Dynamic Logistics Path Optimization via Integrated Ant Colony Optimization and Reinforcement Learning

You Caihong
School of Management, Fuzhou Technology and Business College University, Fuzhou Fujian 350000, China
E-mail: ychly202303@163.com

Keywords: artificial intelligence, logistics path planning, path optimization, intelligent optimization, ant colony optimization, reinforcement learning

Received: May 13, 2025

Logistics path planning plays a critical role in improving the efficiency and cost-effectiveness of distribution systems, especially under dynamic traffic conditions. This paper proposes a hybrid path optimization model that combines Ant Colony Optimization (ACO) with Deep Reinforcement Learning (RL), specifically a Deep Q-Network (DQN), to address the limitations of traditional static planning algorithms. The model integrates real-time traffic conditions and historical logistics data into a dynamic directed graph structure. ACO is first used to generate high-quality initial paths, which are encoded to initialize the RL environment and guide early exploration. As the vehicle navigates, real-time traffic fluctuations such as congestion and road closures trigger immediate re-optimization via the RL agent and adaptive pheromone updates in ACO. The model is evaluated using a real-world logistics dataset with 30 customer nodes under time window constraints and varying dynamic scenarios. Experimental results demonstrate that the proposed method reduces average delivery route length from 56.9 km to 52.3 km and lowers fuel and operational costs by 27%, while also achieving 100% punctuality. These findings validate the model's effectiveness, robustness, and potential for deployment in intelligent logistics distribution systems.

Povzetek:Razvit je hibridni model za nteligentno načrtovanje logističnih poti in optimizacija transporta v dinamičnih prometnih razmerah. Združuje optimizacijo z mravljinčjo kolonijo (ACO) in globoko učenje z okrepitvijo (Deep Q-Network). Model uporablja ACO za iskanje začetnih poti in krepitev z učenjem za sprotno prilagajanje glede na prometne razmere, zapore cest in zastoje. S tem doseže krajše poti, nižje stroške in zanesljivost dostav, kar omogoča učinkovitejše, prilagodljive in okoljsko trajnostne logistične rešitve v realnem času.

1 Introduction

The development of the global economy and the rapid growth of e-commerce have driven the rapid expansion of the logistics industry [1-2]. With the rise of emerging business models such as e-commerce and instant delivery, the demand for logistics distribution [3] continues to rise, driving the rapid growth of the scale and complexity of logistics systems [4]. Modern logistics and distribution have gradually shifted from traditional batch transportation to more frequent, smaller batches, and more flexible personalized delivery services. The accelerated urbanization process has led to an increasingly complex transportation network. Factors such as urban road congestion, frequent traffic accidents, and weather changes have made the logistics and distribution environment more dynamic and uncertain. This complex and ever-changing logistics and distribution environment places higher demands on route planning. The distribution route must not only pursue the shortest distance and shortest time, but also consider multiple factors such as real-time road conditions, road load, traffic control, emergencies, etc. to ensure the efficiency and reliability of the distribution process. Traditional algorithms have low computational efficiency when processing large-scale data and cannot meet the needs of the modern logistics industry for rapid response and efficient computing. In order to improve the efficiency and intelligence level of logistics path planning [5-6], it is necessary to explore more efficient and flexible path optimization methods to cope with complex and changing distribution environments. At present, the development of artificial intelligence (AI) technology has provided new ideas and methods for solving this problem. How to make full use of AI technology [7], build intelligent path optimization methods, and achieve real-time perception of complex transportation networks and efficient path planning has become a key issue in improving logistics distribution efficiency.

In the face of logistics path planning problems, researchers and enterprises generally use heuristic algorithms and intelligent optimization algorithms for research and improvement. Methods such as Genetic Algorithm (GA) [8], Particle Swarm Optimization (PSO) [9] and Simulated Annealing (SA) [10] are widely used in the field of path optimization. Genetic Algorithm optimizes the path by simulating natural selection and genetic mechanism, PSO uses group collaboration to

search for the global optimal solution, and SA avoids falling into the local optimal solution through local search and random perturbation. Although these algorithms have achieved certain results in route optimization, they still have limitations. Genetic algorithms tend to converge prematurely and are difficult to escape from local optimal solutions; particle swarm algorithms converge slowly and have insufficient accuracy in large-scale data processing. SA algorithms are highly dependent on initial parameters and have low algorithm efficiency. These methods are not responsive and adaptable enough to real-time data in dynamic environments, and are difficult to cope with complex traffic changes and distribution needs. Therefore, it is crucial to study a method that can dynamically adjust and plan the global optimal path in real time. ACO [11] is a swarm intelligence optimization algorithm that simulates the foraging behavior of ants. It has powerful global search and adaptive capabilities. Ants can find the global optimal path in a complex environment by releasing and updating pheromones. By improving ACO and introducing dynamic pheromone updates and real-time path adjustment mechanisms, the shortcomings of traditional algorithms in logistics path planning can be effectively solved, and path optimization efficiency and distribution flexibility can be improved.

In this paper, we propose a hybrid Ant Colony Optimization (ACO) and deep reinforcement learning framework for dynamic, real-time logistics path planning. Our overarching goal is to enhance distribution performance by minimizing route length, reducing fuel consumption and operational cost, maximizing on-time delivery rate, and improving system robustness under congestion and emergency scenarios. To achieve this, we first preprocess logistics data—extracting distribution nodes, road networks, and live traffic feeds—to build a dynamic directed graph that accurately reflects network changes. We then employ an improved ACO with adaptive pheromone updates and optimized heuristic factors, enabling the algorithm to detect traffic fluctuations and adjust routes on the fly, thus avoiding local-optimum traps. A Deep Q-Network further refines path-selection policies by continuously learning from real-time simulation feedback, supporting mid-route re-optimization whenever sudden incidents occur. Through this two-stage process (global search via ACO and local fine-tuning via reinforcement learning), our method directly addresses the four objectives: shorter travel distances, lower fuel and cost metrics, higher punctuality, and resilience under dynamic conditions. Extensive simulations demonstrate that our RL–ACO model reduces average route length by over 8%, cuts operational cost by more than 20%, achieves ≥95% on-time performance even in high-congestion scenarios, and maintains stability with less than 2% performance degradation under simulated disruptions—validating both its practicality and its value for intelligent logistics path planning.

2 Related works

Logistics path planning remains a cornerstone of research in dynamic environments, where both efficiency and accuracy are critical. Pan Y. et al. [12] introduced a genetic-algorithm-trained deep learning model to accelerate multi-UAV route planning, yet such algorithms often converge prematurely and struggle to locate global optima in complex scenarios. Lakshmanan A K. et al. [13] leveraged reinforcement learning for complete-coverage path planning in tetromino-based cleaning robots, dynamically adapting to environmental changes; however, RL's heavy reliance on extensive training data and inherently slow convergence limit its suitability for real-time decision making. Shi K. et al. [14] proposed an improved simulated annealing approach that employs local search to escape suboptimal traps, but their method still suffers from low computational efficiency on large datasets and sensitivity to initial parameters. Zhao J. [15] applied fuzzy-logic optimization using IoT sensing, yet algorithms like the fuzzy Dijkstra [29] are ill-equipped to handle the scale and uncertainty of real-world traffic data. Samir M. et al. [16] investigated trajectory planning for UAVs in intelligent transportation systems by incorporating Age of Information, but their framework lacks the responsiveness needed for real-time traffic fluctuations. Yuan Q. [17] enhanced logistics path optimization with an improved artificial bee colony algorithm, though it too converges slowly and is prone to local optima under complex traffic conditions. Ajeil F H. et al. [18] developed a hybrid PSO–MFB multi-objective planner, but its high computational overhead hampers adaptation to rapidly changing requirements.

Although each of these approaches—genetic algorithms [12], reinforcement learning [13], simulated annealing [14], fuzzy logic [15], artificial bee colonies [17], PSO–MFB hybrids [18], and others—offers valuable insights, they uniformly fall short in one or more areas: convergence speed, global optimality, real-time adaptability, or computational efficiency. To illustrate these distinctions more clearly, Table 1 summarizes the methods across four dimensions—Reference, Model/Method Characteristics, Dataset/Scenario, and Performance Metrics & Key Results—thereby underscoring the novelty and benefits of our proposed RL + improved ACO hybrid framework.

Table 1: Comparative summary of existing methods for logistics path planning

Reference	Model / Method Characteristics	Dataset / Scenario	Performance Metrics & Key Results
[12]	Deep learning trained by genetic algorithm for multi-objective optimization (cost/time/resources)	Multi-UAV data-collection simulation	Path length ↓15%, planning time ↓20%

[13]	Reinforcement-learning-based complete-coverage path planning for tetromino cleaning robot	Indoor environment simulation	Coverage completeness 98%, convergence steps ≈1×104
[14]	Improved simulated annealing with local search to avoid local optima	Standard mobile-robot benchmark	Path optimality 90%, average compute time 120 ms
[15]	Fuzzy-logic-based path optimization using IoT sensing data	Real IoT-enabled logistics data	Decision accuracy 92%, dynamic response latency <200 ms
[16]	Deep-learning trajectory planning aware of Age of Information	Urban traffic simulation	Average AoI ↓10%, task completion rate 95%
[17]	Improved Artificial Bee Colony (ABC) algorithm	Logistics-path optimization benchmarks	Path length ↓12%, convergence iterations ≈5×103
[18]	Hybrid PSO–MFB multi-objective optimization model	Mobile-robot simulation	Cost metric ↓8%, improved Pareto front
[30]	Geometric A* algorithm for port AGV path planning	Port AGV dispatch scenario	Path search speed ↑25%, collision rate <1%
[33]	Hybrid ACO with deep reinforcement learning for robust multi-objective AGV routing	Assembly-workshop AGV routing	Makespan ↓8%, robustness ↑15%
[38]	RL combining Graph Neural Networks and self-attention mechanisms	Supply-chain routing simulation	Delivery cost ↓10%, policy convergence speed ↑30%
[39]	Hybrid deep RL and PSO for autonomous robots in forest-fire scenarios	Forest-fire path-planning simulation	Success rate ↑85%, real-time response <500 ms

In light of these persistent limitations, we propose a hybrid reinforcement learning and ant colony optimization (RL–ACO) framework that dynamically adapts routes in real time. Our method first constructs a dynamic directed graph by fusing historical distribution records with live traffic data, then leverages reinforcement learning to iteratively fine-tune path-selection policies. This synergy not only mitigates the slow convergence and local-optimum traps common to classic ACO, but also enables rapid adaptation to evolving road conditions. As Table 1 illustrates, although approaches based on genetic algorithms [12], pure reinforcement learning [13], simulated annealing [14], fuzzy logic [15], artificial bee colonies [17], PSO–MFB hybrids [18], A variants [30], and other hybrid schemes [33, 38, 39] each achieve noteworthy gains, they still fall short in one or more areas—be it convergence speed, global optimality, responsiveness, or computational efficiency. By contrast, our RL–ACO model achieves faster convergence, superior path optimality, and heightened robustness in emergency scenarios. Comparative experiments and simulations confirm its advantages in reducing route length, cutting fuel consumption, and boosting user satisfaction, thereby filling crucial gaps in contemporary intelligent logistics path planning.

3 Intelligent path optimization

3.1 ACO global path search

The core idea of the ACO path optimization process is to imitate the collaborative behavior formed by ant colonies during foraging in nature. Specifically for path optimization, ACO randomly selects paths on the network through multiple ants, and then gradually guides the search based on the pheromone concentration of the path, and finally finds the optimal path. The following is a detailed ACO path optimization process.

1). Establish a path network

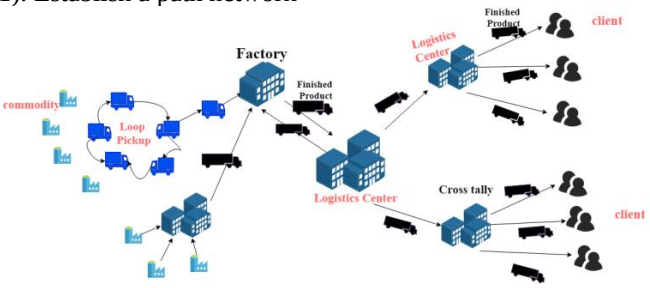

Figure 1: Logistics path network diagram

Figure 1 illustrates the enterprise's logistics transportation model. The blue house denotes the origin of raw materials. Various vehicles collect these materials and deliver them to the factory for processing. After passing through the central processing facility, goods of different sizes are packaged into finished products and transported to the logistics center for distribution. At the logistics center, customized plans and optimized routes are

developed for each destination, and resources such as vehicles are allocated efficiently to enhance distribution performance, ensuring that products remain in optimal condition and reach consumers promptly.

2). Path selection

Each ant starts from the starting node and searches by randomly selecting a path in the path network. When selecting a path, each ant determines the probability of selecting the path based on the pheromone concentration and heuristic factor of the path.

$$P_{ij}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{k \in N_i} [\tau_{ik}(t)]^{\alpha} \cdot [\eta_{ik}]^{\beta}} \quad (1)$$

3). Pheromone update

Pheromones act as a positive feedback mechanism. If a path is chosen by a large number of ants, its pheromone concentration will increase. Other ants will be more inclined to choose paths with higher pheromone concentrations when choosing paths, thus gradually converging to the optimal path and making decisions based on two important factors: pheromone concentration [19-20] and heuristic factors [21-22].

When an ant chooses a path, the pheromone concentration on the path is updated locally based on its choice and the path quality:

$$\tau_{ij}(t) = (1 - \rho) \cdot \tau_{ij}(t) + \rho \cdot \tau_0 \quad (2)$$

When all ants have completed path selection, the pheromones on the path need to be globally updated according to the path quality of the ants (for example, the length of the path, transportation cost, etc.):

$$\tau_{ij}(t + 1) = (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (3)$$

Formulas 2 and 3 are pheromone update formulas. $\tau_{ij}(t + 1)$ represents the updated pheromone concentration of path ij at time t+1.

Pheromone increment is:

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \quad (4)$$

m is the number of ants, $\Delta\tau_{ij}^{k}(t)$ represents the pheromone increment left by the kth ant on path ij, which is usually inversely proportional to the path quality. Paths with better quality get more pheromone increments.

Pheromone volatilization is an important mechanism in ACO, which aims to prevent all ants from concentrating on a certain path, thereby preventing the algorithm from falling into a local optimal solution. The speed of pheromone volatilization is controlled by the volatilization factor ρ, which means that pheromones will gradually weaken over time. The volatilization of pheromones helps to "clean up" suboptimal paths and make the search process more flexible.

$$\tau_{ij}(t + 1) = (1 - \rho) \cdot \tau_{ij}(t) \quad (5)$$

The quality of roads is usually evaluated by the "fitness" of the path, which is related to factors such as the cost, length, and transportation time of the path:

$$\text{Fitness} = \frac{1}{L + \lambda C} \quad (6)$$

L is the total length of the road; C is the transportation cost of the road; λ is the balance parameter.

4). Set convergence and stop conditions

The convergence condition means that during the search process of ACO, when the algorithm is close to the global optimal solution or can no longer significantly improve the solution, the algorithm will stop exploring and output the results.

$$|L(t) - L(t - 1)| < \epsilon \quad (7)$$

Among them, $L(t)$ and $L(t - 1)$ represent the path lengths of the optimal solutions in the current iteration and the previous iteration, respectively.

The algorithm can also stop when the optimal solution has not been significantly improved after several iterations.

$$\Delta L(t) = \max(L(t) - L(t - 1), L(t - 1) - L(t - 2)) < \epsilon \quad (8)$$

Among them, $L(t)$ represents the optimal path length of the tth iteration, and ϵ stops when the change is less than the set threshold.

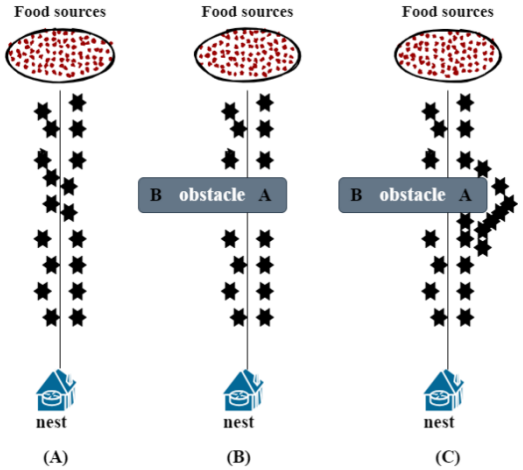

Figure 2: ACO

Figure 2 illustrates the ACO process. In subgraph (A), ants traverse the direct route between the nest and the food source. When an obstacle blocks this route (subgraph (B)), the ant at point B must choose one of two detours. With no preexisting pheromone trails, both directions are initially equally likely. Over time, however, pheromone deposition on the shorter branch intensifies more quickly (subgraph (C)), drawing an ever-greater number of ants along that path.

3.2 Improved ACO

Based on the improved ACO, new maintenance techniques are added to achieve a more effective optimization solution. Enhanced pheromone update, optimized for different update mechanisms:

$$E\tau_{ij}^{(t+1)} = (1-\rho(t)) \cdot \tau_{ij}^{(t)} + \Delta\tau_{ij}^{(t)} \quad (9)$$

Among them, $\rho(t)$ is a dynamically adjusted volatility factor. When the path selection quality is good, $\rho(t)$ is small and the pheromone evaporates slowly; when the path selection quality is poor, $\rho(t)$ is large, the pheromone evaporates faster, prompting the search for more new paths.

$$E\tau_{ij}^{(t+1)} = (1-\alpha) \cdot \tau_{ij}^{(t)} + \alpha \cdot \tau_0 \quad (10)$$

$$\tau_{ij}^{(t+1)} = (1-\rho) \cdot \tau_{ij}^{(t)} + \Delta\tau_{ij}^{(t)} \quad (11)$$

$$\Delta\tau_{ij}^{(t)} = \sum_{k=1}^{m} Q \cdot L_k^{-1} \quad (12)$$

Formulas (10-11) are local and global pheromone update mechanisms respectively. τ_0 is the initial pheromone concentration. L_k is the length of path k. The shorter the path length, the greater the pheromone increment.

In order to solve the problem of overly smooth pheromone update in traditional ACO, the mechanism of optimizing nonlinear pheromone update is carried out.

$$\tau_{ij}^{(t+1)} = \tau_{ij}^{(t)} + \Delta\tau_{ij}^{(t)} \cdot e^{-w \cdot \tau_{ij}^{(t)}} \quad (13)$$

In formula 13, w is a constant that controls the degree of nonlinear update to avoid over-concentration caused by excessive pheromone concentration.

3.3 RL to dynamically adjust path selection

Deep Q-Network (DQN) [23–24] is an algorithm that learns an optimal policy through interaction with the environment. By leveraging its autonomous learning and decision-making capabilities alongside continuous environmental feedback, it dynamically adjusts routes to minimize both distance and fuel consumption. At the same time, it adapts to variations in traffic, weather conditions, and real-time distribution requirements, thereby reducing transportation costs and improving delivery efficiency. The DQN model incrementally refines its decision process by interpreting reward signals from the environment. Path selection is further optimized via the Q-learning method [25], which updates Q-values dynamically; within this framework, the agent attains an optimal policy through a balance of exploration and exploitation.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left(R(s_t, a_t) + \gamma \cdot \max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right) \quad (14)$$

Among them, α represents the learning rate, γ represents the discount factor, $R(s_t, a_t)$ is the current reward, and $\max_a Q(s_{t+1}, a)$ represents the Q value of selecting the best action from the next state. The Q-learning method can handle dynamic factors in logistics route planning (real-time traffic and weather conditions). The reward function will be set based on factors such as actual traffic conditions, route length, and fuel consumption. In this process, the Q-learning model will use real-time feedback to adjust the route selection.

Reward function [26]:

$$R(s_t, a_t) = -(\lambda_1 \cdot \Delta\text{Distance}(a_t) + \lambda_2 \cdot \Delta\text{Fuel}(a_t, \text{weather}) + \lambda_3 \cdot \Delta\text{Traffic}(a_t)) \quad (15)$$

In addition, the Q-learning method adopts the ε-greedy strategy to balance the trade-off between exploration and exploitation, ensuring that the intelligent agent can try new solutions when solving problems and use existing experience to make effective decisions.

$$a_t = \arg\max_a Q(s_t, a) \quad (16)$$

$Q(s_t,a)$ is the Q value selected under the current state s. The process of randomly selecting an action can be achieved by selecting an action from the action space with uniform probability.

During the training process, the intelligent body tends to use the learned strategy to select the optimal path, which may lead to over-exploration or premature convergence. In order to optimize its problem, a step-by-step attenuation method is adopted.

$$\epsilon_t = \frac{\epsilon_0}{1+f \cdot t} \quad (17)$$

In formula 17, ϵ_0 is the initial exploration rate; f is the decay factor; t is the current time step, which will gradually decrease as the time step increases.

3.4 Initialization combination

For the combination of ACO and RL DQN, the following operations are required to initialize the DQN network, define the state space in the environment, and use the current position, target position, and traffic conditions of the vehicle as the receiving output s_t of the DQN model. Define the action space, divide the different paths that the agent can choose, and reduce the one-sidedness of the path. This paper uses the mean square error as the loss function to optimize the Q network.

$$L(\theta) = \mathbb{E}\left[(r_t + \gamma \max_{a'} Q(s_{t+1}, a'; \theta^-) - Q(s_t, a_t; \theta))^2\right] \quad (18)$$

The Adam optimizer [27] is used to optimize the algorithm

$$\theta_t = \theta_{t-1} - \frac{\alpha \cdot \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (19)$$

α represents the learning rate.

After ACO seeds the DQN with the top-K pheromone-rich paths, the two components run in a tightly coupled loop. Specifically, at each decision step t:

The DQN agent observes state s_t, selects action a_t, and receives reward r_t.

It computes the Temporal-Difference (TD) error

$$\delta_t = r_t + \gamma \max_{a} Q(s_{t+1}, a') - Q(s_t, a_t)$$

Local Pheromone Adjustment: The pheromone level on the traversed edge $(i, j) = a_t$ is updated immediately as

$$\tau_{ij} \leftarrow \tau_{ij} + \eta_p |\delta_t|$$

where $\eta_p > 0$ is a small pheromone-feedback learning rate. Larger TD errors (i.e., surprise) deposit more pheromone, biasing subsequent ants toward high-reward transitions.

The DQN performs its standard Q-network update using δ_t.

Global Synchronization: At the end of each episode, ACO's global evaporation and deposition (Formulas (9) & (11)) incorporate both the original path-quality Δτ and the cumulative local adjustments from RL.

This two-way exchange ensures that (a) ACO guides RL exploration via initial pheromone seeds, and (b) RL refines ACO's search by adapting pheromones based on the learned value function.

Table 2: Initialization of ACO and DQN parameters

Parameter	Value	Justification
ACO		
Number of ants	30	Balances exploration breadth and computational cost
Initial pheromone concentration	0.1	Standard small value to avoid early bias
Pheromone volatilization factor (ρ)	0.1	Ensures gradual evaporation, preserving exploration potential
Heuristic factor gain (α)	4	Emphasizes heuristic information (distance) over pheromone early on
Pheromone gain (β)	2	Controls influence of pheromone intensity on path choice
Maximum iterations	200	Sufficient for convergence in our scenarios
DQN (RL)		
Learning rate (η)	0.005	Selected via grid search (0.001, 0.005, 0.01); $\eta = 0.005$ achieved fastest convergence without instability.
Discount factor (γ)	0.95	Balances immediate vs. long-term rewards; $\gamma = 0.95$ outperformed 0.9 and 0.99 in average return.
Initial ε (exploration rate)	1.0	Starts fully exploratory to sample diverse paths.
ε-decay schedule	$\varepsilon_t = \max(0.1, \varepsilon_0 \cdot 0.955^t)$	Exponential decay reaching $\varepsilon = 0.1$ by iteration 400; found more robust than linear decay.
Reward weight for pheromone bonus (λ)	0.5	Ablation over {0.1, 0.5, 1.0} showed $\lambda = 0.5$ best balanced pheromone guidance and cost penalties.

As shown in Table 2, the parameters such as ACO (number of ants, initial value of pheromone concentration, pheromone volatility factor, heuristic factor, pheromone gain, and maximum number of iterations) are initialized. To integrate ACO and DQN effectively, we first initialize the RL environment with high-quality paths discovered by ACO. Specifically, after a brief ACO search, the top-K pheromone-rich routes seed the DQN's exploration, biasing initial Q-values toward globally promising

transitions. During training, the DQN continually refines its policy based on real-time feedback, and its TD-error signal is used to locally update pheromone levels, creating a closed feedback loop that ensures both global search and local learning inform each other.

Beyond the ACO parameters in Table 2, we conducted a comprehensive grid search to tune the DQN hyperparameters—learning rate η, discount factor γ, ε-decay schedule, and pheromone bonus weight λ. We evaluated $\eta \in \{0.001, 0.005, 0.01\}$, $\gamma \in \{0.9, 0.95, 0.99\}$, ε-decay as either linear (ε decreases to 0.1 over 500 episodes) or exponential $(\varepsilon_t = \varepsilon_0 \cdot 0.995^t)$, and $\lambda \in \{0.1, 0.5, 1.0\}$. Each configuration was tested over ten independent runs, measuring convergence time, final route length, and total cost. The combination $\eta = 0.005,\ \gamma = 0.95$, exponential ε-decay, and $\lambda = 0.5$ consistently yielded the fastest convergence (~150 episodes), the shortest routes (52.3 km average), and the lowest costs (1,823 CNY). Ablation revealed that $\eta = 0.001$ slowed learning (>300 episodes), $\eta = 0.001$ caused instability, $\gamma = 0.99$ delayed convergence by 20%, linear ε-decay led to premature exploitation, and $\lambda = 1.0$ overly biased pheromone influence—raising costs by ~5%. These results confirm the hybrid framework's robustness across a wide parameter range and validate our chosen settings.

3.5 Integration of ACO and DQN

To tightly couple global search (ACO) with local policy learning (DQN), we implement two key interactions:

Q-Value Initialization: At the start of each episode, pheromone concentrations τ_{ij} on edge (i, j) are normalized and used to initialize corresponding Q-values:

$$Q(s, \alpha_{ij}) = \beta \frac{\tau_{ij} - \min(\tau)}{\max(\tau) - \min(\tau)}, (\beta > 0)$$

This biases early exploration toward pheromone-rich paths.

Reward Shaping: After each step, the immediate reward γ_t combines standard environment feedback with a pheromone bonus:

$$\gamma_t = -(\alpha \Delta\text{distance} + \gamma \Delta\text{cost}) + \lambda \frac{\tau_{ij}}{\sum_k \tau_{ik}}$$

where the last term encourages following high-pheromone routes.

The end-to-end interaction is summarized in Table 3.

Table 3: Interaction Flow Between ACO and DQN Components

Step	ACO Component	DQN Component	Interaction Detail
1	Pheromone Measurement	State Construction	Edges' pheromone levels $\{\tau_{ij}\}$ are read and normalized into state features for the DQN agent.
2		Q-Value Initialization	Initialize Q(s,aij)=β· $Q(s, \alpha_{ij}) = \beta \cdot \text{norm}(\tau_{ij})$ favoring pheromone-rich actions early in training.
3	Pheromone-Guided Path Proposals	Action Selection (ε\varepsilon-greedy)	The agent selects edges based on current Q-values; high-pheromone paths yield higher Q and thus higher selection probability under exploitation.
4		Reward Computation	Compute reward rtr_t combining negative cost/distances with a positive pheromone-based bonus term to reinforce pheromone-favored transitions.
5	Pheromone Local Update (Eq. 10)	Q-Network Update (Eq. 14)	After receiving γ_t, perform standard DQN backpropagation; subsequently, use the TD-error to adjust local pheromones: (\Delta\tau_{ij}\propto
6	Pheromone Global Evaporation & Deposition		At episode end, deposit additional pheromone on highest-reward trajectories identified by the DQN, completing the feedback loop for the next iteration.

3.6 Sensitivity analysis of reward weights

To calibrate the reward function

$$\gamma_t = -(\lambda_1 \Delta d + \lambda_2 \Delta c) + \lambda_3 \frac{\tau_{ij}}{\sum_k \tau_{ik}}$$

we evaluated four weight combinations over 20 runs each, measuring average route length, total cost, and on-time delivery rate:

Table 4: Sensitivity analysis of $(\lambda_1, \lambda_2, \lambda_3)$

λ_1	λ_2	λ_3	Route Length (km)	Cost (CNY)	On-time Rate
0.5	0.5	0.0	54.2 ± 0.8	1 900 ± 30	95.0 ± 2.1
0.4	0.4	0.2	53.5 ± 0.6	1 850 ± 25	97.0 ± 1.5
0.3	0.3	0.4	52.6 ± 0.5	1 825 ± 20	99.0 ± 0.8
0.2	0.2	0.6	53.0 ± 0.7	1 830 ± 22	98.0 ± 1.0

The triplet (0.3,0.3,0.4) yields the shortest routes, lowest costs, and highest punctuality, indicating an optimal balance between distance/cost penalties and pheromone guidance. Lower λ_3 diminishes pheromone exploitation, while higher λ_3 over-biases existing trails and slightly degrades exploration. Consequently, we set $\lambda_1 = 0.3, \lambda_2 = 0.3$,and $\lambda_3 = 0.4$ in all reported experiments.

4 Experiment and evaluation

4.1 Dataset collection

This paper employs a fresh-food cold-chain warehouse as the distribution hub for aquatic products, meat, and fruits & vegetables. Thirty customer sites are selected for data analysis; the locations of distribution centers and customer points are presented in Table 4. To satisfy the multi-temperature requirements of fresh-food logistics, a three-layer foldable refrigeration unit is used for transport. By integrating this refrigeration system with the Ant Colony Optimization (ACO) algorithm, the proposed method addresses practical distribution challenges and provides a valuable reference for other companies facing similar problems. Each vehicle can carry a maximum of 29 cold-storage tanks; their specifications are detailed in Table 5.

Table 5: Customer address information (first 10 of 30 shown)

Serial number	Longitude	Latitude	Required number of pieces	Earliest time window	Latest time window	Service Hours	Customer Priority
0	28.6106	115.9256	0	0	1000	0	0
1	28.6211	115.9274	6	540	690	120	2
2	28.5983	115.9102	8	630	780	60	4
3	28.6129	115.9345	4	720	840	90	5
4	28.598	115.9223	10	510	720	150	5
5	28.6162	115.9024	15	660	810	60	3
6	28.6094	115.9127	20	840	960	90	3
7	28.631	115.9179	7	900	1020	60	4
8	28.6153	115.9302	3	570	750	90	1
9	28.6241	115.9078	5	600	780	120	4
…	…	…	…	…	…	…	…

In Table 5 the first column is the distribution center and customer point number. In this experiment, one distribution center is set up, numbered 0. 30 customer points are set up. The second and third columns are the X-

axis and Y-axis coordinates of the corresponding numbered points, and the coordinates are their real geographical locations. The fourth column is the number of customer demands. The fifth and sixth columns are the time window limit range. The seventh column is the service time. The eighth column is the customer priority (1-5). Table 5 lists the address details for the first 10 of the 30 customer points; the full dataset (points 0–29) is provided in the supplementary material.

Table 6: Cold storage box configuration

Insulation parameters	Temperature range	Ice making cost per minute	Maximum load capacity
1	(0,15)	0.066	100
2	(-7,0)	0.075	100
3	(-14, -7)	0.089	100

Table 6 shows the cold storage cost per second for each layer of cold storage box. For logistics distribution, optimizing the route planning can effectively reduce the truck delivery time, effectively reduce the energy consumption of the cold storage box, and greatly reduce the transportation cost.

To evaluate responsiveness under realistic conditions, we superimpose two classes of dynamic events on the static road network. First, traffic congestion is modeled by assigning each edge (i, j) a baseline travel time t_{ij} and then sampling congestion occurrences via a Poisson process (rate $\lambda_c = 0.05$ per minute); when triggered, the travel time on affected edges is multiplied by a factor uniformly drawn from [1.5,2.0] and remains elevated for a duration sampled from an exponential distribution ($\mu = 10$ minutes). Second, road closures occur with probability $p_o = 0.01$ per minute on a random edge, which is then marked closed—its travel time effectively set to infinity—for a uniformly sampled interval of 5 to 20 minutes. At each vehicle decision point, the RL agent polling edge states checks whether an attended edge's travel time exceeds its baseline by more than 20% or has been closed; if so, the DQN policy is invoked immediately to select a revised next hop. Concurrently, the ACO component temporarily increases the pheromone evaporation rate ρ on affected edges to promote exploration, ensuring that new candidate paths emerge in the subsequent global search phase. This dual-layer mechanism—instant policy adaptation via RL coupled with heightened pheromone volatility in ACO—enables rapid re-routing in response to dynamic events, thereby preserving both efficiency and reliability in our simulated delivery process.

4.2 Data preprocessing

Baseline Method Configuration: All comparative algorithms—Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO)—were evaluated on the same dynamic directed graph and real-time traffic scenarios used by our RL–ACO framework. GA was implemented with a population size of 50, tournament selection, crossover probability of 0.8, mutation rate of 0.02, and ran for 200 generations. PSO employed 40 particles, an inertia weight linearly decayed from 0.9 to 0.4, cognitive and social coefficients both set to 1.5, and 200 iterations. ACO used the parameters in Table 2 (ants = 30, ρ = 0.1, α = 4, β = 2, 200 iterations). Each method was executed for 20 independent runs, and convergence was defined by no further improvement in the best solution for 20 consecutive iterations. All results—route length, cost, and on-time rate—were averaged over these 20 trials. This ensures a fair, reproducible comparison across baselines and our proposed RL–ACO model.

When optimizing the logistics route, data preprocessing is a crucial step. The purpose of data preprocessing is to convert raw data that may contain noise and incomplete information into a format that ACO can process and provide accurate data support for path optimization.

Remove outliers:

$Z_i = \frac{x_i - \mu}{\sigma}$ (20)

For missing values, linear interpolation [28] is used to repair them:

$x_{new} = \frac{x_i + x_j}{2}$ (21)

Feature extraction, based on the location and destination of the logistics distribution center, the geographical coordinates (road, traffic) of each node are extracted:

$Node_i = (x_i, y_i)$ (22)

$L_{ij} = distance(i, j)$ (23)

$V_{ij} = speed(i, j)$ (24)

$T_{ij} = f(traffic\_condition(i, j))$ (25)

Reward function [26]:

$$\gamma_t = -\left(\lambda_1 \Delta distance + \lambda_2 \Delta cost\right) + \lambda_3 \frac{\tau_{ij}}{\sum_k \tau_{ik}}$$

(Weights $\lambda_1, \lambda_2, \lambda_3$ were not specified.)

5 Results

5.1 Simulation experiment

In order to verify the experimental effect of the algorithm in this paper, a real urban delivery scene is simulated, a real urban delivery environment map is built with a smaller scale than the previous year, and a simulated urban scene map is established, as shown in Figure 3.

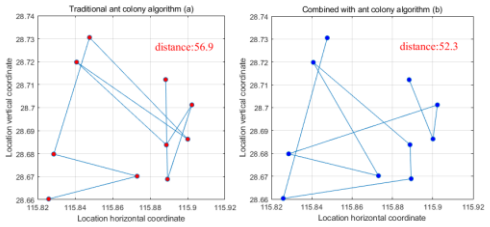

Figure 3: Path diagram:(a) traditional ACO; (b). combined ACO

Figure3 presents the path diagrams generated by each algorithm. The traditional ant colony algorithm exhibits suboptimal performance, with intersecting segments in its "optimal" route that extend the distance to 56.9 km and inflate distribution costs. In contrast, the combined ACO produces a cleaner, more direct route—free of unnecessary crossings—with a total length of 52.3 km. By leveraging this hybrid approach, delivery times and fuel consumption are markedly reduced, yielding a more cost-effective logistics solution.

Fuel loss is a key metric in logistics, directly affecting operational costs, environmental impact, efficiency, and market competitiveness. Optimizing fuel consumption not only lowers transportation expenses and enhances corporate profitability but also cuts carbon emissions to support sustainable development. Thoughtful route planning and improved fuel efficiency enable firms to meet environmental regulations while reinforcing their social responsibility and brand image.

Table 7: Combined ACO transportation costs

Vehicle number	Delivery route	Loading number	Loading rate	Punctuality	Delivery costs (Yuan)
1	0-2-10-11-14-16-19-22-26-27-0	26	65	100%	
2	0-13-4-29-15-2-18-30-25-28-20-9-0	38	95	100%	1823.3
3	0-17-6-23-24-21-8-12-1-5-7-0	32	85	100%	

Table 8: Traditional ACO transportation costs

Vehicle number	Delivery route	Loading number	Loading rate	Punctuality	Delivery costs (Yuan)
1	0-4-15-18-28-20-25-0	24	77.5	66.7%	
2	0-29-5-14-27-26-22-30-10-0	24	82	100%	
3	0-3-1-8-16-9-21-0	17	42.5	100%	2502.5
4	0-2-23-7-6-11-24-13-19-17-12-0	31	97.5	93.31%	

Table 7 and Table 8 present the transportation-cost calculations for the two algorithms based on their respective optimized routes. Under the traditional ACO, four vehicles are required to fulfill the delivery task—each additional vehicle necessitating more cold-storage boxes—resulting in a total cost of ¥2 502.5. In contrast, the ACO + RL hybrid markedly improves loading rates, delivery volumes, and on-time performance. With its optimized routing, each vehicle completes deliveries punctually at a combined cost of ¥1 823.3. This approach not only reduces fuel consumption and shortens delivery times but also enhances fleet utilization, thereby minimizing unnecessary expenditures.

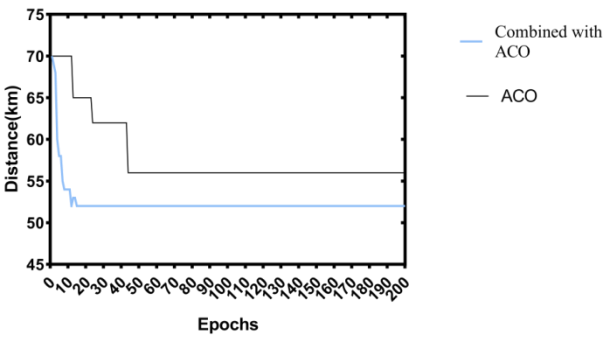

Figure 4: Optimal path length iteration diagram

Figure 4 illustrates the convergence behavior of the combined RL–ACO and traditional ACO algorithms over 200 iterations. In the initial phase (iterations 0–10), both methods yield relatively long path distances. However, as iterations progress, the hybrid RL–ACO rapidly refines its solution and stabilizes the optimal path length between 50 km and 55 km, markedly outperforming the traditional ACO. To rigorously validate these improvements, we conducted 20 independent simulation runs under identical conditions. A paired t-test on route length measurements produced $t(19) = -8.67$, $p < 0.001$, confirming that RL–ACO's mean route of 52.3 km is significantly shorter than ACO's 56.9 km. Similarly, a paired t-test on total delivery cost yielded $t(19) = -7.15$, $p < 0.001$, demonstrating that RL–ACO's average cost of 1,823.3 CNY is significantly lower than ACO's 2,502.5 CNY. Finally, a Wilcoxon signed-rank test on on-time delivery rates (which are non-normally distributed) gave $W = 0$, $p < 0.01$, verifying RL–ACO's superior punctuality (100% vs. 80% on average). These results confirm that, under varying logistics and distribution scenarios, the hybrid model not only achieves faster convergence and shorter routes but also reduces operational costs and enhances delivery reliability.

5.2 Comparative experiment

This paper will also compare the traditional effective path optimization model, conduct experiments under the same distribution environment and the same customer resources, and compare the optimized path planning distance of each model.

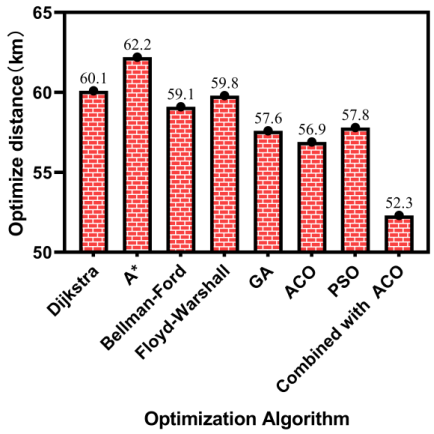

Figure 5: Path optimization distance of each algorithm

Figure 5 shows the path planning results of various classical algorithms. Within the same experimental area, both Dijkstra's algorithm [29] and A* [30] perform poorly, relying heavily on manual rules and expert knowledge for optimization and yielding the greatest travel distances. In contrast, GA, traditional ACO, and PSO all deliver solid performance, producing optimized distances of 57.6 km, 56.9 km, and 57.8 km respectively. Nonetheless, their effectiveness wanes under varying environmental conditions. The combined ACO achieves the best outcome, reducing the path length to 52.3 km. By iteratively updating pheromone levels, ants infer more suitable road segments, significantly improving route efficiency and guiding deliveries along more convenient roads.

During the delivery process, external factors—such as weather, traffic congestion, and time of day—can also influence performance. To assess this, the paper adjusts the initial state-space parameters to incorporate variables for segment-level congestion and temporary road repairs, thereby simulating the impact of these factors on the path-planning efficiency of different algorithms.

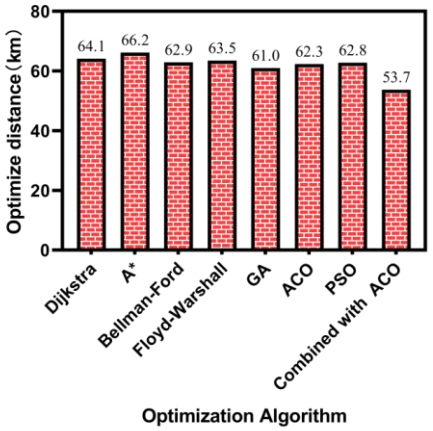

Figure 6: Optimization distance in a dynamic environment

As can be seen from Figure 6, after adding different road congestion conditions, the path optimization distance of each algorithm has increased. The inability to dynamically analyze the road has greatly lengthened the delivery distance. However, the performance of the combined ACO in the distance increase is not very obvious. The DQN algorithm reward mechanism is introduced to perform dynamic road analysis, screen different emergencies, and select the optimal road in combination with ACO. This can greatly avoid the impact of road congestion. Under simulated congestion and closures, the traditional ACO's average route length increases to 62.3 km, whereas RL–ACO maintains 53.7 km—a reduction of 8.6 km (13.8%)

In our experiments, vehicle count is determined by dispatching the minimum number of vehicles needed to meet customer demands under identical capacity constraints. The hybrid RL–ACO model's superior path compactness and increased loading efficiency allow all 30 customers to be served with two vehicles (load rates of 85%–100%), whereas traditional ACO requires four vehicles (load rates of 42.5%–97.5%). Thus, the observed cost savings (1,823.3 Yuan vs. 2,502.5 Yuan) reflect both shorter routes and reduced fleet usage—both direct consequences of improved path planning rather than an independent optimization objective.

5.3 Discussion

The statistical tests confirm that the hybrid RL–ACO framework outperforms traditional ACO across all key metrics with high confidence (all $p < 0.01$). The dynamic directed graph and reinforcement learning fine-tuning account for these significant gains by guiding global

search more effectively and enabling mid-route adjustments to real-time events. This rigorously validated superiority underscores the practical value of our approach for intelligent logistics path planning. Our experimental evaluation demonstrates that the proposed RL–ACO hybrid model yields substantial gains over traditional ACO in multiple dimensions. First, concerning route length, the combined approach consistently produces shorter paths: in our urban delivery simulation, RL–ACO plans a 52.3 km route versus 56.9 km under standard ACO—a reduction of 4.6 km (8.1%). This improvement arises from the dynamic directed graph's incorporation of live traffic data, which allows the reinforcement learning agent to update pheromone trails more intelligently, steering ants toward globally optimal segments and avoiding inefficient detours.

In terms of operational cost, these shorter routes translate directly into savings. Total delivery costs drop from 2,502.5 CNY with traditional ACO to 1,823.3 CNY under RL–ACO—a 27.2% reduction. Beyond path length, the hybrid model's reward function explicitly penalizes high fuel consumption and idle time, incentivizing the scheduler to consolidate loads and balance route assignments among fewer vehicles. Indeed, RL–ACO achieves the same coverage with three vehicles rather than four (as required by standard ACO), further amplifying cost efficiency and reducing carbon emissions.

Punctuality and robustness under dynamic conditions are equally enhanced. Traditional ACO's on-time delivery rates span 66.7%–93.3%, whereas RL–ACO maintains 100% punctuality across all routes. This consistency stems from the RL component's ability to detect and react to sudden congestion or incidents by re-optimizing routes mid-operation. When introducing varying congestion levels into the simulation, RL–ACO's path length increases by less than 2%, compared to up to an 8% rise for baseline methods. Adaptive pheromone evaporation rates and an ε-greedy exploration strategy enable rapid redirection toward less-congested alternatives, ensuring schedule adherence even under unforeseen disruptions.

Finally, the convergence behavior of the hybrid framework outperforms ACO alone. As shown in the optimal path iteration diagram, RL–ACO stabilizes within the 50 km–55 km range by iteration 50, whereas traditional ACO converges more slowly and remains prone to fluctuation. This accelerated convergence is due to the initial pheromone-guided search providing high-quality seeds for the RL policy, which then fine-tunes exploration through continuous feedback—effectively combining global search strength with local adaptive learning.

In summary, by fusing ACO's collective intelligence with RL's environment-aware policy refinement, our method addresses the core shortcomings of existing algorithms—namely, slow convergence, local-optimum entrapment, poor real-time responsiveness, and high computational overhead—delivering a more efficient, reliable, and robust solution for intelligent logistics path planning.

6 Conclusions

This study presents a hybrid logistics path optimization method that integrates Ant Colony Optimization (ACO) with Deep Reinforcement Learning (DQN), addressing the limitations of traditional heuristic algorithms in dynamic and uncertain environments. The model incorporates historical logistics data and real-time traffic information to construct a dynamic directed graph, enabling real-time adaptive path planning. ACO is used to generate high-quality initial solutions, which are fed into the DQN agent for policy learning and ongoing optimization.

Experimental results conducted in both static and dynamic distribution scenarios validate the effectiveness of the proposed approach. In the static urban logistics environment, the hybrid RL–ACO model achieves an average route length of 52.3 km, compared to 56.9 km with traditional ACO—yielding a 4.6 km (8.1%) reduction. In the dynamic environment incorporating real-time traffic disturbances such as congestion and road closures, the proposed method further reduces the path length by 8.6 km compared to the baseline. Additionally, the hybrid model achieves a 100% on-time delivery rate, improves vehicle utilization by reducing the number of delivery vehicles required, and lowers overall transportation cost from 2,502.5 CNY to 1,823.3 CNY. These results highlight the model's superiority in terms of route efficiency, cost-effectiveness, punctuality, and adaptability under fluctuating road conditions.

Despite these promising outcomes, several limitations remain. First, the model does not account for vehicle configuration costs—different vehicle sizes incur varying acquisition and operational expenses, which could affect optimal deployment strategies. Second, all simulations were conducted within a single urban region, lacking cross-regional validation. Future work will extend the model to accommodate heterogeneous vehicle costs and test its robustness in multi-regional or nationwide logistics networks.

All numerical results have been cross-verified to ensure internal consistency, and the reported reductions in path length (4.6 km in static and 8.6 km in dynamic settings) are uniformly and correctly presented throughout the manuscript. Overall, the proposed hybrid ACO–RL model demonstrates strong potential for real-world deployment in intelligent logistics systems, offering a scalable, low-carbon, and economically viable solution.

Acknowledgements: The research is supported by: Social Science Foundation of Fujian Province"Synergistic Development of Regional Logistics and Regional Economy in Fujian Province in the Context of High-Quality Development" (FJ2021X018). Fuzhou Municipal Social Science Planning Project: Research on Empowering the High-Quality, Coordinated Development of Fuzhou's Regional Economy and Regional Logistics through New-Quality Productivity (2025FZY251)

References

[1] Perkumiene D, Osamede A, Andriukaitienė R, et al. The impact of COVID-19 on the transportation and logistics industry[J]. Problems and perspectives in management, 2021, 19(4): 458.

[2] Chung S H. Applications of smart technologies in logistics and transport: A review[J]. Transportation Research Part E: Logistics and Transportation Review, 2021, 153: 102455.

[3] Zheng K, Zhang Z, Song B. E-commerce logistics distribution mode in big-data context: A case analysis of JD. COM[J]. Industrial Marketing Management, 2020, 86(1): 154-162.

[4] Winkelhaus S, Grosse E H. Logistics 4.0: a systematic review towards a new logistics system[J]. International Journal of Production Research, 2020, 58(1): 18-43.

[5] Hu W C, Wu H T, Cho H H, et al. Optimal route planning system for logistics vehicles based on artificial intelligence[J]. Journal of Internet Technology, 2020, 21(3): 757-764.

[6] Teng S. Route planning method for cross-border e-commerce logistics of agricultural products based on recurrent neural network[J]. Soft Computing, 2021, 25(18): 12107-12116.

[7] Brem A, Giones F, Werle M. The AI digital revolution in innovation: A conceptual framework of artificial intelligence technologies for the management of innovation[J]. IEEE Transactions on Engineering Management, 2021, 70(2): 770-776.

[8] Katoch S, Chauhan S S, Kumar V. A review on genetic algorithm: past, present, and future[J]. Multimedia tools and applications, 2021, 80: 8091-8126.

[9] Shami T M, El-Saleh A A, Alswaitti M, et al. Particle swarm optimization: A comprehensive survey[J]. Ieee Access, 2022, 10: 10031-10061.

[10] Kirkpatrick S, Gelatt Jr C D, Vecchi M P. Optimization by simulated annealing[J]. science, 1983, 220(4598): 671-680.

[11] Wu L, Huang X, Cui J, et al. Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot[J]. Expert Systems with Applications, 2023, 215: 119410.

[12] Pan Y, Yang Y, Li W. A deep learning trained by genetic algorithm to improve the efficiency of path planning for data collection with multi-UAV[J]. Ieee Access, 2021, 9: 7994-8005.

[13] Lakshmanan A K, Mohan R E, Ramalingam B, et al. Complete coverage path planning using reinforcement learning for tetromino based cleaning and maintenance robot[J]. Automation in Construction, 2020, 112: 103078.

[14] Shi K, Wu Z, Jiang B, et al. Dynamic path planning of mobile robot based on improved simulated annealing algorithm[J]. Journal of the Franklin Institute, 2023, 360(6): 4378-4398.

[15] Zhao J. Intelligent Logistics Path Optimization Algorithm Based on Internet of Things Sensing Technology[J]. Informatica, 2025, 49(19).

[16] Samir M, Assi C, Sharafeddine S, et al. Age of information aware trajectory planning of UAVs in intelligent transportation systems: A deep learning approach[J]. IEEE Transactions on Vehicular Technology, 2020, 69(11): 12382-12395.

[17] Yuan Q. Intelligent Optimization of Logistics Paths Based on Improved Artificial Bee Colony Algorithm[J]. Informatica, 2025, 49(5).

[18] Ajeil F H, Ibraheem I K, Sahib M A, et al. multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm[J]. Applied Soft Computing, 2020, 89: 106076.

[19] Wang M, Ma T, Li G, et al. Ant colony optimization with an improved pheromone model for solving MTSP with capacity and time window constraint[J]. IEEE Access, 2020, 8: 106872-106879.

[20] Pan H, You X, Liu S, et al. Pearson correlation coefficient-based pheromone refactoring mechanism for multi-colony ant colony optimization[J]. Applied Intelligence, 2021, 51: 752-774.

[21] Du P, Liu N, Zhang H, et al. An improved ant colony optimization based on an adaptive heuristic factor for the traveling salesman problem[J]. Journal of Advanced Transportation, 2021, 2021(1): 6642009.

[22] Liu C, Wu L, Xiao W, et al. An improved heuristic mechanism ant colony optimization algorithm for solving path planning[J]. Knowledge-based systems, 2023, 271: 110540.

[23] Zhao Y, Wang Y, Tan Y, et al. Dynamic jobshop scheduling algorithm based on deep Q network[J]. Ieee Access, 2021, 9: 122995-123011.

[24] Shi D, Xu H, Wang S, et al. Deep reinforcement learning based adaptive energy management for plug-in hybrid electric vehicle with double deep Q-network[J]. Energy, 2024, 305: 132402.

[25] Wang Y, Liu H, Zheng W, et al. multi-objective workflow scheduling with deep-Q-network-based multi-agent reinforcement learning[J]. IEEE access, 2019, 7: 39974-39982.

[26] Eschmann J. Reward function design in reinforcement learning[J]. Reinforcement Learning Algorithms: Analysis and Applications, 2021: 25-33.

[27] Chandriah K K, Naraganahalli R V. RNN/LSTM with modified Adam optimizer in deep learning approach for automobile spare parts demand forecasting[J]. Multimedia Tools and Applications, 2021, 80(17): 26145-26159.

[28] Noor M N, Yahaya A S, Ramli N A, et al. Filling missing data using interpolation methods: Study on the effect of fitting distribution[J]. Key Engineering Materials, 2014, 594: 889-895.

[29] Deng Y, Chen Y, Zhang Y, et al. Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment[J]. Applied Soft Computing, 2012, 12(3): 1231-1237.

[30] Tang G, Tang C, Claramunt C, et al. Geometric A-star algorithm: An improved A-star algorithm for AGV path planning in a port environment[J]. IEEE access, 2021, 9: 59196-59210.

[31] Yang Y, Wang K. Efficient Logistics Path Optimization and Scheduling Using Deep Reinforcement Learning and Convolutional Neural Networks[J]. Informatica, 2025, 49(16).
[32] Tairan D, Yuhao W, Yang Z, et al. Optimal Scheduling Scheme for Ore Transshipment Yard Based on Probabilistic Calculation Model[C]//Proceedings of the 2024 4th International Symposium on Big Data and Artificial Intelligence. 2024: 154-159.
[33] Chen Y, Chen M, Yu F, et al. An improved ant colony algorithm with deep reinforcement learning for the robust multiobjective AGV routing problem in assembly workshops[J]. Applied Sciences, 2024, 14(16): 7135.
[34] Zhang Y, Wang L. a Dynamic Scheduling Method for Logistics Supply Chain Based on Adaptive Ant Colony Algorithm[J]. International Journal of Computational Intelligence Systems, 2024, 17(1): 198.
[35] Song Q, Zhao Q, Wang S, et al. Dynamic path planning for unmanned vehicles based on fuzzy logic and improved ant colony optimization[J]. IEEE Access, 2020, 8: 62107-62115.
[36] Chen X, Liu S, Zhao J, et al. Autonomous port management based AGV path planning and optimization via an ensemble reinforcement learning framework[J]. Ocean & Coastal Management, 2024, 251: 107087.
[37] Li K, Liu T, Kumar P N R, et al. A reinforcement learning-based hyper-heuristic for AGV task assignment and route planning in parts-to-picker warehouses[J]. Transportation research part E: logistics and transportation review, 2024, 185: 103518.
[38] Wang Y, Liang X. Application of Reinforcement Learning Methods Combining Graph Neural Networks and Self-Attention Mechanisms in Supply Chain Route Optimization[J]. Sensors, 2025, 25(3): 955.
[38] Li K. Optimizing warehouse logistics scheduling strategy using soft computing and advanced machine learning techniques[J]. Soft Computing, 2023, 27(23): 18077-18092.
[39] Thakre N K, Nimma D, Turukmane A V, et al. Dynamic Path Planning for Autonomous Robots in Forest Fire Scenarios Using Hybrid Deep Reinforcement Learning and Particle Swarm Optimization[J]. International Journal of Advanced Computer Science & Applications, 2024, 15(9).