Journal of Engineering, Project, and Production Management
2026, 16(2), 2025-196

Route Optimization for Unmanned Delivery Vehicles Using Improved Q-Learning and Lego-Loma

Haohao Yue
Associate Professor, College of Business Administration, Zibo Polytechnic University, Zibo, 255300, China, E-mail: haoyi121121@163.com

Project Management
Received September 4, 2025; revised October 22, 2025; accepted October 23, 2025
Available online March 7, 2026

Abstract: With the development of technology and economy, unmanned delivery vehicles are being applied across multiple industries. To more accurately design optimal delivery routes and reduce time and energy consumption, this paper builds a route optimization model by integrating an improved Q-learning algorithm with an enhanced, lightweight laser-based odometry and mapping algorithm. The innovation of this study lies in the integrating multiple algorithms for route optimization. By augmenting the Q-learning algorithm with a simulated annealing algorithm and an improved reward mechanism, the approach effectively avoids local optima while enhancing global search capabilities. This advancement overcomes key limitations of traditional reinforcement learning, enabling the improved algorithm to outperform deep Q-networks and random reward reinforcement learning methods demonstrating faster convergence, stronger learning capacity, and better adaptability. Furthermore, this study incorporates data from an inertial measurement unit and an extended Kalman filter to optimize the lightweight LiDAR-based ranging algorithm, significantly improving the system's stability and positioning accuracy. Through these integrations, the study establishes a high-performance route optimization model that avoids local optima, maintains enhanced stability, and exhibits superior environmental adaptability, making it particularly suitable for logistics enterprises and urban planning departments. This study improves the original algorithm by incorporating inertial measurement unit data for better mapping and pose estimation. Subsequently, Simulated Annealing (SA)and a modified reward mechanism are then applied to optimize Q-learning for path generation. The proposed model generates routes that are 23 m and 61 m shorter than those of the two comparison models, respectively. In addition, the Root Mean Square Error (RMSE) of the proposed model is 0.32, which is lower than those of the baseline models, demonstrating higher accuracy and better fit. These results indicate that the proposed model performs well in predicting optimal routes. It offers significant advantages in route optimization and can reliably analyze and predict complex road conditions, supporting the safe operation of unmanned delivery vehicles.

Keywords: Q-learning, LeGO-LOAM , simulated annealing, unmanned delivery vehicle.

DOI 10.32738/JEPPM-2025-196

1. Introduction

Driven by the remarkable pace of progress in artificial intelligence, unmanned delivery vehicles have been introduced into various fields and applied in industries such as transportation, food service, and healthcare (Orbay et al., 2025). Route planning plays a crucial role in the operation of unmanned delivery vehicles. One major research challenge is optimizing these routes so that the vehicles can reach their destinations safely and efficiently (Shafaei et al., 2025). Most existing methods rely on global planning algorithms, which often demonstrate weak obstacle avoidance and slow performance, making them less suitable for the fast and efficient operation required in real-world scenarios (Kong et al., 2024). Q-learning is a reinforcement learning technique that uses temporal difference learning to update value estimates. It does not rely on a predefined model and instead learns through direct interaction with the environment, which makes it highly adaptive (Verma et al., 2025). Lightweight and Ground-Optimized LiDAR Odometry and Mapping (LeGO-LOAM) algorithm is an efficient, laser-based localization and mapping method . It is recognized for being lightweight and accurate, estimating poses with minimal computational cost (Demertzis and Iliadis, 2023). To better plan and optimize the routes of unmanned delivery vehicles, this study constructs a route optimization model by combining an improved Q-learning algorithm with an enhanced LeGO-LOAM . The LeGO-LOAM is improved by integrating data from the Inertial Measurement Unit (IMU) and the Extended Kalman Filter (EKF) to enhance mapping and pose estimation. The improved Q-learning algorithm is applied for path generation. The innovation of this study lies in applying Simulated Annealing (SA)

to improve the ε-greedy strategy in Q-learning and combining a distance-based reward function to create an enhanced reward mechanism. These improvements lead to faster convergence and better optimization performance. This model is designed to generate and optimize routes more accurately, supporting unmanned delivery vehicles in delivering goods efficiently and with lower energy consumption.

2. Related Works

The improved Q-learning enables autonomous interaction with the environment to generate optimal paths, while LeGO-LOAM performs mapping and pose estimation based on LiDAR scan data. Both of these algorithms have been extensively studied. Wu et al. proposed a path search model for unmanned aerial vehicles based on adaptive transformation Q-learning to address the challenges of autonomous search and rescue tasks. They initialized the state-action value table using sensor data and applied a subdomain search algorithm with a compound reward function for path planning (Wu et al., 2023). Zhao et al. (2024) proposed a scale optimization framework leveraging Q-learning to address scheduling challenges within manufacturing workshops. They improved the greedy strategy of Q-learning using a variable neighborhood descent algorithm, sorted unscheduled jobs according to disturbance strategies, and applied a learning mechanism to identify solutions with lower objective values. Zamfirache et al. (2023) proposed a Q-learning-based control optimization model for servo systems. They initialized relevant parameters using a gravitational search algorithm and completed policy training through neural networks for control optimization. Yao et al. (2024) proposed a navigation model based on LiDAR odometry and LeGO-LOAM to reduce environmental interference on greenhouse robots. They combined map matching with an open planner to identify terrain and the distribution of plants, filtering out ground interference for autonomous navigation. Li et al. (2023) developed a shortest path generation model based on LeGO-LOAM for object detection. They integrated a quantum genetic algorithm and a dynamic window approach for path optimization and obstacle detection.

Path planning is essential for the stable operation of unmanned vehicles, and research in this field is ongoing. Tirkolaee et al. (2023) proposed a path optimization model based on particle swarm optimization to reduce factory transportation costs. They combined the gray wolf optimization algorithm with particle swarm optimization for numerical computation and conducted sensitivity analysis to identify the optimal solution. Yuan et al. (2024) addressed the trade-off between transportation cost and patient satisfaction in non-emergency medical transport by proposing a path planning model based on a hybrid heuristic algorithm. They designed pricing levels and calculated optimal charging routes to improve both user satisfaction and energy efficiency (Yuan et al., 2024). Tang et al. (2024) proposed a real-time route planning model to address the imbalance in urban traffic development across different areas. They analyzed path accessibility for various transportation modes and used the Theil index and Gini coefficient to further evaluate path performance based on accessibility results, aiming to explore the causes of traffic inequality (Tang et al., 2024). Cao (2025) proposed a path optimization model for unmanned aerial vehicle speed adjustment based on an improved adaptive genetic algorithm. The experimental findings demonstrated that the enhanced algorithm reduced processing time by 17.31 seconds compared to a conventional genetic algorithm. Saga et al. (2024) addressed the issue of obstacle interference in marine navigation by proposing an obstacle avoidance path optimization model based on deep reinforcement learning. They applied reinforcement learning to adaptively learn environmental conditions and avoid obstacles during navigation. Four different navigation scenarios were analyzed to generate optimal routes based on accumulated learning experience.

In summary, while existing path planning research has made notable progress, challenges remain in accurately mapping complex road conditions. To address navigation errors and obstacle avoidance difficulties commonly encountered during unmanned delivery vehicle operations, this study proposes an optimized route model combining an enhanced Q-learning algorithm with improved LeGO-LOAM algorithm. The proposed approach aims to improve path optimization efficiency and advance the development of autonomous delivery vehicles.

3. Route Optimization Model based on Improved Q-Learning and LeGO-LOAM Algorithms

3.1. Q-learning Optimization based on SA and Modified Reward Mechanism

Q-learning is an efficient and adaptive reinforcement learning algorithm. It interacts with the environment by autonomously selecting actions to receive rewards, builds a state-action value table, and updates e continuously based on these actions. After multiple iterations, it selects the optimal values from the Q-table for path planning (Zahedy et al., 2024). The construction of the state-action value table depends on the calculation of the action-value function, as shown in Equation (1).

$$Q(s,a)=\sum_{t=0}^{T-1}\gamma^{t}r_{t+1}(s_t,a_t,s_{t+1}) \tag{1}$$

In Equation (1), s represents the environmental state during interaction, a refers to the action taken by the unmanned delivery vehicle, r is the reward obtained after the vehicle takes action a, and γ is the discount factor that reflects future reward changes. During interaction, the vehicle updates the state-action value table by adjusting actions to obtain Q values until the optimal value is selected. The Q value during the updating process is calculated as shown in Equation (2).

$$Q(s,a)\leftarrow Q(s,a)+\alpha(r+\gamma*\max_{a'}Q(s',a')-Q(s,a)) \tag{2}$$

In Equation (2), $\max_{a'}Q(s',a')$ represents the optimal value under state $t+1$, and α is the learning rate. By iteratively updating the Q-table, the unmanned delivery vehicle learns the optimal action for every state until the optimal path emerges. The Q-learning path formation procedure is depicted in Figure 1.

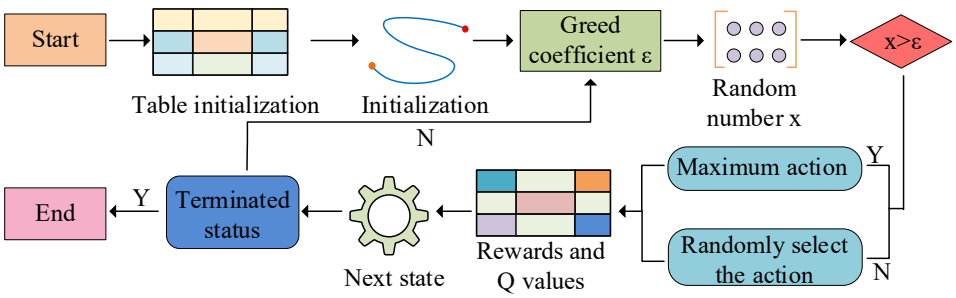


Fig. 1. Optimal path generation process of Q-learning

As shown in Figure 1, the Q-learning process searches for optimal actions by iteratively updating the value table. First, it initializes parameters, including the values of Q, the starting point, and the endpoint. Then, it uses the ε-greedy method to obtain the greedy parameter ε, and generates a random number between 0 and 1 for comparison. If the random number exceeds , it selects the action with the highest value as the optimal action; otherwise, it randomly selects an action. Actions are executed iteratively to obtain rewards and update the state-action value table. After multiple updates, the algorithm checks whether the optimal state has been reached. If so, the iteration stops. In this process, the parameter calculation of the ε-greedy method is critical, as shown in Equation (3).

$$\pi(a|s)=\begin{cases}1-\varepsilon+\dfrac{\varepsilon}{m}, a=a*\\ \dfrac{\varepsilon}{m}, a\neq a*\end{cases} \tag{3}$$

In Equation (3), $\pi(a|s)$ is the probability of selecting action a under state s. m is the number of available actions under state s, and $a*$ is the optimal action selected. The ε-greedy method helps the algorithm identify optimal values, but it often converges to local optima (Wu et al., 2023). SA is an optimization algorithm based on the solid-state physical annealing principle. It helps algorithms escape local optima by accepting suboptimal solutions near local optima with a certain probability (Li et al., 2025). This probability is expressed in Equation (4).

$$p(1\to 2)=\begin{cases}1, E_2<E_1\\ e-\dfrac{E_2-E_1}{T}, E_2>E_1\end{cases} \tag{4}$$

In Equation (4), E_1 and E_2 represent the system energy under two different states. To speed up the algorithm and avoid early convergence to local optima, this study uses SA to optimize the ε-greedy method. The optimized ε-greedy method is expressed in Equation (5).

$$\varepsilon_k=\varepsilon_f+(\varepsilon_1-\varepsilon_f)(\mu_1+e^{-\mu 2(k-N)}) \tag{5}$$

In Equation (5), N is the total number of training rounds, ε_k is the exploration rate at round k, ε_1 is the initial exploration rate, ε_f is the final exploration rate, and μ_1 and μ_2 are scaling factors. The process of finding the optimal solution using the improved ε-greedy method is shown in Figure 2.

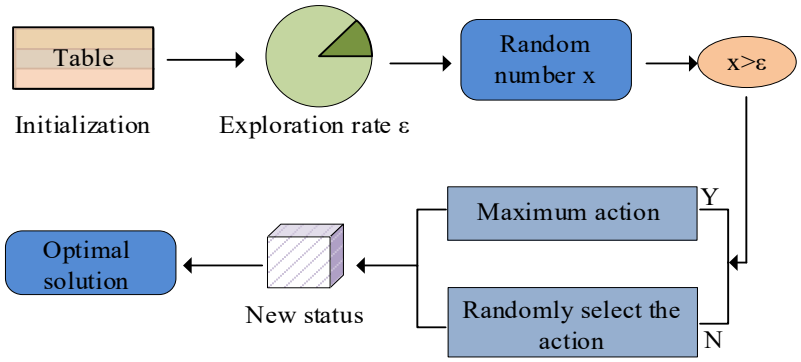


Fig. 2. Flowchart of improved ε- greedy method

In Figure 2, the ε-greedy method uses SA to adjust the exploration rate for the current state. This rate is compared against a random generated number ranging from 0 to 1. When the random number exceeds the current exploration rate, the algorithm selects the action with the highest Q- value; otherwise, it randomly selects an action. The algorithm then enters a new state, updates the Q- value and the Q- table, and gradually converges toward the optimal solution. In this process, the reward function is key to accelerating convergence. To improve exploration efficiency and reduce sensitivity to environmental interference, this study designs a distance reward function and a collision reward function based on heuristic search. After improvement, the reward obtained by the unmanned delivery vehicle after performing action a in state s is the sum of the collision reward value and the distance reward value. To better reflect real-world road conditions, the task distance is set as the parameter for the collision reward, and the collision reward function is given in Equation (6).

$$reward_1 = \begin{cases} -1, no\ collision\ occurred \\ \lambda\sqrt{(x_{s_t} - X)(y_{st} - Y)}, reach\ the\ finish\ line \\ -\beta, collision\ occurred \end{cases} \tag{6}$$

In Equation (6), λ is a target parameter in the range [3, 5], β is a collision coefficient in the range [20, 50], X is the x-coordinate, Y is the y-coordinate, s_t is the current interaction state, and x_{s_t} and y_{st} are the x- and y-coordinates of state s_t. To handle long-distance tasks, the reward function is improved using a distance reward function, as shown in Equation (7).

$$reward_2 = \varphi \frac{(\sqrt{(x_{st} - X)^2 + (y_{st} - Y)^2} - \sqrt{(x_{s_{t+1}} - X)^2 + (y_{s_{t+1}} - Y)^2})}{D} \tag{7}$$

In Equation (7), φ is the reward coefficient and D is the Euclidean distance between the starting and target locations. The improved ε-greedy method and reward function are integrated into the Q-learning structure. The improved Q-learning process is shown in Figure 3.

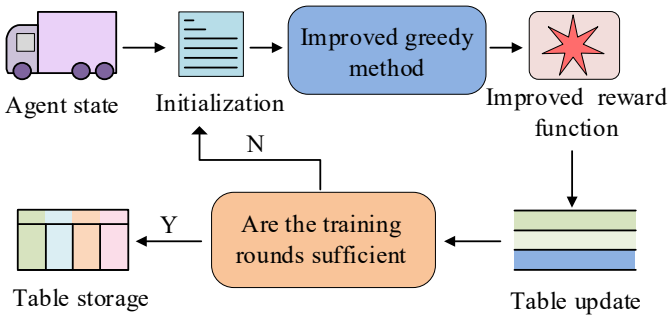


Fig. 3. Flowchart of improved Q- Learning

As shown in Figure 3, the enhanced Q-learning algorithm workflow begins with initializing table data and parameters. By implementing a simulated annealing algorithm to refine the ε-greedy method, it prevents getting trapped in local optima. The optimized ε-greedy approach then conducts memory-based global search to identify the optimal value, which guides the execution of optimal actions. The reward function is enhanced through distance-based and collision-based reward mechanisms, enabling calculation of action benefits and subsequent state-action value table updates. This iterative process continues until the specified training rounds are completed, ultimately storing the resulting table and generating an optimal path strategy.

3.2. Construction of Route Optimization Model

The improved Q-learning enhances convergence speed and optimization performance. However, it still cannot accurately extract environmental location information. LeGO-LOAM estimates ground pose information with high accuracy and completes mapping (Li et al., 2024). Therefore, this study combines improved Q-learning with LeGO-LOAM to construct a route optimization model for designing and optimizing unmanned delivery vehicle paths. Feature extraction is an important stage in LeGO-LOAM . The smoothness calculation at this stage is shown in Equation (8).

$$s = \frac{1}{|S|\|r_i\|}\left\| \sum_{j \in S, j \neq i} (r_j - r_i) \right\| \tag{8}$$

In Equation (8), S represents the set of neighborhood points, r_i is the depth value of the point being calculated, and r_j refers to the depth value of points with known curvature in set S. After feature extraction, LeGO-LOAM uses LiDAR odometry for mapping and pose estimation. The pose estimation process of LeGO-LOAM is shown in Figure 4.

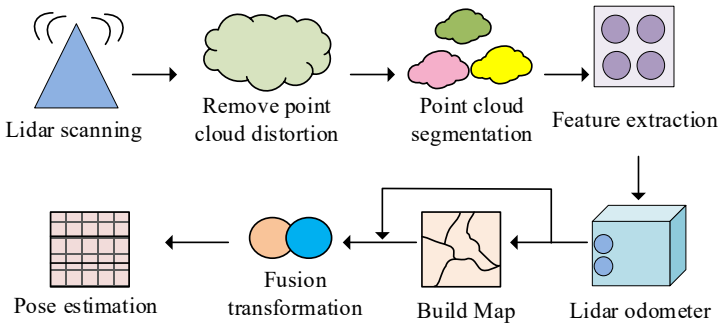


Fig. 4. Pose estimation process of LeGO-LOAM

As shown in Figure 4, the process first removes point cloud distortion from LiDAR scan data and performs projection transformation. Then, the projected image is divided into different point cloud clusters. Ground points are extracted based on their categories. LiDAR odometry is used for point cloud registration and pose calculation, at this point, mapping and pose estimation are completed. Point cloud matching is the core step of LeGO-LOAM . The extracted point sets are matched point-to-point and point-to-plane to calculate six degrees of freedom for pose estimation. The six degrees of freedom are defined in Equation (9).

$$[t_x, t_y, t_z, \theta_{roll}, \theta_{pitch}, \theta_{yaw}] \tag{9}$$

In Equation (9), t_x , t_y , and t_z are translations along the three axes of the global coordinate system, while θ_{roll} , θ_{pitch} , and θ_{yaw} represent rotations around the three axes. The nonlinear least squares function is used for distance transformation adjustment. Its expression is shown in Equation (10).

$$f(\tilde{P}_{t_i}^{af}) = f(T_{t_i}^{l}) = d \tag{10}$$

In Equation (10), $\tilde{P}_{t_i}^{af}$ is the reprojected point set at moment t_i , d is the distance matrix from all points in set $P_{t_i}^{af}$ to the plane, $f(.)$ represents the nonlinear processing including reprojection, and T_{ti}^{l} is the pose value of the point at moment t_i . The point cloud matching stage in LeGO-LOAM relies on the localization system and is vulnerable to environmental interference, which can lead to localization failure (Yan et al., 2023). This study improves the LeGO-LOAM structure by inputting both IMU data and lidar data into EKF to obtain more accurate pose information. The structure of LeGO-LOAM is given by Figure 5.

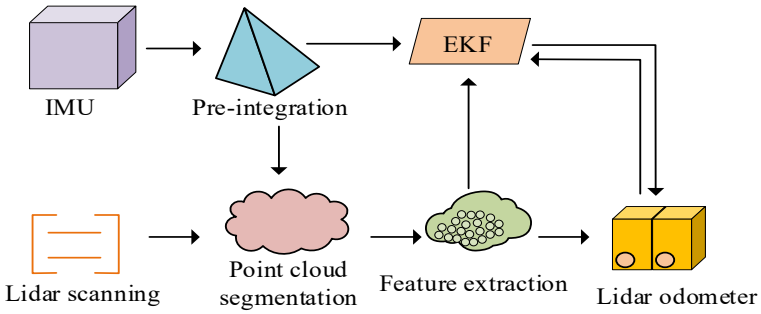


Fig. 5. Structure of LeGO-LOAM combined with IMU

As shown in Figure 5, IMU pre-integrated data is used first to remove point cloud distortion in the lidar scan data. Then, the ground point cloud clusters are segmented, and different types of ground feature points are extracted. EKF is then used to update data during the lidar odometry phase and optimize the pose estimation. The improved LeGO-LOAM uses IMU data to calculate the speed, position, and pose of the unmanned delivery vehicle. Assuming constant steering angle and acceleration in the IMU coordinate system, the speed at moment $t + \Delta t$ is calculated as shown in Equation (11).

$$v_{t+\Delta t} = v_t + g\Delta t + R_t(\tilde{a}_t - b_t^a - n_t^a)\Delta t \tag{11}$$

In Equation (11), g is the fixed gravity vector in the LeGO-LOAM world coordinate system, R represents the rotation matrix that transforms coordinates from the IMU frame to the world frame, n is white noise, $\tilde{a}_t$ is acceleration, and b is

the bias offset. The position of the unmanned delivery vehicle at moment $t+\Delta t$ is calculated using Equation (14).

$$p_{t+\Delta t} = p_t + v_t \Delta t + \frac{1}{2} g \Delta t + \frac{1}{2} R_t (a_t - b_t^a - n_t^a) \Delta t^2 \tag{12}$$

In Equation (12), p_t is the position information at time t. According to the calculation results of speed and position, the attitude information of the unmanned delivery vehicle in period $t+\Delta t$ is obtained. LeGO-LOAM improved with IMU data shows enhanced localization accuracy but can only provide pose information and cannot independently complete path planning. Therefore, this study constructs a route optimization model combining improved Q-learning and enhanced LeGO-LOAM to complete path planning and optimization for unmanned delivery vehicles. The full process is shown in Figure 6.

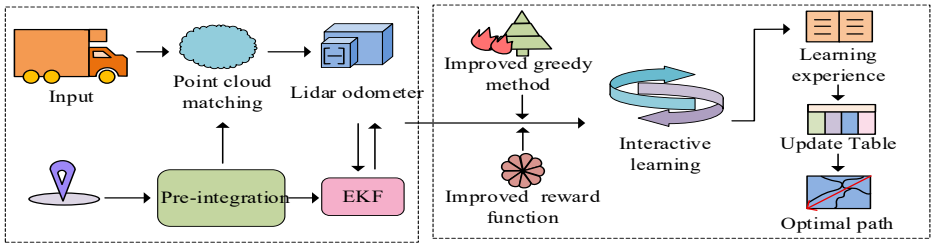


Fig. 6. Route optimization process for unmanned delivery vehicles

As shown in Figure 6, the proposed model begins by removing point cloud distortion using IMU data and lidar scan data. Then, EKF and lidar odometry are used for prediction and update, completing mapping and pose estimation. The model then uses the SA-improved reward mechanism to calculate the optimal value and interact with the environment. The optimal action is executed repeatedly. Finally, the learning experience obtained from interaction is used to update the state-action value table, and the optimal path is generated.

4. Performance Verification of the Unmanned Delivery Vehicle Route Optimization Model

4.1. Performance of Q- Learning based on SA and Modified Reward Mechanism

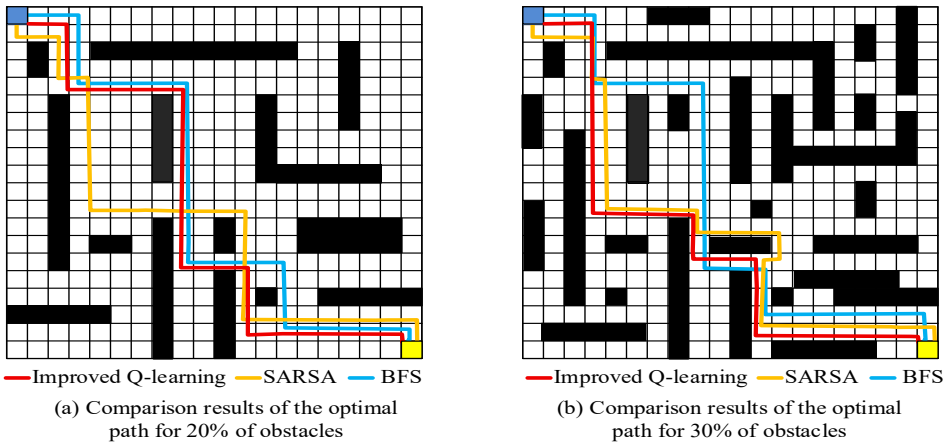
To evaluate the path optimization performance of the improved Q-learning, it was compared with the State-Action-Reward-State-Action (SARSA) algorithm and Breadth-First Search (BFS). A grid-based method was used to simulate the environment, and obstacles were manually set for simulation experiments. A 20 m × 20 m grid map was created, in which different colors represented the start point, end point, and obstacles. Black grids indicated obstacle cells, white ones represented normal cells, yellow grids marked the end point, and blue grids marked the start point. Python was used as the programming language for the experiments, and Adam was selected as the optimizer. The experiment used Windows 10 as the operating system, Python as the programming language, and PyTorch as the deep- learning framework. To maintain the Model's optimal state, it was trained multiple times until the best hyperparameter combination was obtained. The specific hyperparameter settings of the experiment is shown in Table 1.

Table 1. Experimental parameter settings

Parameter	Symbol	Value
Learning rate	α	0.1
Exploration rate	ε_k	0.4
Initial exploration rate	ε_1	0.4
Eventually explore rate	ε_f	0.001
Training Rounds	N	5000
Immediate target reward	r_1	100
Collision reward value	$reward_1$	-100
Proportional factor	μ_1	-1
Proportional factor	μ_2	0.0001

According to the above parameter settings, the model was constructed and experiments were carried out. Two grid maps with obstacle densities of 20% and 30% were designed. The optimal path results of the improved Q-learning, SARSA, and BFS were compared, as shown in Figure 7.

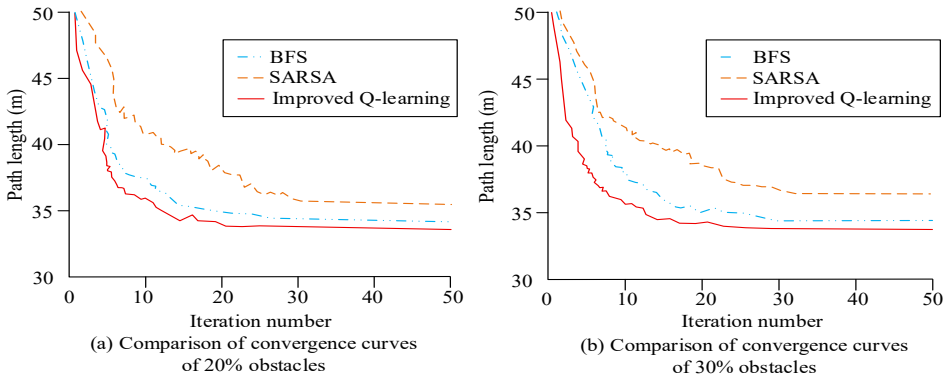
As shown in Figure 7(a), on the grid map with 20% obstacle density, only the BFS and the improved Q-learning successfully generated an optimal path. The path length generated by the improved Q-learning was 33.86m, shorter than the 34.34 m generated by the BFS and the 36.78 m generated by the SARSA, indicating superior optimization performance. As shown in Figure 7(b), on the grid map with 30% obstacle density, the path length generated by the improved Q-learning was 34.12 m, which remained shorter than those generated by the comparison algorithms. Overall, the improved Q-learning showed stronger adaptability to complex maps. The convergence curves of the three algorithms over 50 iterations on the two grid maps were calculated and compared. The results are shown in Figure 8.



(a) Comparison results of the optimal path for 20% of obstacles

(b) Comparison results of the optimal path for 30% of obstacles

Fig. 7. Optimal path comparison on grid maps with 20% and 30% obstacle density



(a) Comparison of convergence curves of 20% obstacles

(b) Comparison of convergence curves of 30% obstacles

Fig. 8. Path length convergence curves of different algorithms

As shown in Figure 8(a), on the grid map with 20% obstacle density, the improved Q-learning reached a stable state after 20 iterations, while the SARSA and BFS reached convergence at 25 and 29 iterations respectively. The optimal path length of the improved Q-learning at convergence was 33.76m, shorter than those of the other algorithms. As shown in Figure 8(b), on the grid map with 30% obstacle density, the improved Q-learning achieved the shortest path length of 34.06 m and reached convergence after 23 iterations, demonstrating faster convergence speed. To further compare the convergence performance, the convergence time and prediction accuracy of each algorithm were calculated. The results are shown in Table 2.

Table 2. Comparison of convergence performance under different obstacle densities

Obstacle ratio	Algorithm	Accuracy	Iteration	Time-consuming (s)
20%	BFS	0.83	29	3.85
	SARSA	0.86	25	2.96
	Improved Q-learning	0.91	20	2.53
30%	BFS	0.81	33	4.06
	SARSA	0.85	29	3.21
	Improved Q-learning	0.89	25	2.81

As shown in Table 2, on the grid map with 20% obstacle density, the improved Q-learning achieved an accuracy of 0.91, which was significantly better than the comparison algorithms, indicating strong path optimization performance. On the map with 30% obstacle density, the improved Q-learning reached an accuracy of 0.89 and a convergence time of 2.81 s, both better than the other algorithms, reflecting stronger stability and faster convergence speed. In summary, the improved Q-learning successfully identified optimal paths on maps with different obstacle densities and demonstrated better stability and convergence performance, which effectively supported path planning and optimization.

4.2. Analysis of Route Optimization Combining Improved Q- Learning and Enhanced LeGO-LOAM

The optimal paths were generated and compared with ground truth. The results are shown in Figure 9. Building upon the

validation of Q-learning performance improvements, this study further evaluates model effectiveness through practical application experiments. Comparative analyses with the A* algorithm and Floyd's algorithm were conducted to verify the model's real-world performance. The experiment was conducted on a long urban street using a Velodyne VLP-16 LiDAR sensor-equipped autonomous vehicle, powered by a Lenovo laptop and a Wheeltec N100 IMU. A total of 4,000 training episodes were set for the model. The unmanned delivery vehicle traveled two full laps at 1 m/s to collect environmental data, identify spatial features, and perform loop detection for map updates. The experimental environment matched the validation setup with target point reward values of 100 and collision penalty values of -100. Optimal path generation based on road conditions was compared with ground truth results, as illustrated in Figure 9.

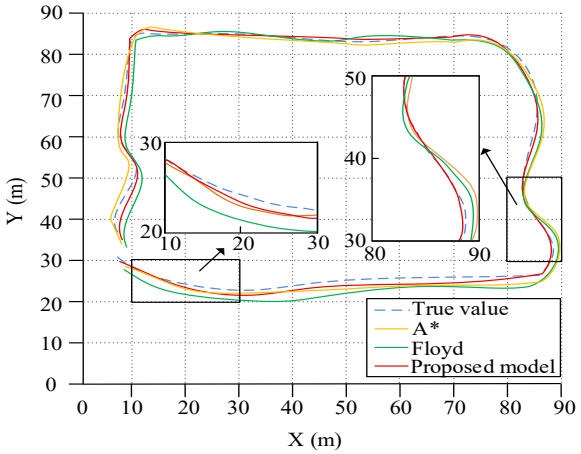
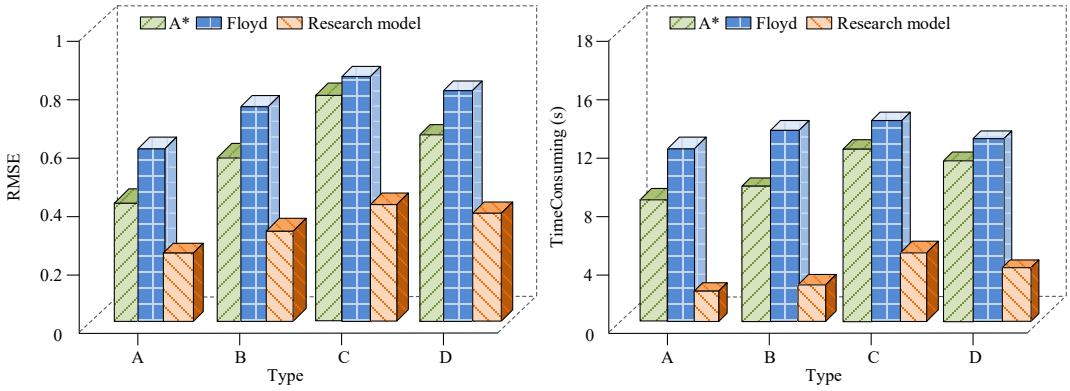


Fig. 9. Trajectory comparison of optimal paths for different models

As shown in Figure 9, the model proposed in this study generated XY-axis path trajectories that were 23 meters shorter than the A* algorithm and 61 meters shorter than the Floyd algorithm. This demonstrates that the proposed model can generate optimal shortest paths, enabling delivery vehicles to reach destinations faster and complete more tasks per unit time, thereby enhancing delivery efficiency. Overall, the trajectories generated by this model show higher consistency with real-world data. In two local zoom-in views, the alignment performance of this model with optimal paths outperforms other models, indicating superior fitting capabilities. The study introduced Gaussian noise (A) and pulse noise (B) with varying intensities in sensors, and simulated dynamic traffic environments through randomly placed pedestrians (C) and vehicles (D). To verify the model's robustness, comparative analysis of pose errors and computational time across four environments for all three models was conducted, with results presented in Figure 10.



(a) Error comparison of each model in different environments

(b) The computational time consumption of each model in different environments

Fig. 10. The comparison diagram of attitude error and calculation time of each model in different environments

As shown in Figure 10(a), the proposed model demonstrated RMSE values of 0.28 and 0.37 in environments A and B, respectively, both lower than those of the comparison models. This indicates the model's enhanced adaptability to noisy environments and its strong anti-interference capability. In environments C and D, the RMSE values were 0.46 and 0.43, respectively, also outperforming the comparison models and demonstrating superior adaptability to dynamic traffic changes, further validating the model's robustness. This improvement is attributed to the adoption of the SA algorithm and other optimization techniques that enhance model stability. Figure 10 (b) reveals faster computation times: 3.12s and 3.71s for environments A/B versus 5.91s and 4.81s for environments C/D, significantly outperforming the comparison model. Overall, the model effectively adapts to dynamic traffic conditions and diverse noise environments while maintaining computational efficiency, proving its practicality and scalability for path planning in large fleets or complex urban road networks. To further verify the convergence performance of each model, the study set up roadblocks on the route under

two different scenarios: temporary road conditions (Road Condition 1) and construction road conditions (Road Condition 2), which varied in roadblock complexity. The learning steps of each model under these two road conditions were then calculated and compared. The calculation results are shown in Figure 11.

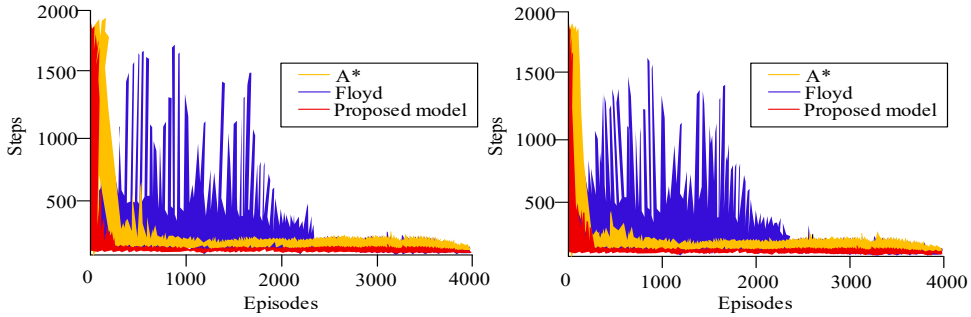


Fig. 11. Step convergence results under different route conditions

As shown in Figure 11(a), the proposed model demonstrated faster step convergence speed under Road Condition 1 compared to the comparison model. After obstacle avoidance, it reached a stable state approximately after 250 learning episodes, with the step size remaining below 500 and minimal fluctuation after 100 learning episodes, indicating superior stability and faster convergence. Figure 11(b) reveals that under Road Condition 2, the proposed model achieved convergence after 300 learning episodes, still outperforming the comparison model. Notably, the step size stayed below 500 even after 50 learning episodes, further demonstrating enhanced stability and scalability. In summary, the proposed model outperforms the comparison model in accuracy, convergence speed, stability, and computational efficiency. It enables efficient urban logistics delivery with lower computational costs and strong robustness, effectively assisting robots in adapting to city-wide distribution tasks.

5. Conclusion

To address common navigation errors and obstacle avoidance challenges in autonomous delivery vehicles, this study integrated an enhanced Q-learning algorithm with an improved LeGO-LOAM algorithm to develop a path optimization model. The proposed method improved the reward mechanism of the Q-learning algorithm by incorporating SA to enhance optimization efficiency, and integrated IMU data into the LeGO-LOAM algorithm to boost pose estimation accuracy. To validate the path optimization effectiveness, experimental setups were established to analyze the performance of both the enhanced Q-learning algorithm and the optimized model. Experimental results demonstrated that the improved Q-learning algorithm successfully found optimal paths in 20x20 grid maps with 20% and 30% obstacle occupancy rates, achieving convergence within 20 and 25 iterations respectively, and exhibiting high stability and fast convergence. The average path lengths of the improved Q-learning algorithm across both map types were 33.92 m and 34.16 m, surpassing those of the benchmark algorithm. The model exhibited RMSE values of 0.28 and 0.37 in noise-enriched environments, which were significantly lower than the benchmark model, indicating better noise adaptation and stronger anti-interference capabilities. Additionally, the RMSE values under dynamic traffic change scenarios were 0.46 and 0.43 for the two environments, both lower than those of the benchmark model, demonstrating enhanced adaptability to traffic dynamics and further validating the model's robustness. Under both temporary roadblocks and construction site conditions established in the experiment, the proposed model demonstrated convergence after 250 and 300 training sessions respectively, requiring significantly fewer iterations compared to the benchmark model. This indicates robustness and superior resistance to obstacle interference. The experimental results showed that the model outperformed the comparison model in accuracy, convergence speed, stability, and computational efficiency, thus enabling efficient urban logistics delivery while maintaining strong adaptability. Although the model demonstrated high stability, precision, and an ability to effectively handle dynamic traffic changes, discrepancies remained between the model's predictions and the actual optimal path values. Additionally, real-world factors such as communication delays in fleet coordination were not addressed. Future improvements will focus on these aspects to enhance practical applicability.

Author Contributions

Haohao Yue contributed to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision, project administration, and funding acquisition.

Institutional Review Board Statement

not applicable.

Funding

This research received no specific financial support from any funding agency.

Declaration of Artificial Intelligence (AI) Tools

The author used AI tools solely for language editing and readability improvement. The author reviewed and verified all

content and takes full responsibility for the accuracy and integrity of the manuscript.

Reference

Cao, Z. (2025). Simulation investigation of autonomous route planning for unmanned aerial vehicles based on an improved genetic algorithm. *Neural Computing and Applications*, 37(5), 3343-3354. doi: 10.1007/s00521-024-10817-8

Demertzis, K., and Iliadis, L. (2023). Next generation automated reservoir computing for cyber defense. In *IFIP International Conference on Artificial Intelligence Applications and Innovations* (pp. 16-27). Springer Nature Switzerland, Cham. doi: 10.1007/978-3-031-34107-6_2

Kong, M., Wang, W., and Deveci, M. (2024). A novel carbon reduction engineering method-based deep Q-learning algorithm for energy-efficient scheduling on a single batch-processing machine in semiconductor manufacturing. *International Journal of Production Research*, 62(18), 6449-6472. doi: 10.1080/00207543.2023.2252932

Li, B., Huang, X., Cai, J., and Ma, F. (2025). LB-LIOSAM: an improved mapping and localization method with loop detection. *Industrial Robot: the international journal of robotics research and application*, 52(3), 381-390. doi: 10.1108/IR-07-2024-0314

Li, G., Cai, C., Chen, Y., and Chi, Y. (2024). Is Q-learning minimax optimal? a tight sample complexity analysis. *Operations Research*, 72(1), 222-236. doi: 10.1287/opre.2023.2450

Li, S. A., Chen, Y. C., Wu, B. X., and Feng, H. (2023). 3D lidar SLAM-based systems in object detection and navigation applications. *Journal of the Chinese Institute of Engineers*, 46(8), 912-925. doi: 10.1080/02533839.2023.2261983

Orbay, K., Sagbas, M., and Demir, M. (2025). Design of Cardiac Pacemaker Controller Based on Reinforcement Learning. *Artificial Intelligence Theory and Applications*, 5(1), 29-41.

Saga, R., Kozono, R., and Nihei, T. Y. (2024). Deep-reinforcement learning-based route planning with obstacle avoidance for autonomous vessels. *Artificial Life and Robotics*, 29(1), 136-144. doi: 10.1007/s10015-023-00909-4

Shafaei, A., Jokar, M. R. A., Rafiee, M., and Hemmati, A. (2025). Using the route planning for supplying spare parts to reduce distribution costs: a case study in a roadside assistance company. *International Journal of Shipping and Transport Logistics*, 20(1), 131-158. doi: 10.1504/IJSTL.2025.144995

Tang, H., Li, Y., Li, M., and Zhao, S. (2024). Accessibility and Equity Analysis of Highway-Railway Traffic Network Based on Real-Time Route Planning Data: A Case Study of Shandong Peninsula Urban Agglomeration, China. *Transportation Research Record*, 2678(11), 16-28. doi: 10.1177/03611981241239963

Tirkolaee, E. B., Goli, A., and Mardani, A. (2023). A novel two-echelon hierarchical location-allocation-routing optimization for green energy-efficient logistics systems. *Annals of Operations Research*, 324(1), 795-823. doi: 10.1007/s10479-021-04363-y

Verma, A. K., Singh, V. K., Khan, M. R. and Sethy, P. K.,(2025). Q-learning based heterogeneous network selection decision algorithm for ultra reliable and low latency communication services. *Wireless Networks*, 31(4), 3095-3110. doi: 10.1007/s11276-025-03931-5

Wu, J., Huang, S., Yang, Y., and Zhang, B. (2023). Evaluation of 3D LiDAR SLAM algorithms based on the KITTI dataset. *The Journal of Supercomputing*, 79(14), 15760-15772. doi: 10.1007/s11227-023-05267-3

Wu, J., Sun, Y., Li, D., Shi, J., Li, X., and Gao, L. (2023). An adaptive conversion speed Q-learning algorithm for search and rescue UAV path planning in unknown environments. *IEEE Transactions on Vehicular Technology*, 72(12), 15391-15404. doi: 10.1109/TVT.2023.3297837

Yan, Y., Li, G., Chen, Y., and Fan, J. (2023). The efficacy of pessimism in asynchronous Q-learning. *IEEE Transactions on Information Theory*, 69(11), 7185-7219. doi: 10.1109/TIT.2023.3299840

Yao, X., Bai, Y., Zhang, B., Xu, D., Cao, G., and Bian, Y. (2024). Autonomous navigation and adaptive path planning in dynamic greenhouse environments utilizing improved LeGO-LOAM and OpenPlanner algorithms. *Journal of Field Robotics*, 41(7), 2427-2440. doi: 10.1002/rob.22315

Yuan, P., Luo, M., Miao, G., and Li, J. (2024). Problem of Patient Transport Route Planning for Battery Electric Vehicles Considering a Flexible Charging Strategy. *Transportation Research Record*, 2678(8), 198-225. doi: 10.1177/03611981231214523

Zahedy, N., Barekatain, B., and Quintana, A. A. (2024). RI-RPL: a new high-quality RPL-based routing protocol using Q-learning algorithm. *The Journal of Supercomputing*, 80(6), 7691-7749. doi: 10.1007/s11227-023-05724-z

Zamfirache, I. A., Precup, R. E., and Petriu, E. M. (2023). Q-learning, policy iteration and actor-critic reinforcement learning combined with metaheuristic algorithms in servo system control. *Facta Universitatis, Series: Mechanical Engineering*, 21(4), 615-630. doi: 10.22190/FUME231011044Z

Zhao, F., Zhuang, C., Wang, L., and Dong, C. (2024). An iterative greedy algorithm with Q-learning mechanism for the multiobjective distributed no-idle permutation flowshop scheduling. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 54(5), 3207-3219. doi: 10.1109/TSMC.2024.3358383

Haohao Yue obtained her master's degree in International Trade from Gachon University, South Korea, in 2013. She is working as an associate professor in the College of Business Administration, Zibo Polytechnic University, China. Her research interests include e-commerce and new media marketing.