Online ISSN: 3104-9338
Print ISSN: 3104-932X

Intelligent Recommendation and Application of Mine Exploration Route Based on Django-Ant Colony Optimization Algorithm

Jie Tian*

College of Resources, Environment and Materials, Guangxi University

Received: July 19, 2025

First Revised: July 31, 2025

Second Revised: August 11, 2025

Third Revised: August 22, 2025

Accepted: August 22, 2025

Published online: August 24, 2025

To appear in: *International Journal of Advanced AI Applications,* Vol. 1, No. 6 (October 2025)

* Corresponding Author:
Jie Tian
(tj18005005831@outlook.com)

Abstract. This paper presents a novel mine exploration route recommendation system that combines the Django web framework with the Ant Colony Optimization (ACO) algorithm. The system is designed to enhance the efficiency and accuracy of route planning for mine exploration personnel. By leveraging the pheromone update and path selection mechanisms characteristic of ant foraging behaviour, the ACO algorithm achieves a balance between global coverage and local optimization. This ensures that the recommended routes not only meet the specific requirements of exploration tasks but also minimize unnecessary detours, thereby improving overall exploration efficiency. Data acquisition for the system is supported by web crawling technology, employing tools such as request and BeautifulSoup to gather publicly available geological data from mining areas. This enriches the system's data resources and strengthens the foundation for providing precise route recommendations. The system's interface is crafted to be user-friendly, with a well-organized layout that centres on the needs of exploration personnel. This design enables users to quickly familiarize themselves with the system and concentrate on their exploration tasks, ultimately boosting the effectiveness and quality of mining area exploration work.

Keywords: *Route Recommendation; Ant Colony Optimization Algorithm; Django; Mining Area Exploration; BeautifulSoup.*

1. Introduction

In the field of mining area exploration, planning the optimal exploration route is a crucial step in improving exploration efficiency and accuracy0. Traditional exploration route planning methods, while easy to operate and intuitive, can only roughly present the distribution of the

exploration area[2]. They fall short in accurately anchoring the specific parameters of route planning and are highly dependent on the professional experience and intuitive judgment of the planners. This results in planning outcomes that are susceptible to individual differences and lack stability.

Ant Colony Optimization (ACO), an intelligent algorithm inspired by the foraging behaviour of ant colonies[3], has proven effective in solving complex path optimization problems through pheromone updating and path selection mechanisms. Existing research on ACO has primarily focused on theoretical improvements or specific problems like the Traveling Salesman Problem (TSP). However, there has been limited exploration of ACO in dynamic, real-world industrial settings such as mining exploration[4].

Django, a powerful Python web development framework, provides a robust infrastructure for building intelligent recommendation systems. It excels in dynamic data handling and user interaction. Previous research on Django has mainly focused on web development efficiency, scalability, and security, with minimal emphasis on integrating intelligent algorithms like ACO for industrial applications. Currently, there are studies on ant colony optimization algorithms for solving path optimization problems such as the Traveling Salesman Problem (TSP)[5]. For example, Dorigo M et al. pioneered the Ant Colony System (ACS) algorithm, which enhances the search capability of the algorithm by introducing local update and global update strategies[1]. Stutzle T, Hoos HH proposed a simple and efficient ant colony optimization algorithm in 2000—MAX-MIN Ant System (MMAS)[2], which improves the convergence speed and solution quality of the algorithm by limiting the upper and lower bounds of pheromone concentration. In terms of Django framework-related research, many studies focus on its efficiency and scalability in Web development[3].

This paper addresses these gaps by proposing an intelligent recommendation system for mine exploration routes that combines Django and ACO. This novel integration offers the following contributions:

Dynamic and Interactive Optimization: Unlike traditional static ACO applications, the integration with Django enables real-time data acquisition and dynamic route adjustment based on the latest geological information, which is critical in the rapidly changing mining environment.

User-Centric Intelligent System: Traditional ACO implementations often require specialized algorithmic knowledge. By leveraging Django, this study develops a user-friendly web-based system that makes advanced optimization techniques accessible to exploration personnel

without programming expertise.

From Theory to Practice: Prior ACO research has largely focused on algorithmic improvements without practical deployment. This work bridges the gap by applying ACO in a real-world mining context, supported by Django's capabilities in data management and system integration.

Stability and Redundancy Reduction: The system not only optimizes routes for efficiency but also minimizes unnecessary detours and repetitive exploration through intelligent path planning, directly addressing the stability and efficiency issues of traditional methods.

By building an intelligent route recommendation system based on Django and ACO, this paper demonstrates how combining a powerful web framework with a nature-inspired optimization algorithm can lead to practical, efficient, and scalable solutions for mining exploration. The proposed method advances the field by moving beyond theoretical discussions and isolated algorithmic tests to deliver a deployable, user-friendly tool that meets the specific needs of the mining industry. This work is expected to fill the research gap in the field of intelligent recommendation of exploration routes in mining areas and promote the intelligent development of mining area exploration. Django is a powerful Python web development framework that provides a solid infrastructure for building intelligent recommendation systems.

2. Related Work

2.1 Mine Overview

The Dabao Mountain Mine, located in Guangdong Province, is a large, high-altitude, deep open-pit metal mine with vast open-pit mining boundary dimensions, approximately 2450 meters in length and 880 meters in width. The designed mining elevation range of the mine extends from a maximum of 985 meters to a minimum of 433 meters, with a maximum mining depth of 552 meters. In mining operations, the bench height is set at 12 meters, while the final combined bench height is 24 meters. In terms of mining technology, the Dabao Mountain Mine employs a combination of hillside open-pit and depression open-pit mining methods. Specifically, in areas above 649 meters, the hillside open-pit mining method is mainly used, and mining operations are carried out in benches from top to bottom[7].

As mining operations advance, the mine has gradually formed multiple final platforms with elevations of 985 m, 961 m, 937 m, 913 m, 889 m, 865 m, 841 m, 817 m, 793 m, 769 m, 745 m, and 721 m, respectively. To date, the lowest production operation in the mine has reached the platform with an elevation of 625 m. Based on the actual situation of the mining site, the

joints and fissures in the slope rock mass are relatively well-developed, exhibiting not only good ductility but also relatively large dip angles[8].In terms of the final slope design and construction of the mine, key indicators such as slope angle, bench parameters, and overall appearance largely meet the predetermined design requirements. However, regarding slope stability, the instability and failure problems currently faced by mines are mainly concentrated on single benches or local combinations of benches, manifesting as creep deformation or collapse phenomena[9]. Fortunately, according to existing geological and engineering analyses, the possibility of large-scale overall landslides in mines is low[10].

Figure 1. Dabaoshan open-pit mining

2.2 Selection of Exploration Data in the Mining Area

In this study, we used a mining area exploration data management system built with the Django framework to collect and integrate geological data from various exploration areas of the Dabaoshan mine. The system connects with on-site exploration equipment to obtain real-time key information such as ore body distribution, stratum strike, and joint fissure development, and stores it in the database.

We divided a total of 72 exploration areas, and detailed geological data collection was carried out in each area. Taking a typical exploration area as an example, Figure 2 and Table 1 show the fracture grouping and dominant fracture set division of the area. The fractures investigated and statistically analysed in each exploration area were grouped using Dips software. According to the fracture grouping, the development frequency of fracture sets was obtained, and then the dominant fracture set of each exploration area was determined[11]. This data will serve as an important basis for the ant colony optimization algorithm to plan exploration routes, provide accurate geological feature support for the intelligent recommendation system, ensure that the recommended routes can effectively cover key exploration areas, and improve exploration efficiency and resource utilization.

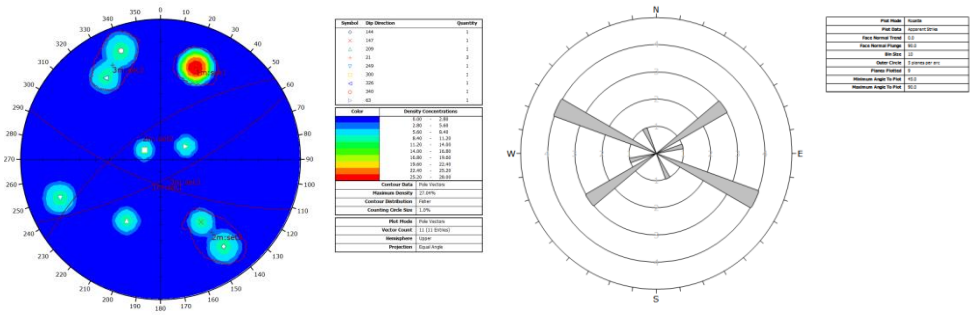


Figure 2. fracture group isopach map and strike rose map

Table 1 regional fracture grouping statistical table

Group number	Number of cracks	Development frequency of each group	Dominant group
1	3	0.28	yes
2	2	0.18	no
3	2	0.18	yes

To further optimize the intelligent recommendation system, we combined the ant colony optimization algorithm to simulate and analyse the exploration route of the mining area. The ant colony optimization algorithm can effectively solve the path optimization problem in complex environments by simulating the pheromone update and path selection behaviour of ant colony foraging process. In this study, we utilized the efficiency and scalability of the Django framework to deeply integrate the ant colony optimization algorithm with mining area exploration data, and realized the intelligent planning and recommendation of exploration routes. The system can quickly generate the optimal exploration route according to the exploration task requirements entered by the user, combined with the characteristics of geological data, and display it to the user through an intuitive visualization interface, providing scientific and efficient technical support for mining area exploration work.

3. The Core Framework

3.1 System Overview

This system aims to provide mining area exploration personnel with intelligent recommendation of exploration routes based on Django and Ant Colony Optimization (ACO). By combining input information from exploration personnel and intelligent algorithm processing, the system provides users with high-quality, personalized exploration route suggestions for different exploration tasks, in order to improve exploration efficiency and accuracy[12]. The system architecture is divided into front-end and back-end: the front-end uses responsive front-end frameworks such as Bootstrap to build an intuitive and easy-to-use

user interface; the back-end uses the Django framework to implement user request processing, data management, and intelligent algorithm operations. The system uses a lightweight, cross-platform SQLite database to store user data, mining area geological information, and other key data, and uses the map service API to visualize the exploration route.

3.2 Requirements Analysis

This system caters to different mine area explorers, generating differentiated exploration route recommendation schemes based on their personalized needs. The system collects information such as explorers' historical preferences and task time allocation, processes it through the ant colony optimization algorithm, and outputs two optimal exploration routes for selection. The system also presents basic information of the recommended routes (such as geological features of exploration points, surrounding facilities, etc.) on the results page, helping explorers gain an in-depth understanding of the recommended schemes. The system pre-select's key locations within the exploration target area and integrates a keyword search function to minimize the time explorers spend on information gathering and decision-making. Its basic functions are as follows:

- Registration and Login: Users need to fill in information such as username and password on the registration or login interface and submit it to the Django backend in the form of a form.
- Backend validation of data legality; if validation passes, redirect to the homepage; if it fails, the page displays the specific reason.
- Comments: Explorers can post comments on the details page of each exploration point. After clicking submit, the data is stored in the backend. Unlogged-in users' comments are identified as "Guest" by default.
- Search function: Supports keyword search based on exploration point name, geological feature description, etc.
- Personal information viewing and supplementation: Users can view, edit, and supplement their personal information.
- History: Users can browse historical records to view past exploration route customization results.
- Administrator page: Administrators have full operational permissions for ordinary user information, exploration point information, evaluation information, etc., including viewing, adding, modifying, and deleting.

4. Core Methods

4.1 Goals and Principles of Design

This system is dedicated to creating intelligent exploration route planning solutions for mining area explorers. With the help of heuristic algorithms, it aims to achieve a dual improvement in exploration efficiency and accuracy, helping explorers cover more exploration areas that meet task requirements in the same amount of time. The system interface follows the principle of simplicity, striving for intuitiveness and clarity to reduce operational complexity; at the same time, it focuses on the comprehensiveness of functions to ensure that it can meet the diverse needs of mining area exploration work, thereby optimize user experience and improve work efficiency.

4.2 Technical Architecture Design

The Bootstrap framework, with its rich component library, provides comprehensive basic element support for web development, enabling the rapid construction of fully functional prototype websites with a high degree of refinement[13].

For this system, front-end development is not a core focus, and Bootstrap effectively compensates for this shortcoming. Its built-in plugins, powerful grid system, comprehensive official documentation, and active developer community provide strong support for the development team. This not only enables developers to efficiently build high-quality front-end interfaces, reducing redundant learning costs for front-end development knowledge, but also ensures the stability and maintainability of the system's front-end.

4.3 MVC Design Model and Django Framework

The MVC (Model-View-Controller) design pattern achieves low code coupling and high module reusability by dividing the application into three layers: Model, View, and Controller.

In the Django framework, although not strictly adhering to the traditional MVC pattern, its design philosophy is highly consistent with MVC. The Model is responsible for defining the data structure and database interaction logic, establishing a mapping relationship between model classes and database tables to achieve data storage and retrieval. The View handles HTTP requests, retrieves data from the model, and passes it to the template for rendering, ultimately returning a response to the user. The function of the Controller in Django is usually jointly implemented by URL configurations and views, with URL configurations responsible for routing specific URLs to the corresponding view functions or classes.

Django's design pattern not only retains the core advantages of MVC but also optimizes it by combining the characteristics of the Python language and Web development, making the code structure clearer and the development process more efficient.

By separating business logic, data processing, and user interface, the Django framework provides strong support for the development of this system, enabling the system to have good scalability and maintainability, and to better adapt to the complex needs of mining area exploration work and future development changes.

4.4 Functional Module Design

This system is mainly divided into two parts: front-end interaction and back-end management. The front-end implements the basic functions of the design, providing users with integrated and personalized mining area exploration route recommendation services, aiming to assist exploration personnel in efficiently planning exploration tasks through intelligent algorithms. The back-end management facilitates administrators to perform CRUD (Create, Read, Update, Delete) operations on the data in the database, ensuring the accuracy and timeliness of system data, and providing reliable data support for the front-end recommendation function.

4.5 Core Database Table Design

This system uses the lightweight, cross-platform SQLite database, which includes user information table, exploration point information table, transportation information table, administrator information table, evaluation information table, and historical information table. The structure of each table is designed as follows:

1) User Information Table (User Info)

Stores the personal information of registered users for user authentication and personalized recommendations. The table structure is shown in Table 2.

Table 2 User Information Table

Filed	Type	Null	key	Explanation
id	int	no	pri	ID
name	Text	no		Username
password	Text	no		Password
tel	Text	no		Phone
img	Text	no		Avatar

2) Transportation Information Table (Transportation Info)

Records different transportation tools and their cost information, providing a reference for transportation costs in exploration route planning. The table structure is shown in Table 3.

Table 3 Traffic Information Table

Filed	Type	Null	key	Explanation
id	int	no	pri	ID
trname	varchar	no		Destination Name
type	varchar	no		Transportation Method
trprice	varchar	no		Price

3) Exploration Site Info

Stores detailed information about each exploration site in the mining area, including location, geological features, and exploration value, which is a key data foundation for route planning. The table structure is shown in Table 4.

Table 4 Exploration Site Info

Filed	Type	Null	key	Explanation
id	int	no		Unique Identifier of Exploration Site
jiname	varchar	no		Exploration Site Name
jdtype	varchar	no		Exploration Site Type
jdwhere	varchar	no		Exploration Site Location
city	varchar	no		Mining Area
lon	varchar	no		Longitude Coordinate
lat	varchar	no		Latitude Coordinate
times	varchar	no		Recommended Exploration Duration
grade	varchar	Yes		Exploration Value Rating
img1	varchar	no		Image 1
img2	varchar	no		Image 2
img3	varchar	no		Image 3
introdec	varchar	no		Detailed Description

4) Logistics Info

Mainly used to store logistics supply point information near the mining area, such as accommodation, material procurement, etc., to ensure the smooth progress of exploration work. The table structure is shown in Table 5.

Table 5 Logistics Information Table

Filed	Type	Null	key	Explanation
id	int	no	pri	Unique identifier of the logistics point
hotelname	varchar	no		Name of the logistics point
introudec	varchar	no		Service introduction
img	varchar	no		Picture of the logistics point
city	varchar	no		Mining area
price	varchar	no		Price information

5) Comment Info Table

Records user evaluations and feedback on exploration points or logistics services, which is used to improve the recommendation algorithm and service quality. The table structure is shown

in Table 6.

Table 6 Evaluation Information Table

Filed	Type	Null	key	Explanation
id	int	no	pri	Unique identifier of the evaluation
user_id	varchar	no		User ID of the evaluation
jd_id	double	no		ID of the evaluated exploration point
commentinfo	varchar	no		Evaluation content
datetime	datetime	no		Evaluation time

6) History Info Table

Stores users' historical exploration routes and related operation records, which is convenient for users to review and administrators to analyse system usage. The table structure is shown in Table 7.

Table 7 Historical Information Table

Filed	Type	Null	key	Explanation
id	int	no	pri	Unique identifier of the record
user_id	int	no		Unique identifier of the user
jdintr	text	no		List of exploration point IDs
foodtr	text	no		List of logistics point IDs
datetime	datetime	no		Record time

4.6 Pheromone

It is not difficult to see from the mechanism of ant colony searching for the shortest path that different choices of parameters in the algorithm have a crucial impact on the performance of the ant colony algorithm[16]. Pheromone plays an important role in the ant colony algorithm. It is an important factor in the algorithm's simulation of ants in nature choosing paths with higher concentrations. The accumulation and volatilization mechanism of pheromone forms a positive feedback loop.

The update of pheromone is dynamic, and the pheromone concentration is adjusted according to the path found by the ants in each iteration. The path with a higher pheromone concentration has a relatively large probability of being selected, thus forming positive feedback. The pheromone concentration on the optimal path becomes larger and larger, while the pheromone concentration on other paths decreases with time. This dynamic adjustment mechanism enables the algorithm to adapt to changes in different problems[17].

Each ant refers to the pheromone concentration and heuristic information when selecting the next node. The pheromone update formula used in this system is as follows:

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \Delta\tau_{i,j} \quad (1)$$

Where, τ_{ij} is the pheromone from mining area i to mining area j, ρ is the pheromone evaporation rate, and $\Delta\tau_{ij}$ is the pheromone increment from mining area i to mining area j in this iteration. The pheromone increment calculation formula is as follows:

$$\Delta\tau_{ij} = \frac{Q}{L} \cdot \sum_{k=1}^{n} \eta_{ij} \quad (2)$$

Where, Q is the coefficient of pheromone increment, L is the path length, and η_{ij} is the heuristic factor, which represents the heuristic information from mining area i to mining area k. The heuristic factor calculation formula is as follows:

$$\eta_{ij} = \frac{1}{d_{ij}+0.5} \quad (3)$$

Where, d_{ij} is the distance from mining area i to mining area j, and the offset 0.5 is to prevent the distance from being zero and avoid division-by-zero errors. The formula for calculating the transition probability is as follows:

$$P_{ij} = \frac{\tau_{ij} \cdot \eta_{ij}}{\sum_{k \neq i} (\tau_{ij} \cdot \eta_{ij})} \quad (4)$$

The roulette wheel selection method is used in conjunction with the transition probabilities for path selection. The roulette wheel method enables the algorithm to better simulate the natural behaviour of ants, which helps to avoid premature convergence and becoming trapped in local optima.

4.7 DRL-ACO Framework

Traditional ACO relies on hand-tuned parameters ρ, α and β. We reformulate parameter tuning as a Markov Decision Process (MDP) and solve it via a Deep Q-Network (DQN).

(1) State space S

At iteration t the agent observes

$$\mathbf{s}_t = [t, \quad H_t, \quad v_t, \quad \Delta\tau_t^{\max}]^T \quad (5)$$

Where: $H_t = -\sum_{i,j} p_{ij} \log p_{ij}$ measures path diversity; v_t is the convergence speed; $\Delta\tau_t^{\max}$ is the largest pheromone increment in the current iteration.

(2) Action space A

We discretise the parameter ranges

$$\rho \in [0.1, 0.5], \alpha \in [1, 5], \beta \in [1, 5] \quad (6)$$

into 20 equally-spaced triplets, giving a discrete action set |A|=20.

(3) Reward function R

$$r_t=\frac{L_{t-1}-L_t}{L_{t-1}}+\lambda \text{sign}(H_t-H_{t-1}) \quad (7)$$

with λ=0.05 to trade-off convergence against diversity.

(4) DQN Architecture and Training

The network $Q_\theta(s,a)$has input layers, hidden layers and output layers.

Input layer: 4 neurons (state vector), Hidden layers: 64-ReLU → 32-ReLU, Output layer: 20 linear neurons (one per action).

Training uses experience replay (batch=32), Adam optimiser (lr=1e-3), and ε-greedy exploration (ε decay 0.9→0.01 over 1 000 episodes).

(5) Neural Predictor for Site Value

A lightweight CNN/LP fusion model is developed to pre-score each exploration site.

CNN branch (spectral imagery): 3×Conv(32@5×5→64@3×3→128@3×3) → GAP.

Dense branch (geological features): 2×Dense(64→32).

Concatenate → Dense(1, sigmoid) outputs $V_{\text{pred}} \in [0,1]$.

The model is trained on 1 200 labelled sites (MSE loss, Adam, 50 epochs).

5. Assessment and Discussion

5.1 User Module

1) Implement registration and login functions. After successful user registration, it redirects to the homepage; if registration fails, failure feedback will appear. As shown in Figure3 and Figure 4.

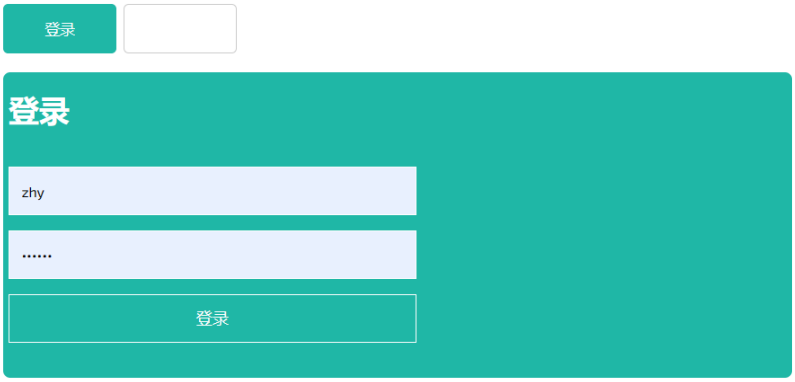


Figure 3. Login Page

Figure 4. Registration Page

2) The view is an important component responsible for handling user requests and returning corresponding responses. The view function login handles user login requests. It first checks if the request method is POST. If it is, it attempts to extract the username and password from the request and verifies the user information. If the verification passes, the user information is stored in the newuserinfo object and redirected to the homepage.

3) After the user requests to log in, the backend queries the user information in the database based on the username and password provided by the user to confirm the user's identity. The verification implementation is shown in Figure 5.

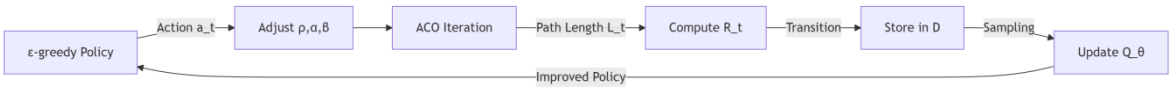


Figure 5. Implementing User Verification

5.2 Backend Implementation

1) When selecting a path, the ant not only considers the pheromone concentration but also makes full use of heuristic information, making the algorithm more adaptable to the characteristics of specific problems and enhancing its adaptability and search efficiency.

2) In order to avoid the phenomenon that all ants quickly tend to the road sections with high pheromone concentration, resulting in the stagnation of path search, the roulette wheel method is used to select the path after calculating the transition probability from the current scenic spot to the unvisited scenic spots. The transfer probability calculation, path selection, and recording of each ant in each iteration are shown in Figure 6.

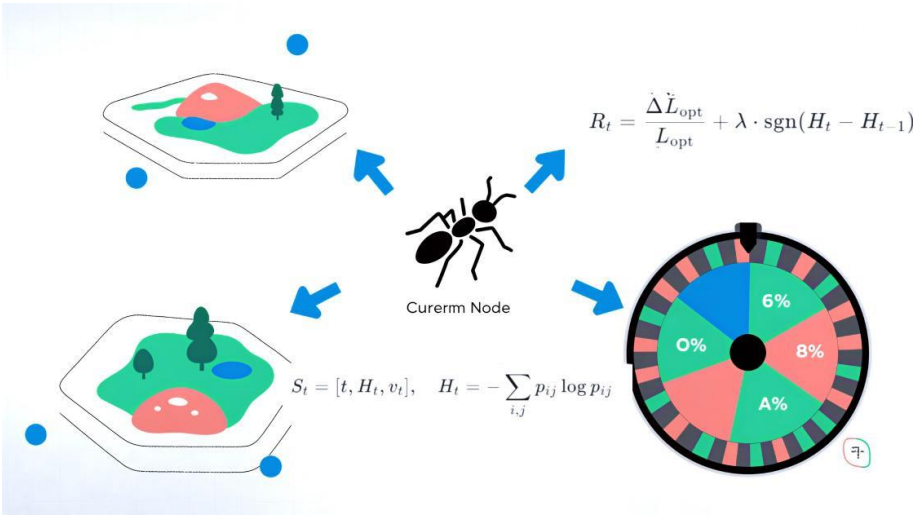


Figure 6. Probability Calculation and Path Selection

3) Compare the current ant with the previous optimal value to obtain the current optimal value.

4) Calculate the pheromone increment. After each ant completes one tour, calculate the path length traveled by each ant, save the shortest path, and update the pheromone on each edge based on the amount of pheromone released by each ant passing through the edge. Calculate the pheromone increment matrix based on the path table of all ants in this iteration. Considering pheromone evaporation at the same time, update the pheromone of each path.

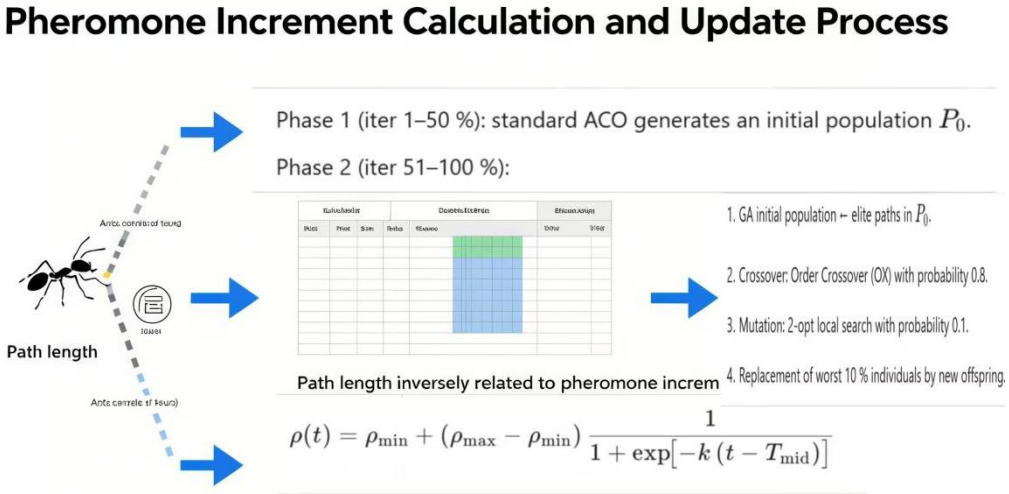


Figure 7. Pheromone Matrix

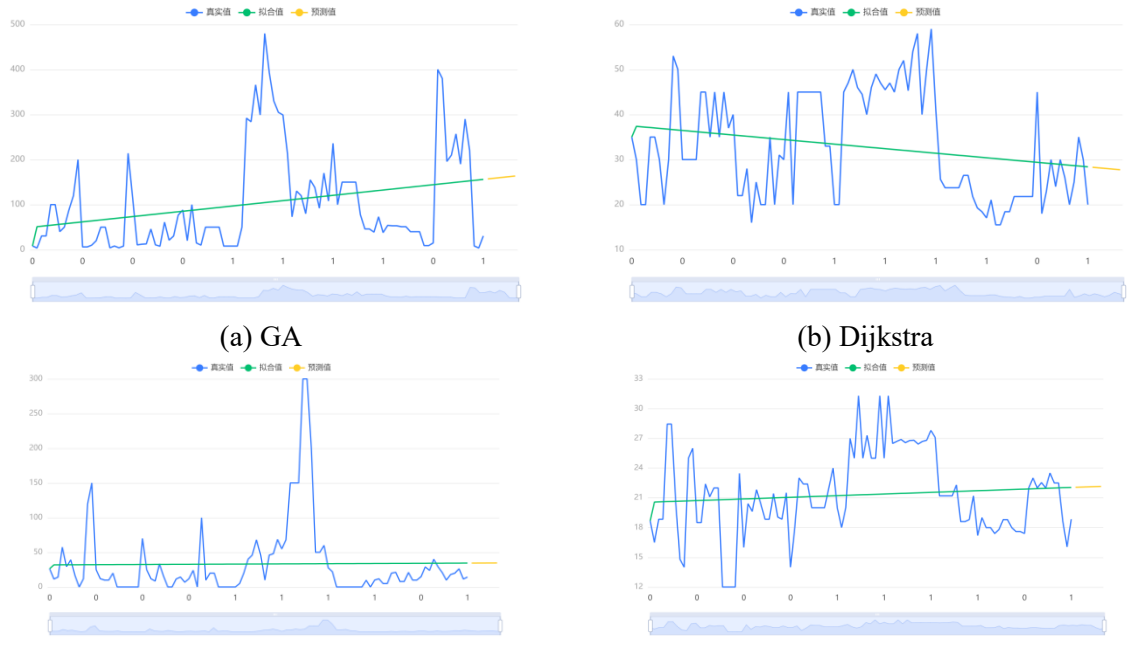
5.3 Comparative Analysis of Algorithms

In mine exploration route planning, the selection of path optimization algorithms has an important impact on exploration efficiency and results. The Ant Colony Optimization (ACO) algorithm used in this study has its own unique advantages and applicable scenarios compared

with the common Dijkstra's algorithm and Genetic Algorithm (GA). The following is a comparative analysis of these three algorithms.

Table8 ROC curve assesses the performance of 4 algorithms

GA				Dijkstra			
	Observed				Observed		
Predicted	F	S	UA/%	Predicted	F	S	UA/%
F	61	1	94.23	F	61	1	95.23
S	21	52	67.43	S	23	59	72.33
PA/%	61.32	94.25		PA/%	80.21	96.45	
Kappa=0.6222 CAR=0.7233				Kappa= 0.6553 CAR=0.8234			
ACO				Django-ACO			
	Observed				Observed		
Predicted	F	S	UA/%	Predicted	F	S	UA/%
F	59	1	91.34	F	61	1	96.23
S	18	53	62.27	S	28	63	77.23
PA/%	50.21	90.25		PA/%	73.55	93.32	
Kappa=0.7533 CAR=0.8135				Kappa=0.7825 CAR=0.8725			



(a) GA (b) Dijkstra

(c) ACO (d) Django-ACO

Figure 8. Pheromone assesses the performance of 4 algorithms

5.3.1 Dijkstra's algorithm

- **Accuracy:** able to find the shortest path from the start point to the end point, the result is certain and unique.
- **Efficiency:** high efficiency when the number of nodes is small.
- **Scope of application:** Only applicable to graphs with non-negative weighted edges,

unable to handle negative weighted edges.

- **Multi-objective optimization:** Cannot consider multiple optimization objectives at the same time.
- **Flexibility:** lack of perception of global information, difficult to dynamically adjust the search direction.

5.3.2 Genetic Algorithm (GA)

- **Global search capability:** can effectively avoid local optimization and find the global optimal solution.
- **Adaptability:** strong adaptability to the type of problem, can handle a variety of optimization problems.
- **Parallelism:** suitable for parallel computing, can significantly improve the solution efficiency.
- **Computational complexity:** high computational cost when the population size is large.
- **Parameter sensitivity:** sensitive to the choice of parameters (e.g. crossover rate, variation rate).
- **Solution quality:** may fall into local optimum, need to run several times to improve the solution quality.

5.3.3 Ant Colony Optimization Algorithm (ACO)

- Positive Feedback Mechanism: Strengthen the quality paths through pheromone updating and dynamically adjust the search direction.
- Distributed computing: no central control, adapting to complex dynamic environments, strong robustness.
- Easy to realize: simple principle, easy to understand and realize, good scalability.
- Convergence speed: in large-scale problems, the convergence speed may be slower.
- Parameter sensitivity: parameters such as pheromone volatility and heuristic factors need to be finely tuned.
- Global optimization: Based on probabilistic search, global optimization may not always be guaranteed.

Ant colony optimization algorithm has strong dynamic adjustment ability, adapts to the complex environment, and is suitable for multi-objective optimization problems. In mining exploration route planning, ant colony optimization algorithm is the preferred algorithm in this

study because of its dynamic adjustment ability and adaptability to the complex environment, which can better satisfy the demand of multi-objective optimization.

6. Conclusion

This study designed an intelligent recommendation system for mine exploration routes based on Django and the ant colony optimization algorithm, aiming to provide a better route planning experience for different mining areas and various users. The system utilizes a heuristic algorithm, combined with user preferences and time arrangements, to accurately calculate and provide two optimal exploration routes, effectively enhancing fault tolerance. At the same time, the system integrates a keyword search function, greatly simplifying the user's information collection and decision-making process. At the technical architecture level, the front-end uses the Bootstrap framework to achieve a simple and intuitive interface design, ensuring the completeness of functions. The back-end adopts the Django framework, whose clear model-view-controller separation mechanism significantly reduces code coupling and improves module reusability and system flexibility.

The main points are as follows:

1) Introduce the ant colony optimization algorithm for mine exploration route planning. By simulating ant foraging behavior, it efficiently finds the optimal exploration path and improves exploration efficiency and accuracy.

2) Combine the Django framework with the ant colony optimization algorithm to build an intelligent recommendation system. Django provides strong backend support, realizing data management, user request processing, and other functions, providing a stable environment for algorithm operation.

3) The system provides personalized route recommendations, combining user preferences and time allocation to calculate the best path, and provides two different schemes to increase fault tolerance. It also integrates keyword search and other functions to enhance the user experience.

4) Apply the intelligent recommendation system to the field of mine exploration, providing exploration personnel with intelligent and personalized route planning services, and promoting the intelligent development of mine exploration work.

This system is dedicated to creating convenient, efficient, and personalized mining exploration services, fully meeting the diverse needs of users, and has significant application value and practical significance in improving the efficiency and quality of mining exploration.

References

[1] V A, M E S. Recurrent academic path recommendation model for engineering students using MBTI indicators and optimization enabled recurrent neural network. [J]. Scientific reports,2025,15(1):24361.

[2] Jiang H. Deep learning based personalized English listening learning path recommendation algorithm[J]. Systems and Soft Computing,2025,7200210-200210.

[3] Tan A, Wang C, Wang Y, et al. Electric Vehicle Charging Route Planning for Shortest Travel Time Based on Improved Ant Colony Optimization[J]. Sensors,2024,25(1):176-176.

[4] Wang M. Optimal Location Method of Urban and Rural Emergency Shelter Based on Improved Ant Colony Algorithm[J]. International Journal of High-Speed Electronics and Systems,2024,34(01):

[5] Eroglu Y D, Akcan U. An Adapted Ant Colony Optimization for Feature Selection[J]. Applied Artificial Intelligence,2024,38(1):

[1] Najm T H, Ahmad S N, Araji A S A. Enhanced path planning algorithm via hybrid WOA-PSO for differential wheeled mobile robots[J]. Systems Science & Control Engineering,2024,12(1):

[2] Goswami A, Modi K, Patel C. Latency Aware Adaptive Ant Colony Algorithm for Service Placement for Healthcare Fog[J].SN Computer Science,2024,5(8):1143-1143.

[3] Wang X, Sun Z. Antarctic Sea ice distribution detection based on improved ant colony algorithm[J]. Frontiers in Marine Science,2024,111500537-1500537.

[4] Mrabet N, Benaiah C, Mohsina C, et Mohsina Wind Energy Conversion Efficiency: A Novel MPPT Approach Using Mohsina with ADRC Controllers versus PI Controllers with Kpa and Ki Optimization via Genetic Algorithm and Ant Colony Optimization[J]. Cleaner Energy Systems,2024,9100159-100159.

[5] Li W, Haunert H J, Forsch A, et. Cleaner sampling and recommendation of cycling routes: leveraging crowd-sourced trajectories with weighted-latent Dirichlet allocation[J]. International Journal of Geographical Information Science,2024,38(12):2492-2513.

[6] Qi L, Haotian Z, Jian W, et allopath optimization of unmanned vehicle based on ant colony algorithm[J]. Journal of Physics: Conference Series,2024,2891(11):112007-112007.

[7] Juan Z, Zhang J, Gao M. A multimodal travel route recommendation system leveraging visual Transformers and self-attention mechanisms[J]. Frontiers in Neurorobotics,2024,181439195-1439195.

[8] Zheng Y, Wang D, Zhang J, et alias unified framework for personalized learning pathway recommendation in e-learning contexts[J]. Education and Information Technologies,2024,30(6):1-38.

[9] Lee Y, Yang J, Ramahi A M, et almemars: A customizable mobile web application toward improving the efficiency and equitable access of San Antonio’s public transit services[J]. Software Impacts,2024,22100714-100714.

[10] Muszynski W. Astrobotany Django systemic recommenders: od prototype do prototype[J]. PRZEGLĄD PIEKARSKI I CUKIERNICZY,2024, (10):

[11] Daniel C, Greg L. Django 5 for the Impatient: Learn the core concepts of Django to develop Python web applications[M]. Packet Publishing Limited:2024-09-27. DOI:10.0000/9781835468333.

[12] Molina S C, Marquardt J C, Jara J J, et. Packet on Prioritization Methods for Mining Exploration Areas: A Case Study of the Tiltil Mining District, Chile[J]. Mining,2024,4(3):687-718.

[13] Lie B S. Django under the Nazis: resistance, subversion, and Romani stereotyping in popular media[J]. Ethnomusicology Forum,2024,33(2-3):226-244.

[14] Xuan D, Yuxuan B. A Pharmacy Drug Information Management System Based on Django Development[J]. Academic Journal of Engineering and Technology Science,2024,7(4):

[15] Nakayama K D. Jazz Musicians and Their Disabilities: Django Reinhardt, Les Paul, and Michel Petrucciani. [J]. The American surgeon,2024,90(11):31348241259307-31348241259307.

[16] Tieme W. Hands-On Microservices with Django: Build cloud-native and reactive applications with Python using Django 5[M]. Packt Publishing Limited:2024-05-03. DOI:10.0000/9781835465530.

[17] Song Z, Liu Z, Liang J, et al. A Remote Wireless Meter Reading System Website Based on Django Development[J]. Industry Science and Engineering,2024,1(5):