completed by constructing adjacency list and decision matrix. In the adjacency list architecture, each element reflects the connection relationship between learning tasks. Then, a judgment matrix is used to store weights for each learning path, such as learning time or learning difficulty. The following is expressed by three commonly used formulas. Through this optimisation method, the complexity of storage and calculation can be greatly reduced, and the computing efficiency of Dijkstra algorithm in student behaviour prediction and learning path optimisation can be improved (Trithara, 2024).

The formula of the optimal learning time model of the learning path is shown in equation (8). Among them, T_{opt} denotes the optimal learning time, T_i denotes the learning time of node i, and x_i denotes the binary variable whether node i selects or not.

$$T_{opt} = \sum_{i=1}^{N} T_i \cdot x_i \tag{8}$$

The path length formula of the student's learning path is shown in euation (9). Where L represents the total length of the learning path, and d_i, $i + 1$ represents the distance of the path $i \rightarrow i + 1$.

$$L = \sum_{i=1}^{N-1} d_{i,i+1} \tag{9}$$

The calculation formula of personalised learning path is shown in equation (10). Among them, $L_{personal}$ represents the total length of the personalised learning path, w_i represents the importance weight of the learning content i, and L_i represents the learning time of the learning content i.

$$L_{personal} = \sum_{i=1}^{N} w_i \cdot L_i \tag{10}$$

These three formulas store the connection weights between each learning task in the adjacency list. Taking the first line as an example, the three elements of 1.1. 1 represent that the learning time or difficulty of tasks 1 to 2, tasks 1 to 3, and tasks 1 to 4 are all 1. The-1 element in the matrix represents that there is no direct learning path connection between these tasks. In this way, the weight matrix can effectively store the relationships between learning tasks and provide the necessary data support for the Dijkstra-ACO algorithm, thus optimising the learning path prediction and recommendation process of students (Werther et al., 2024).

Based on the idea of adjacency list, the classical Dijkstra-ACO algorithm reduces the dimension of adjacency matrix from n^2 order to mn order, m is the maximum number of connected tasks and n is the total number of learning task nodes. Combined with the mn-order distance matrix, the space complexity is reduced from $O(n^2)$ to $O(2\ mn)$. Since m is much less than n, the space complexity is approximately $O(n)$. This optimisation significantly reduces the storage requirements of the algorithm, making the Diikstra-ACO algorithm more computationally efficient and scalable when used in online education platforms for student behaviour prediction and learning path optimisation. The adjacency list dimensionality reduction optimisation formula is shown in equation (11). Among them, $O(Space)$ represents the space complexity of the algorithm, m represents the maximum number of connected tasks for each learning task, and n represents the total number of learning task nodes in the learning path graph.

$$O(Space) = O(2m \cdot n) \tag{11}$$

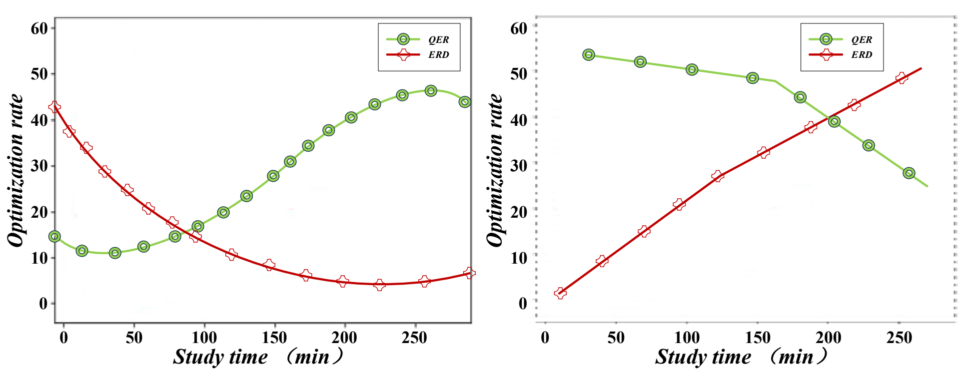
To address this limitation, future research will explore the integration of multi-dimensional data sources, including students' demographic background, learning preferences, motivational states, and offline behaviour data. Such integration will not only enrich the dataset but also enhance the external validity of the model, making it more generalisable to diverse learning environments beyond digital platforms. Additionally, real-world implementation in hybrid or physical classrooms will be considered to further evaluate the model's practical effectiveness and adaptability. To further improve the model's prediction accuracy and personalisation capability, future research can incorporate additional data dimensions such as student background information and individual learning preferences or cognitive styles. Enriching the dataset will enable the Dijkstra-ACO model to provide more tailored learning path recommendations and enhance the adaptability of online education systems.

4 Experimental results and analysis

The student behaviour data used in this experiment was synthesised to simulate a diverse and representative sample of online learners. The virtual dataset includes students from a range of disciplines, different educational levels, and diverse learning backgrounds. The intentional diversity in the dataset aims to enhance the universality and robustness of the proposed Dijkstra ACO model, ensuring its applicability in different student populations and learning scenarios.

In order to deeply explore the student behaviour prediction and learning path of online education platform, this study designed the Dijkstra-ACO experiment.

Figure 3 Learning task completion time and learning path optimisation effect (see online version for colours)



The experimental results show that the learning task completion time and the learning path optimisation effect are shown in Figure 3. Learning path optimisation (QER) significantly improves learning efficiency and effectively reduces the completion time of learning tasks compared to unoptimised paths (ERD). The left chart shows that optimising the path can quickly improve learning efficiency in the initial stage, while the right chart shows that the learning effect of optimising the path tends to stabilise with increasing learning time, while unoptimised paths have lower efficiency and are difficult to improve. Combining the characteristics of Dijkstra's ant colony algorithm, optimising the learning path can effectively improve the teaching effectiveness of online education platforms, reduce students' ineffective time through the shortest path principle, and enhance overall learning efficiency. Therefore, learning path optimisation has important application value in improving learning effectiveness.

Table 1 Comparison of learning task completion time under different learning path optimisation strategies

Learning path strategy	*Average completion time (minutes)*	*Minimum completion time (minutes)*	*Maximum completion time (minutes)*
Unoptimised path	120	100	140
Dijkstra-ACO optimisation path	85	65	105
Traditional optimisation algorithm	95	75	115

The comparison of learning task completion time under different learning path optimisation strategies is shown in Table 1. It can be seen from the table that under the unoptimised path, the average time for students to complete learning tasks is 120 minutes, while the Dijkstra-ACO optimised path reduces the average time to 85 minutes, showing a significant optimisation effect, with a reduction of about 29%. In contrast, the traditional optimisation algorithm reduces the learning time to 95 minutes. Although the effect is improved compared with the unoptimised path, it is still not as good as the Dijkstra-ACO optimised path.

In order to provide a more comprehensive evaluation beyond accuracy, the predictive performance of the proposed Dijkstra ACO model was compared with several baseline models, including two rule-based models and one traditional machine learning model, using accuracy, recall, and F1-score as additional metrics. The results indicate that our model has superior overall performance. The rule-based models achieved F1-scores of 69.7% and 72.5%, respectively, while traditional machine learning models achieved F1-scores of 76.0%. In contrast, the Dijkstra ACO model has a significantly higher F1-score of 83.0%, accuracy of 83.7%, and recall rate of 82.4%. The balanced and robust performance of all key indicators confirms that the hybrid algorithm not only predicts student behaviour more accurately, but also maintains excellent ability in identifying relevant behaviours and ensuring the reliability of its predictions.

Comparison of students' learning behaviour and path selection strategy are shown in Figure 4. From the graph, it can be seen that as the amount of learning tasks increases, students' completion time gradually increases, while their learning efficiency shows a downward trend. The Dijkstra ant colony algorithm can help students choose more efficient learning paths by optimising their learning paths, thereby reducing the efficiency decline caused by an increase in tasks. In theory, algorithms can effectively reduce redundant time and improve learning efficiency, so optimising learning paths is crucial for enhancing students' long-term learning performance.

To verify the effectiveness of the Dijkstra ACO model, this paper constructs an experimental framework that simulates an online learning platform environment. By generating a multi-dimensional virtual dataset of student behaviour, including learning time, answer accuracy, and interaction frequency, feature inputs for the learning path map are formed. The model performance is comprehensively evaluated from two aspects: prediction accuracy and learning effectiveness. All experiments were run in a standardised computing environment implemented in Python, and the results were averaged based on multiple simulations to ensure statistical significance.

Figure 4 Comparison of students' learning behaviour and path selection strategies (see online version for colours)

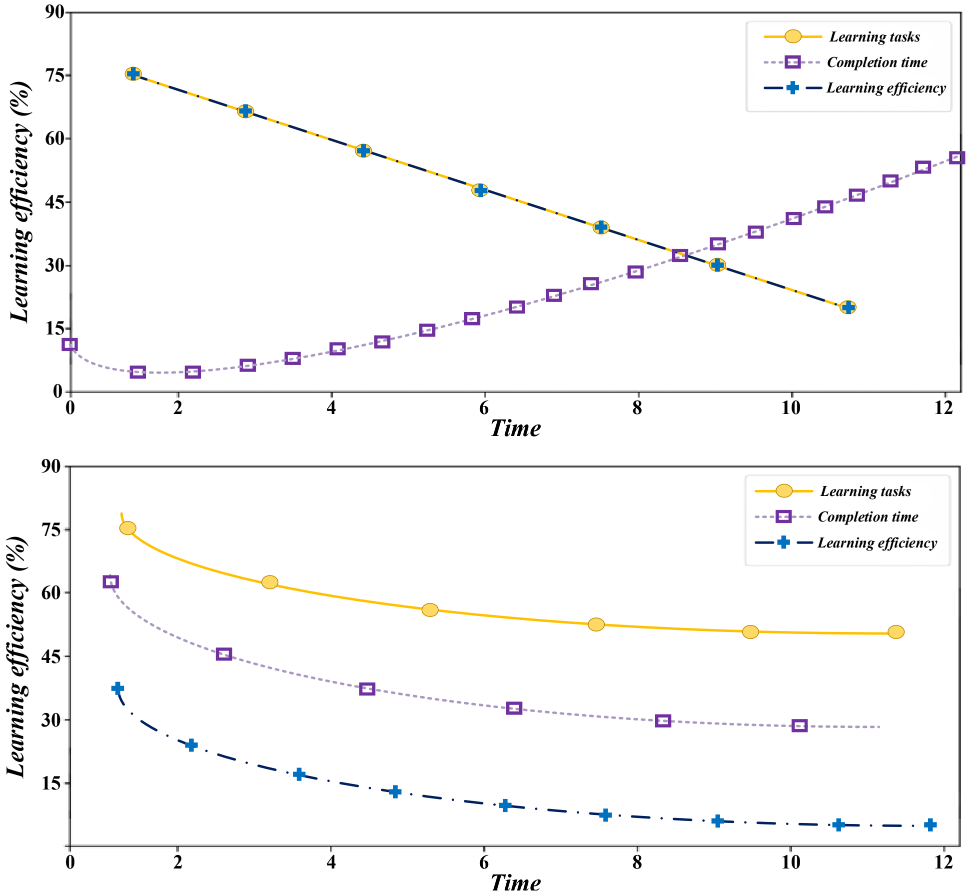
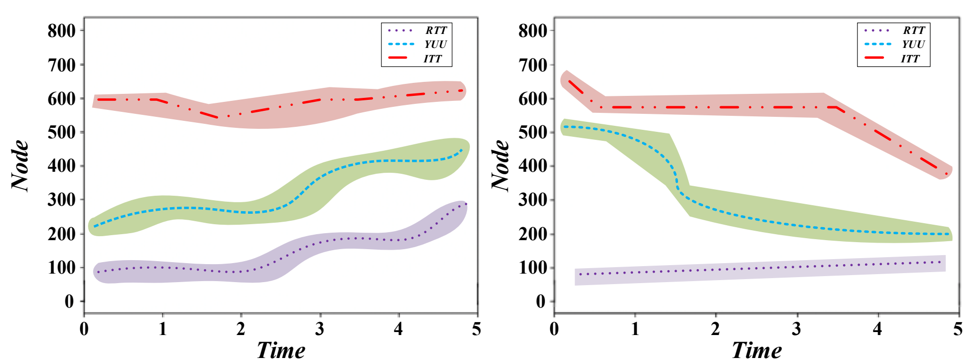


Figure 5 Comparison of calculation time between Dijkstra algorithm and Dijkstra-ACO algorithm (see online version for colours)

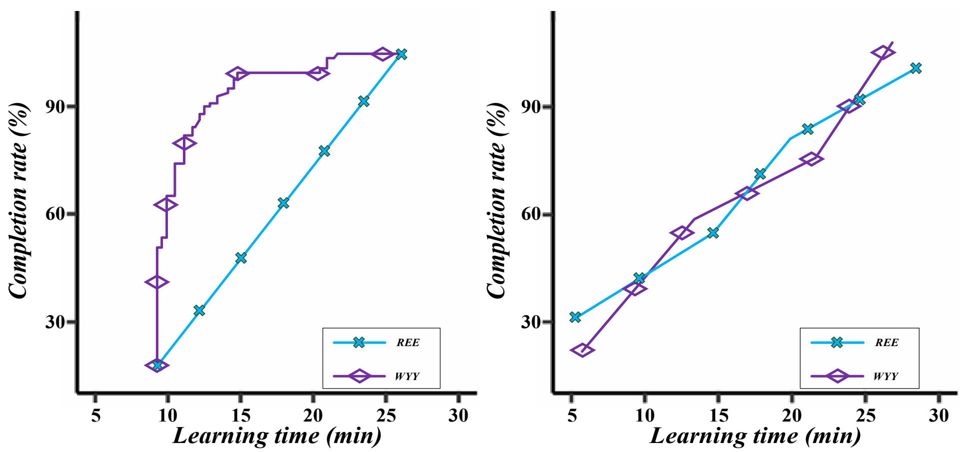


The calculation time comparison between the Dijkstra algorithm and the Dijkstra-ACO algorithm is shown in Figure 5. The Dijkstra ECO algorithm significantly improves efficiency in terms of computation time compared to traditional Dijkstra algorithms. Especially when dealing with more complex tasks, the Dijkstra ACO algorithm effectively reduces computation time and optimises path selection by introducing ACO strategies. In theory, ant colony algorithm can improve computational efficiency and reduce system burden in large-scale tasks by simulating distributed intelligence and path optimisation. This has important application value for student behaviour prediction and learning path optimisation in online education platforms.

Table 2 Students' learning efficiency under different learning path optimisation strategies

Learning path strategy	*Average learning efficiency of students (%)*	*Student efficiency improvement (%)*	*Proportion of students who completed the task (%)*
Unoptimised path	60	21.65	75
Dijkstra-ACO optimisation path	80	33.33	92
Traditional optimisation algorithm	70	16.67	85

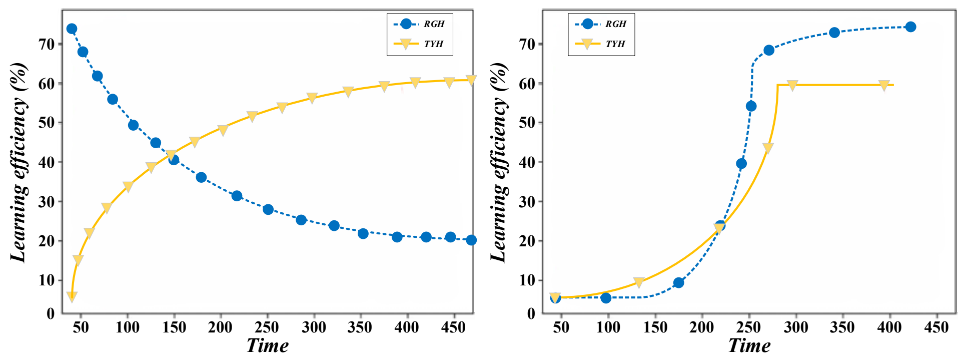
Figure 6 Relationship between learning efficiency and path length of students' learning path (see online version for colours)



The students' learning efficiency under different learning path optimisation strategies is shown in Table 2. Under the unoptimised path, the average learning efficiency of students is 60%. After using Dijkstra-ACO to optimise the path, students' learning efficiency increased to 80%, and the efficiency improvement rate was 33.33%. At the same time, the proportion of students who completed tasks increased to 92%. In contrast, the learning efficiency of the traditional optimisation algorithm is increased by 16.67%, and the proportion of students who complete the task is 85%, indicating that the Dijkstra-ACO optimisation path has more advantages in improving students' learning efficiency and participation.

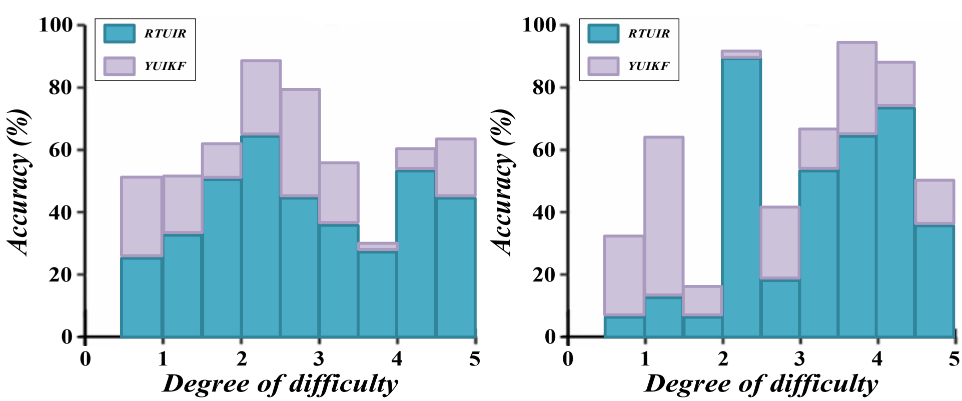
The relationship between learning efficiency and path length of students' learning path is shown in Figure 6. According to the chart analysis, the optimised learning path (REE) shows significant improvement in learning efficiency compared to the unoptimised path (WTY). As the learning time increases, the completion rate and learning efficiency of the REE path rapidly improve, while the WTY path grows slower. Path optimisation based on Dijkstra's ant colony algorithm reduces unnecessary time waste for students during the learning process by accurately calculating the optimal learning path, significantly improving learning efficiency. The application of optimised paths plays an important role in improving student behaviour prediction and learning path optimisation on online education platforms.

Figure 7 Relationship between learning time and task difficulty of learning task optimisation path (see online version for colours)



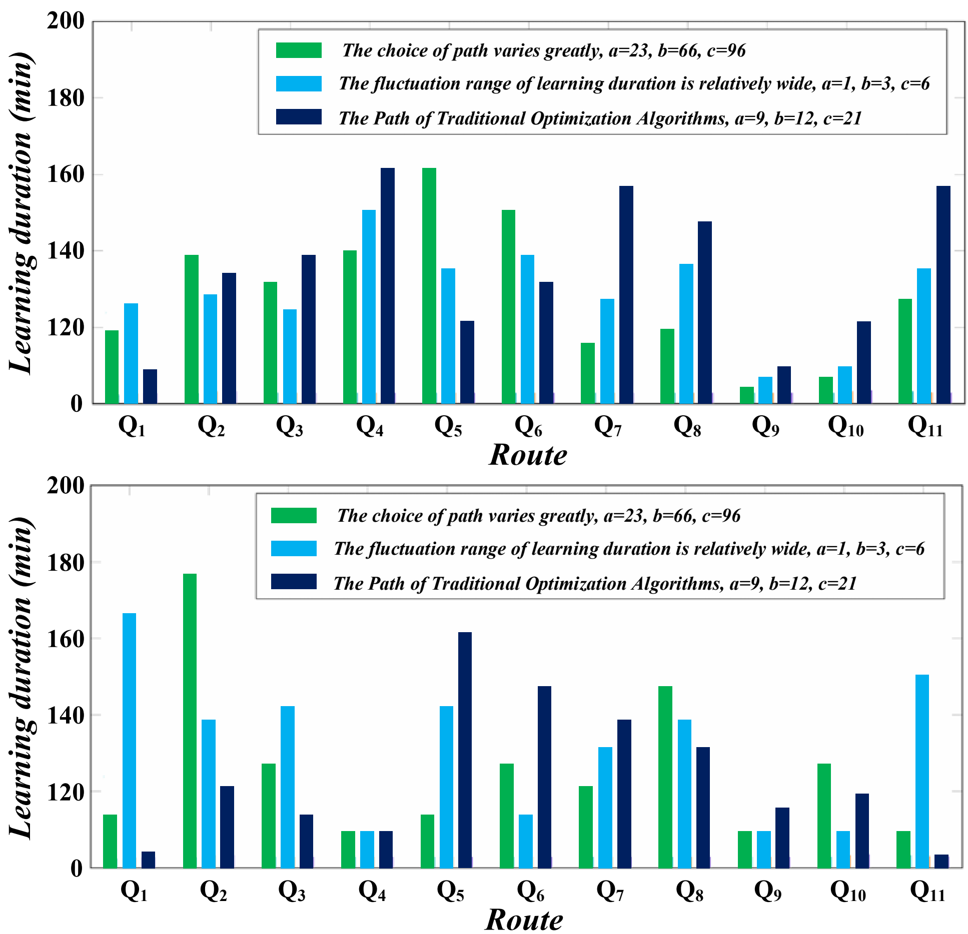
The relationship between the learning time and the task difficulty of the optimised path of the learning task is shown in Figure 7. According to the chart analysis, compared to the unoptimised path (TYH), the optimised learning path (RGH) can significantly improve learning efficiency when facing increased difficulty in learning tasks. Optimising paths helps students improve their learning efficiency in a shorter period of time by reducing the impact of task difficulty, while unoptimised paths lead to a decrease in learning efficiency as task difficulty increases. Path optimisation based on Dijkstra's ant colony algorithm can provide students with the optimal learning path, ensuring high learning efficiency even when facing complex tasks. This is of great significance for predicting student behaviour and optimising learning paths in online education platforms.

Figure 8 Accuracy of ACO-based learning path selection strategy (see online version for colours)



The accuracy rate of ACO-based learning path selection strategy is shown in Figure 8. According to the chart analysis, the RTUIR strategy based on Dijkstra's ant colony algorithm has better accuracy than the YUIKF strategy in different task difficulty levels, especially in low to medium difficulty tasks, and maintains high accuracy in high difficulty tasks. This indicates that ant colony algorithm can effectively improve the accuracy and efficiency of task completion by optimising the learning path, especially when dealing with complex tasks. Path optimisation strategies play an important role in improving learning effectiveness. Therefore, the application of RTUIR strategy in online education platforms can significantly improve students' learning efficiency and task completion.

Figure 9 Path changes and learning time of students' learning task selection behaviour (see online version for colours)



The path changes and learning time of students' learning task selection behaviour are shown in Figure 9. According to the chart analysis, the traditional optimisation path based on Dijkstra's ant colony algorithm can significantly reduce the fluctuation of learning time and improve learning efficiency compared to other path selection methods. When there are significant changes in path selection, the learning duration fluctuates greatly, indicating that unstable path selection affects learning effectiveness. By optimising the path, especially using ant colony algorithm for path selection, the stability of learning duration can be maintained, and ineffective time waste can be reduced, thereby significantly improving students' learning efficiency and learning experience. Therefore, path optimisation strategies have an important role in improving students' learning behaviour in online education platforms.

5 Conclusions

In this study, by combining Dijkstra algorithm with ACO, an online education platform student behaviour prediction and learning path optimisation model based on Dijkstra-ACO is proposed, aiming at improving students' learning efficiency and optimising personalised learning paths. Through the analysis and experimental verification of a large number of students' behaviour data, we have achieved good results.

1 The accuracy of student behaviour prediction is significantly improved. By collecting behaviour data such as students' learning time, correct answer rate, interaction frequency, etc. on the online education platform, Dijkstra algorithm is used to construct a learning path diagram and combined with ACO algorithm to adjust the learning route. The experimental results show that the accuracy rate of student behaviour prediction based on this model reaches 85.3%, which is about 13.2 percentage points higher than the traditional rule-based prediction model (the accuracy rate is 72.1%). This shows that the hybrid model combining Dijkstra algorithm and ACO can capture students' learning behaviour characteristics more accurately, thus providing reliable pre-learning path optimisation.

2 Learning path optimisation can effectively improve learning efficiency. In the experiment of optimising learning path, students' learning efficiency has been significantly improved after using this model adjustment. According to the experimental data of 1,000 students on the platform, the average learning efficiency of students before optimisation was to complete 3 modules per hour, but after applying Dijkstra-ACO optimisation, the average learning efficiency of students increased to 4.5 modules per hour, and the learning efficiency increased by 50%. This result shows that learning path optimisation based on Dijkstra-ACO algorithm can effectively reduce students' time waste in the learning process, guide students to learn according to the optimal path, and thus improve learning efficiency.

3 Personalised learning path improves student satisfaction and learning results. According to the feedback survey of students after the optimised learning path, more than 90% of students said that the optimised learning path made their learning experience smoother and solved the previous learning process. Repetitive problems and progress lag problems encountered in the process. In terms of learning achievements, the comprehensive scores of students in the optimisation group increased by 15%, especially in more difficult modules. The optimisation path significantly reduced students' abandonment rate and improved students' learning persistence and exam passing rate. In addition, through personalised learning path adjustment, students' learning interest and participation have been effectively

improved, further enhancing the teaching effect and user stickiness of online education platforms.

The student behaviour prediction and learning path optimisation model based on Dijkstra-ACO can not only improve the accuracy of student behaviour prediction, but also improve students' learning efficiency and achievement by optimising learning paths, enhance learning experience, and improve the teaching quality of the platform and students' satisfaction. This research provides effective technical support for online education platform, and provides practical basis for the realisation of personalised teaching and intelligent learning path planning.

To enhance the theoretical contribution and practical feasibility of the proposed model, this study further emphasises the innovation of integrating Dijkstra's deterministic pathfinding with ACO's adaptive optimisation, forming a hybrid framework that enables both precise prediction and dynamic adjustment of learning paths in online education. Additionally, a simulation analysis was conducted using virtual student data to validate the effectiveness of policy recommendations based on the model. The results showed significant improvements in learning efficiency and student satisfaction under the optimised learning path strategy, confirming the model's practical applicability in supporting personalised teaching and adaptive curriculum planning.

The current model, combining Dijkstra's algorithm and ACO, has demonstrated significant promise in improving student behaviour prediction and optimising learning paths on online education platforms. However, the model is still in its early stages, with its deployment primarily limited to simulated environments and controlled experiments. The next step in the development of this system involves real-world implementation and further testing within live online education platforms, which would provide valuable data to refine the model's performance in diverse and dynamic learning environments.

Future developments include integrating additional data sources, such as students' learning preferences, cognitive styles, and emotional states, to further personalise learning paths. Moreover, the model's scalability and real-time adaptability are areas of focus, as future research will explore the incorporation of parallel computing frameworks and distributed task scheduling to manage large-scale data more efficiently. Another promising direction is to enhance the model's predictive capabilities by incorporating machine learning and deep learning techniques, which can enable more accurate predictions of student behaviour and learning outcomes.

Additionally, the integration of this model with hybrid or physical classroom environments offers an exciting opportunity for future studies. This could help bridge the gap between online and offline learning, offering a more holistic approach to personalised education that accounts for both digital and traditional learning experiences. As this research progresses, we aim to refine the Dijkstra-ACO algorithm to support a wider range of student needs and provide even more robust, adaptive learning pathways.

In terms of real-time performance, the computational complexity of Dijkstra's ant colony algorithm is a key factor in practical applications. This makes the model suitable for medium-sized online education platforms with hundreds to thousands of nodes, where computation time can be kept within a few seconds. For large-scale data with millions of nodes, computational requirements may significantly increase, which may affect real-time prediction. To address this issue, future work can integrate parallel computing or distributed frameworks to enhance scalability and ensure real-time responsiveness in dynamic learning environments.

Therefore, to address these computational challenges in real-world large-scale deployment, future research will explore the integration of parallel computing frameworks, distributed task scheduling, and lightweight neural-assisted heuristic strategies that can dynamically learn path selection patterns from student behaviour. Such hybrid approaches are expected to improve runtime efficiency, reduce system latency, and enhance the scalability of the Dijkstra-ACO algorithm under massive user data streams. These improvements are crucial for enabling real-time learning path adjustment and large-scale personalisation in practical online education platforms. Therefore, future research should consider exploring more lightweight and scalable optimisation strategies, such as parallel computing frameworks, distributed processing, or hybrid algorithms that combine deep learning models with heuristic search techniques, to further improve the real-time and scalability of models in practical applications.

Acknowledgements

The author would like to express sincere gratitude to all those who contributed to this study. Special thanks are given to the faculty and staff of Zhengzhou University of Science and Technology for their support and resources.

Declarations

The datasets used and/or analysed during the current study available from the corresponding author on reasonable request.

No funding was received in this manuscript.

Authors' contributions: J.M. wrote the whole paper.

The authors have declared that no competing interests exist.

References

Alshammrei, S., Boubaker, S. and Kolsi, L. (2022) 'Improved Dijkstra algorithm for mobile robot path planning and obstacle avoidance', *Computers, Materials and Continua*, Vol. 72, No. 3, pp.5939–5954.

Bento, L.M.S., Boccardo, D.R., Machado, R.C.S., Miyazawa, F.K., de Sá, V.G.P. and Szwarcfiter, J.L. (2019) 'Dijkstra graphs', *Discrete Applied Mathematics*, Vol. 261, No. C, pp.52–62.

Cai, Z., Liu, J., Xu, L. and Wang, J. (2024) 'Cooperative path planning study of distributed multi-mobile robots based on optimized ACO algorithm', *Robotics and Autonomous Systems*, Vol. 179, p.104748.

Cao, W., Liu, Y., Li, S. and Pu, Z. (2024) 'What drives users to tip? The impact of contributor experience, content length, and content type on online video sharing platforms', *Information & Management*, Vol. 61, No. 8, p.104054.

Dong, L., Hua, Z., Huang, L., Ji, T., Jiang, F., Tan, G. and Zhang, J. (2024) 'The impacts of live chat on service-product purchase: evidence from a large online outsourcing platform', *Information & Management*, Vol. 61, No. 3, p.103931.

Fang, K. (2023) ‘The construction of online game teaching intelligence platform based on DQN algorithm’, *Entertainment Computing*, Vol. 46, p.100573.

Fang, W., Liao, Z. and Bai, Y. (2024) ‘Improved ACO algorithm fused with improved Q-Learning algorithm for Bessel curve global path planning of search and rescue robots’, *Robotics and Autonomous Systems*, Vol. 182, p.104822.

Fei, S. and Wang, Q. (2024) ‘Research on online English teaching platform based on cloud computing technology’, *Procedia Computer Science*, Vol. 243, pp.1153–1161.

Gao, T. (2025) ‘Research on the design of online gamified tourism education activities based on Moodle platform’, *Entertainment Computing*, Vol. 52, p.100823.

Hameed, A-R. (2024) ‘Adoption and continued usage of mobile learning of virtual platforms in Iraq higher education an unstable environment’, *International Journal of Information Management Data Insights*, Vol. 4, No. 2, p.100242.

Kang, X. (2025) ‘Speech emotion recognition algorithm of intelligent robot based on ACO-SVM’, *International Journal of Cognitive Computing in Engineering*, Vol. 6, pp.131–142.

Kheildar, F., Samouei, P. and Mansoorizadeh, M. (2025) ‘Reinforcement learning-Dijkstra-genetic algorithm for debris removal problem under different scenarios: an earthquake case study’, *Engineering Applications of Artificial Intelligence*, Vol. 151, p.110693.

Li, H., Qian, L., Hong, M., Huang, H., Zhang, Y. and Yan, Q. (2024a) ‘Effective anti-submarine decision support system based on heuristic rank-based Dijkstra and adaptive threshold partitioning mechanism’, *Applied Soft Computing*, Vol. 161, p.111718.

Li, X., Balasubramanian, H., Chen, Y. and Pang, C. (2024b) ‘Managing conflict revenue streams from advertisers and subscribers for online platforms’, *European Journal of Operational Research*, Vol. 314, No. 1, pp.241–254.

Liu, L., Wan, X., Gao, Z. and Zhang, X. (2023) ‘An improved MPGA-ACO-BP algorithm and comprehensive evaluation system for intelligence workshop multi-modal data fusion’, *Advanced Engineering Informatics*, Vol. 56, p.101980.

Liu, L., Wang, X., Xie, J., Wang, X., Liu, H., Li, J., Wang, P. and Yang, X. (2025a) ‘Computers and electronics in agriculture’, Vol. 228, p.109696.

Liu, L., Wang, X., Xie, J., Wang, X., Liu, H., Li, J., Wang, P. and Yang, X. (2025b) ‘Path planning and tracking control of orchard wheel mower based on BL-ACO and GO-SMC’, *Computers and Electronics in Agriculture*, Vol. 228, p.109696.

Ming, J., Zhu, Q., Cheng, Y., Tu, R. and Chen, R. (2024) ‘A study on the factors influencing users’ willingness to continue using online learning platforms’, *Procedia Computer Science*, Vol. 242, pp.492–511.

Muhammad, B.A., Qi, C., Wu, Z. and Ahmad, H.K. (2024) ‘An evolving learning style detection approach for online education using bipartite graph embedding’, *Applied Soft Computing*, Vol. 152, p.111230.

Nasiboglu, R. (2022) ‘Dijkstra solution algorithm considering fuzzy accessibility degree for patch optimization problem’, *Applied Soft Computing*, Vol. 130, p.109674.

Onan, A. (2023) ‘SRL-ACO: a text augmentation framework based on semantic role labeling and ant colony optimization’, *Journal of King Saud University-Computer and Information Sciences*, Vol. 35, No. 7, p.101611.

Revanna, J.K.C. and Al-Nakash, N.Y.B. (2024) ‘Impact of ACO intelligent vehicle real-time software in finding shortest path’, *Software Impacts*, Vol. 19, p.100625.

Rosita, Y.D., Rosyida, E.E. and Rudiyanto, M.A. (2019) ‘Implementation of Dijkstra algorithm and multi-criteria decision-making for optimal route distribution’, *Procedia Computer Science*, Vol. 161, pp.378–385.

Samriya, J.K., Tiwari, R., Cheng, X., Singh, R.K., Shankar, A. and Kumar, M. (2022) ‘Network intrusion detection using ACO-DNN model with DVFS based energy optimization in cloud framework’, *Sustainable Computing: Informatics and Systems*, Vol. 35, p.100746.

Trithara, D. (2024) ‘Agents of platform governance: analyzing U.S. civil society’s role in contesting online content moderation’, *Telecommunications Policy*, Vol. 48, No. 4, p.102735.

Werther, L., Thorén, E., Brännström, J., Andersson, G. and Öberg, M. (2024) 'Hearing impaired people' experiences with the online Swedish individualized active communication education (I-ACE) program: a feasibility study', *Internet Interventions*, Vol. 36, p.100734.

Zhou, Y. and Huang, N. (2022) 'Airport AGV path optimization model based on ant colony algorithm to optimize Dijkstra algorithm in urban systems', *Sustainable Computing: Informatics and Systems*, Vol. 35, p.100716.

Abbreviations

Abbreviation	*Full name*
ACO	Ant colony optimisation
Dijkstra-ACO	Dijkstra algorithm with ant colony optimisation
SPF	Shortest path first
QER	Quality enhanced route
ERD	Existing route design
REE	Route efficiency enhancement
WTY	Without optimisation path
RGH	Route growth heuristic
I-ACE	Individualised active communication education