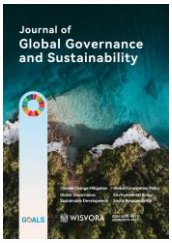
Available online at Wisvora

Journal of Global Governance and Sustainability

Journal homepage: https://wisvora.com/index.php/jggs/index



Transit Time-Based Graph Convolutional Network Integrated with a Q-Learning Approach for Solving the Vehicle Routing Problem to Minimize Carbon Dioxide Emissions in the Transportation of Building Components

Shih-Yang Lin[1*], Taibing Xue[2]

[1] Guangzhou Institute of Science and Technology, No. 638, Taihe Xingtai 3rd Road, Baiyun District, Guangzhou City, 510540, China.
[2] No. 1299 Jinguang Street, Shouguang City, Shandong Province, 262700, China.

ABSTRACT

During the construction process, the timely transportation and efficient allocation of prefabricated components are critical factors influencing project progress, construction quality, and cost management. This study aims to optimize the logistics transit of components in building construction by employing reinforcement learning algorithms, with the objective of improving transportation efficiency and minimizing transportation costs. We address the existing limitations of reinforcement learning in solving Vehicle Routing Problems (VRP), specifically suboptimal solution quality and limited generalization capabilities. To overcome these challenges, we propose an optimization approach based on transit time, which integrates Graph Convolutional Networks (GCN) with Q-learning and incorporates a road traffic congestion index within the three-dimensional coordinate solution space. By combining graph neural networks with reinforcement learning, the method enhances the decision-making effectiveness and adaptability of the reinforcement learning model within complex graph environments, enhances the convergence efficiency and solution quality of Q-learning during path search, and ultimately achieves routes with lower transportation costs. The integration of GCN with Q-learning facilitates a more rapid convergence during the learning phase. This improvement is attributed to the generation of more rational state vectors and more precise Q-value updates, enabling the agent to more effectively differentiate and select optimal actions, thereby enhancing the global optimization capability of path selection. Notably, under conditions of road congestion, the proposed Transit Time-based GCN integrated Q-learning approach significantly outperforms the original GCN+Q-learning approach, achieving reductions in fuel consumption and gas emissions by 21.7%, 22.06%, and 12.78% in scenarios involving 11, 21, and 31 stops, respectively.

KEYWORDS

Vehicle Routing Problem, Graph Convolutional Network, Q-learning, Carbon Dioxide Emissions.

* *Corresponding author: Shih-Yang, Li(shihyang.lin@hotmail.com)*

Received December 15 2025; Received in revised form January 4 2026; Accepted January 4 2026
Available online January 8 2026

1. INTRODUCTION

DURING the construction process, the timely transportation and rational allocation of prefabricated components are critical factors influencing project progress, construction quality, and cost management. Prefabricated building components are often characterized by large volume, heavy weight, and fragility. Their transportation is frequently constrained by factors such as transport routes, vehicle load capacity, delivery time requirements, and on-site storage availability. Inadequate scheduling and planning can lead to increased transportation costs, component backlogs, and construction delays. The advancement of intelligent construction technologies has heightened the building industry's demand for efficient and precise logistics management, rendering intelligent logistics scheduling a vital approach to optimize construction supply chain management [1]. Current logistics transit often depend on manual experience or static algorithm-based route planning, which are insufficiently responsive to the complex and dynamic demands of construction projects, resulting in low efficiency [2][3].

Traditional logistics scheduling methods exhibit several limitations in route optimization, demand matching, and resource utilization. Inadequate optimization of scheduling plans frequently results in uneven vehicle utilization, with some vehicles operating under capacity or remaining idle, thereby leading to inefficient use of transportation resources and elevated logistics costs. The absence of real-time dynamic optimization capabilities in conventional logistics scheduling further impedes the ability to make precise scheduling decisions in response to adjustments in construction progress. These challenges can be effectively addressed through reinforcement learning techniques, which enable continuous optimization via simulated environments and trial-and-error learning, thereby facilitating scheduling optimization under dynamic and uncertain conditions. Enhancements in the accuracy and efficiency of component delivery are also achievable through this approach. The application of reinforcement learning in intelligent construction logistics optimization research offers a promising solution to the shortcomings of traditional scheduling methods, providing efficient and cost-effective strategies that support the intelligent advancement of the construction industry [4][5].

Significant progress has been made internationally in the research and application of intelligent construction logistics management. Germany, as a pioneer, has planned and built multiple logistics centers across the nation. For instance, the Bremen logistics center was developed with careful consideration of local geographic and economic factors, particularly in site selection [6]. The adoption of advanced technologies and management approaches has led to improved operational efficiency and reduced costs within logistics operations. Notable advancements have been achieved in logistics planning and design abroad, with integrated simulation technologies being extensively employed. These technologies enable feasibility analyses of logistics centers, 3D virtual modeling to simulate actual operations, input-output analyses, in-warehouse process evaluations, and logistics transmission simulations, all of which contribute to optimization efforts. Supported by such technological innovations, the design schemes of logistics centers can be effectively aligned with real-world operational requirements [7].

In developed Western countries, logistics costs typically constitute approximately 10% of Gross Domestic Product (GDP), whereas in China, this proportion approaches 20%, significantly exceeding the international average [8]. This significant difference indicates that China’s logistics system still has considerable potential for improvement in terms of efficiency, technological advancement, and resource allocation. Advances in the logistics sector has emerged as a critical driver of national GDP growth. The implementation of scientifically grounded and appropriate logistics fulfillment plans can effectively reduce transportation costs. Vehicle route planning, a central component of the logistics operation process, is directly associated with optimizing delivery efficiency and minimizing operating costs. Many logistics enterprises face the dual challenge of ensuring rapid delivery while controlling transportation costs. Consequently, optimizing routes to enhance timeliness and reduce resource wastage has become a pivotal concern within the industry. The development of efficient route planning strategies can not only decrease transportation costs—estimated to range from 5% to 20% [9]—but also facilitate logistics companies in achieving efficient and sustainable development goals.

This paper mainly focuses on combinatorial optimization of the Vehicle Routing Problem (VRP) [10], with a focus on developing efficient solution methods for the transportation of construction materials. The VRP is an NP-hard optimization problem with integer constraints, frequently encountered in key applications across modern industries such as transportation, supply chain management, and warehouse scheduling. Consequently, it constitutes a central research topic within the domains

of intelligent logistics and operations optimization. As research on the VRP advances, numerous variants tailored to diverse application scenarios have been introduced, maintaining the field highly active. The main approaches to solving vehicle routing problems encompass traditional exact algorithms, heuristic approaches, and the increasingly prominent Reinforcement Learning (RL) algorithms.

Although exact and heuristic algorithms have yielded significant advancements in route optimization, they typically depend on researchers' profound understanding of the problem and extensive expert knowledge to develop appropriate models and design effective search strategies. These approaches frequently encounter a trade-off between search efficiency and solution quality when addressing large-scale, complex combinatorial optimization problems: achieving higher-quality solutions necessitates expanding the search space and increasing computational time, thereby constraining the practical applicability of these algorithms.

Reinforcement learning algorithms have emerged as a significant focus of research in the domain of path planning due to their effectiveness in decision optimization and autonomous learning. Integrating existing neural network architectures to further enhance the solution efficiency and generalization capacity of reinforcement learning algorithms for various real-world variants of the VRP constitutes a critical topic warranting comprehensive investigation. Such advancements can facilitate the design of transportation schemes that reduce logistics costs and improve the practical effectiveness of resource allocation. This is particularly pertinent in large-scale engineering projects, where transportation budgets frequently exceed initial estimates, necessitating the establishment of dynamic transportation monitoring mechanisms, the adoption of optimized vehicle scheduling strategies, the refinement of detailed transportation route planning, and the reinforcement of effective risk control throughout the process. These measures ensure the precise alignment of diverse building materials with the spatial and temporal requirements of construction stops. Such a full-cycle transportation management system essentially forms the foundational support framework enabling modern engineering projects to achieve cost reduction and efficiency enhancement objectives. The pressing engineering demand to satisfy both the timeliness and reliability of route planning solutions underscores the substantial scientific research value of this field.

Reinforcement Learning (RL) [11][12][13] constitutes a type of optimization algorithms that derive policies from extensive datasets or experiential trajectories. Its fundamental principle involves extracting latent features from historical interactions and iteratively refining the decision-making process to enhance overall performance. The “learning” process refers to approximating a function that maps input data features to output results. The model continuously adjusts its internal parameters through repeated training to ensure that the outputs closely approximate the target values, thereby achieving high predictive accuracy [14][15]. Nevertheless, RL is subject to the following two limitations:

1. Limitations on Problem Scale: Scaling reinforcement learning algorithms remains a pivotal area of research. Current RL frameworks frequently face challenges including high computational dimensionality and expansive state spaces when dealing with large-scale or highly complex combinatorial optimization problems. These challenges often result in issues such as multicollinearity and overfitting. Although various methods aimed at improving model expressiveness and convergence efficiency, the quality of solutions tends to deteriorate substantially when these algorithms are directly applied to large-scale VRPs.

2. Limited Generalization Capability: The generalization performance of neural network models is predominantly influenced by the distributional properties of the training data. The input-output mappings acquired by RL models tend to be effective primarily within the confines of specific data distributions and often fail to generalize to scenarios characterized by substantially different data distributions. This characteristic makes RL models highly dependent on the test environments and data features in practical applications, thereby constraining their universality and adaptability.

To overcome these challenges, Graph Neural Networks (GNNs), a deep learning approach designed for modeling graph-structured data, present novel opportunities to improve model generalization [16] and structural awareness. GNNs retain the original graph structure and node attributes while aggregating information from neighboring nodes to achieve comprehensive global feature representation. Additionally, they are capable of projecting high-dimensional graph data from a D-dimensional space into a comparatively lower-dimensional space, thereby facilitating subsequent optimization processes. In contrast to traditional neural networks that rely solely on independent node attributes for learning, GNNs excel at capturing the topological relationships and semantic associations among nodes. Their parameterized aggregation mechanisms, combined with first-order dynamic optimization strategies, confer broad adaptability and robust modeling capabilities across diverse graph-theoretic problems.

Based on the aforementioned research background, this study proposes an innovative route planning model that integrates graph neural networks with reinforcement learning algorithms to enhance both the solution efficiency and generalization capability of vehicle routing problems within complex graph structures. This study focuses on the capacitated-unconstrained vehicle routing problem, incorporating traffic congestion indices, and improves an efficient route planning algorithm that combines Graph Convolutional Networks (GCN) and Q-learning. The capacitated-unconstrained route planning problem can fundamentally be regarded as a Traveling Salesman Problem (TSP) [17][18]. Given that vehicles are not constrained by capacity during transportation, they are not required to return to the depot stop while serving multiple customers, thereby exhibiting typical single-tour visitation characteristics. This paper first provides a systematic description of the problem background, followed by the construction of a corresponding mathematical modeling framework. Comparative experiments conducted on simulated datasets demonstrate the advantages of the proposed model in route selection and cost optimization. The experimental results indicate that, relative to traditional greedy algorithms and standalone reinforcement learning approaches, the integrated model leveraging graph neural networks and reinforcement learning produces superior route planning solutions, substantially reduces travel distance, and achieves significant transportation cost savings.

2. RELATED WORKS

The VRP [19] is a classic combinatorial optimization problem that has been the subject of extensive research over a long period. Its fundamental formulation involves determining an optimal delivery route for multiple stops with specific service demands (i.e., customers) under a given set of constraints. The objective is to enable service vehicles to visit each customer location in a prescribed sequence and complete the assigned tasks while minimizing the total transportation cost. VRP has wide-ranging practical applications, including logistics operations such as goods delivery and collection, as well as urban traffic management activities like taxi route planning, material transportation scheduling, and school bus dispatching. The transportation network in VRP can be modeled as a graph structure $G(V, E)$, where the vertex set V denotes warehouses and customer points (or stops, i.e. its location), and the edge set E represents the paths connecting these stops. Each edge $e_{i,j}$ is generally associated with a cost parameter $c_{i,j}$, which may correspond to distance, time, or a comprehensive transportation cost.

G.B. Dantzig and J.H. Ramser [20] were the pioneers in studying the VRP. Fisher and colleagues categorized the evolution of VRP solution algorithms into three distinct phases: the first phase (1959–1970) emphasized simple heuristic algorithms, such as greedy methods [21]; the second phase (1970–1980) primarily utilized exact algorithms; and the third phase (from 1980 onward) saw widespread adoption of heuristic and metaheuristic approaches. In recent years, advancements in computational power and the emergence of machine learning techniques have led to the increasing application of reinforcement learning algorithms for solving VRP, marking a novel research direction in this domain.

Exact algorithms, which are theoretically capable of obtaining globally optimal solutions, can be primarily categorized into three categories: tree search methods (e.g., branch-and-bound [22]), dynamic programming [23], and integer linear programming approaches (such as column generation algorithms [24]). VRP research results in the 1970s was grounded in these algorithms, resulting in three exact solutions for Capacity-constrained VRP (CVRP) and Dynamic VRP (DVRP). By employing tree search methods, these studies also integrated shortest K-D trees and q-route techniques to establish an algorithmic framework for computing lower bounds during the search process. Although these algorithms possess clear logical structures and well-defined frameworks, their computational complexity increases exponentially with the number of stops, thereby hindering the efficient resolution of large-scale problems and limiting their practical applicability.

According to the classification proposed by Jean-François Cordeau and Gilbert Laporte [25], traditional heuristic algorithms can be categorized into constructive heuristics and two-phase heuristics. Constructive heuristics aim to minimize costs by incrementally building feasible solutions, with the savings algorithm serving as a representative example. Two-phase heuristics typically involve an initial clustering of customers, followed by the design of service routes within each cluster that satisfy relevant constraints. The fundamental aspect of heuristic methods lies in constructing a neighborhood space around the initial solution and optimize route quality through adjustments to stop positions. Due to their undirected search processes, traditional heuristics generally conduct shallow searches within a limited solution space and often encounter difficulties in escaping local optima. In contrast, metaheuristic algorithms incorporate sophisticated search strategies and

memory mechanisms, enabling exploration of a broader solution space and thereby improving both solution quality and diversity. Compared to traditional heuristics, metaheuristics can enhance solution quality by approximately 3% to 7% and demonstrate better stability and convergence when addressing medium-scale VRP problems. This paper compares the greedy algorithm within heuristic methods, which operates by selecting the locally optimal choice at each step. In path planning, the greedy algorithm starts from a starting point and, at each iteration, selects the nearest unvisited stop as the subsequent target until all stops have been visited, ultimately returning to the origin to form a closed route. The central premise of this approach is that a sequence of locally optimal decisions will yield a near-optimal overall path. Although the greedy algorithm is computationally efficient and intuitive—making it suitable for small-scale problems or real-time applications—it focuses exclusively on immediate best options, disregarding the global structure. This limitation may cause the algorithm to become trapped in local optima, resulting in suboptimal final path costs.

RL, as an optimization algorithm endowed with autonomous learning and decision-making capabilities, continuously acquires feedback through interactions between an agent and its environment. It enables the updating and optimization of strategies within dynamic environments and is regarded as a promising alternative for solving combinatorial optimization problems like VRP. This study concentrates on reinforcement learning-based path optimization algorithms [26][27], incorporating GCNs to extract embeddings of graph structures, which are integrated with the perceptual and optimization capabilities of Q-learning. By employing deep learning techniques for policy function modeling, alongside mechanisms including experience replay, reward systems, and state-action value functions, the approach approximates the value of state-action pairs [28]. This facilitates the learning and iterative refinement of the optimal path planning strategy, thereby enabling the model to progressively acquire the most effective routing policy.

Graph convolutional networks primarily utilize convolution operations to learn a mapping function $f(\cdot)$, which aggregates the features x_i of stop v_i with the features of its neighboring stops $x_i(j \in Nj)$ to generate a new feature vector for stop x_i [29]. Due to their capacity to effectively aggregate neighborhood information, GNNs have demonstrated robust performance in handling structured sparse data and have been extensively applied to solve various complex optimization problems. Battaglia et al. developed a GNN-based interaction model to capture interactions among elements in complex physical systems, successfully predicting dynamic processes and abstracting experimental characteristics. Hamilton et al. employed GNNs for knowledge graph representation learning tasks, while Cui et al. incorporated them into traffic flow prediction models to enhance the modeling of spatiotemporal dependencies. Additionally, Kim et al. constructed a hierarchical model using Graph Attention Networks (GAT) to predict corporate performance in the stock market, effectively aggregating and leveraging multiple types of relationships. Furthermore, Gasse et al. designed a graph convolutional neural network to learn variable selection strategies within branch-and-bound algorithms, thereby demonstrating the potential of GNNs in combinatorial optimization search strategies.

The combinatorial optimization of VRP involves an input structure of an unordered set of stops. Dependencies among these stops are represented through edge weights, such as distance, time, or cost. Although the problem is inherently graph-structured and can be addressed using classical frameworks like Convolutional Neural Networks (CNNs), these methods generally depend on ordered sequences of stops and are unable to fully capture the high-dimensional interactions between stops or preserve the graph's topological and adjacency properties. Consequently, this limitation restricts the model's capacity to comprehensively represent the problem's structure.

GCNs inherently represent stop features and their adjacent edges within graph structures, rendering them well-suited to the non-sequential input format characteristic of VRP. They have garnered significant research attention in recent years. Dai et al. were pioneers in applying graph convolutional models to the TSP, introducing a dynamic graph embedding update mechanism that iteratively refines graph representations throughout the solution process, thereby improving model convergence. To tackle ultra-large-scale problems involving up to approximately 10,000 stops, Fu et al. proposed an end-to-end solution framework integrating graph transformation and heatmap fusion techniques. Their approach employed graph sampling to generate multiple subgraphs from the original large graph, utilized a pre-trained GCN model to predict edge probability matrices, and subsequently fused the solution spaces of these subgraphs to approximate an optimal path. In summary, GCNs demonstrate distinct advantages in handling unstructured graph data, capturing complex dependencies among stops, and enhancing the structural representation capabilities of reinforcement learning algorithms. These characteristics render them particularly suitable for combinatorial optimization problems characterized by intricate stop relationships and non-sequential input structures, such as the VRP.

In real-world traffic networks, the travel time of transport vehicles is influenced by congestion, resulting in increased travel durations and transportation costs. Previous research has primarily focused on optimizing either the shortest travel time or the lowest transportation cost, without adequately accounting for conditions arising from road congestion. To address the limitation, this study present study introduces a road congestion index, facilitating the identification of routes that minimize travel time, distance, transportation cost, and carbon dioxide emissions simultaneously under multi-dimensional cost minimization criteria. This approach offers more reliable route selection for the transportation of construction materials.

3. DEVELOPMENT OF REINFORCEMENT LEARNING MODELS

3.1 Mathematical Model

3.1.1 Definition of Symbols

To formally represent the capacitated TSP under investigation, this section develops the corresponding mathematical model and delineates the key parameters and variables involved. Initially, the primary symbols utilized in the model are defined, as presented in the following Table 1:

Table 1. The Symbols Definition Table

Symbol	Explanation
V	The set of all points includes one warehouse and n customer points, $V = \{0, 1, ..., n, n+1\}$, where 0 represents the warehouse.
N	The set of customer points is $C = \{1, ..., n\}$. Here, $C0$ is also defined as $C0 = C \cup \{0\}$, representing the set of customer points along with the warehouse.
C	Maximum vehicle capacity.
K	Total number of vehicles.
A	Set of routes.
$D_{i,j}$	Distance from customer point $i \in$ V to customer point $j \in V$.
u_i	Cumulative service quantity of the vehicle after servicing stop i.
q_i	Demand quantity of customer point j to be visited at the next time step.
$X_{i,j}$	Binary decision variable, $Xi,j = \{0,1\}$, where a value of 1 indicates that the planned route includes traveling from $i \in V$ to $j \in V$, with $i \neq j$; otherwise, it is 0.
$X_{i,0}$	Routes ending at the warehouse point.
$X_{o,j}$	Routes starting from the warehouse point.
$CI_{i,j}$	The congestion index of the road segment from stop i to stop j.
$C_{i,j}$	The weight of edge $e_{i,j}$ corresponding to stop (i, j), which in the Capacitated Vehicle Routing Problem (CVRP) refers to the Euclidean Distance between stops (i, j).

3.1.2 Mathematical Model

The TSP can be rigorously formulated as an integer programming problem. The corresponding mathematical model is presented as follows:

(1) Objective Function

The objective of the TSP is to determine the shortest possible route that begins at a specified starting point, visits each customer stops exactly once, and returns to the original starting point. Based on this formulation, the objective function can be expressed as follows:

$$Min \sum i, j \in AC_{i,j}X_{i,j} \quad (1)$$

In various types of TSP problems, the edge weight $C_{i,j}$ represents various meanings, such as distance, time, and so on. In the path planning problem studied in this paper, $C_{i,j}$ represents the Euclidean distance between stop i and stop j. This distance is the geometric straight-line distance between the two stops in space, calculated by the following formula:

$$d=\sqrt{(X_J - X_i)^2 - (Y_j - Y_i)^2} \quad (2)$$

(2) Variable Constraints

Where $x_{i,j}$ is a binary decision variable used to indicate whether the path from stop i to stop j is selected:

$$X_{i,j} \in [0,1] i, j \in A \quad (3)$$

This variable defines the set of edges in the final path and constitutes a fundamental component in formulating the objective function and constraints of the TSP.

(3) Flow Balance Constraint

Within the scope of the TSP, it is essential to ensure that each stop is visited exactly once. This requirement is enforced by the constraint that each stop must have exactly one incoming path and one outgoing path. Specifically, for any given stop i, there exists exactly one path entering stop i from another stop, and correspondingly, exactly one path leaving stop i is allowed. This constraint can be formally represented as follows:

$$\sum_{j \in V, j \neq i} x_{i,j} = 1, i \in N \quad (4)$$

$$\sum_{i \in V, i \neq j} x_{i,j} = 1, j \in N \quad (5)$$

(4) Integrity Constraint

In the TSP, the vehicle's route must constitute a complete closed loop, commencing from a designated stop, visiting all target stops exactly once, and ultimately returning to the initial stop. In this study, the starting stop can be represented as the warehouse stop (denoted as stop O). Therefore, the route to be determined must begin at stop O and also end at stop O.

$$X_{I,0} = 1 \quad (6)$$

$$X_{0,j} = 1 \quad (7)$$

To satisfy this requirement, the model incorporates the following two constraints: Equation 4 ensures that there is exactly one route originating from stop O, while Equation 5 ensures that there is exactly one route returning to stop O. Collectively, these constraints guarantee that the entire path constitutes a unique, valid closed loop with the warehouse serving as both the starting and ending point, a critical condition for confirming the path's validity.

3.2 Calculation Formula for Transportation Costs

Let C be the total transportation cost, $d_{i,\ i+1}$ be the distance from stop i to stop i+1, $CI_{i,j}$ be the congestion index of the road segment from stop i to stop j. The n be the number of stops.

$$C = \sum_{i=0}^{n} (d_{i,i+1} / AverageSpeed) * CI_{i,i+1} \quad (8)$$

3.3 Construction of a Combined Model Using Graph Convolutional Networks (GCN) and Q-Learning Algorithm

3.3.1 Markov Decision Process

Reinforcement learning is based on a decision-making process without time delay. The objective is to begin at the warehouse stop (labeled 0), visit all construction sites sequentially (stops 1 through 10), and finally return to the warehouse. The aim is to minimize the total transportation cost, defined as the sum of the path distances. In the model proposed in this chapter, the reinforcement learning process is formalized as a Markov Decision Process (MDP). For the convenience of subsequent modeling and training, the state space, action space, reward function, and state transition mechanism are defined as follows:

(1) State

The state S is the information presently accessible within the environment for the purpose of decision-making. In the model, this state encompasses data from all stops within the sample, including each stop's identifier, coordinate position, and the

edge weights between stops, which are typically expressed as Euclidean distances. Consequently, the state S can be formally defined as follows:

$$S = \left\{\{(x_0, y_0), (x_1, y_1), \ldots, (x_j, y_j)\}, \{(a_{0,0}, o_{0,0}), (a_{0,1}, \delta_{0,1}), \ldots, (d_{j,j}, \delta_{j,j})\}\right\} \quad (9)$$

(2) Action

Action a denotes the choice behavior of the current agent in a given state, that is, moving from the current stop to an unvisited stop. In the TSP scenario, the action space consists of the set of unvisited stops $G \subseteq V$, where V is the set of all stops. Therefore, action $a \in G$ indicates selecting an unvisited stop as the next hop target.

(3) Reward

Within the reinforcement learning framework, the reward function quantifies the feedback received following the agent's execution of a specific action. In this model, the reward r is defined as the negative value corresponding to the change in path length resulting from the action a.

This design encourages the agent to choose paths with lower costs, which helps minimize the overall path length. This strategy aligns with the objective function of the TSP mathematical model—"minimizing the total path length"—and effectively drives the model to learn the optimal visiting order, thereby improving the quality of the final solution.

(4) State Transition

After the agent's selection and execution of action a, the environment transitions to the subsequent state. In the context of the TSP, this state transition involves moving from the current stop to the selected next unvisited stop and removing that stop from the set of candidates, thereby generating a new state. This process of state transition represents the state evolution mechanism within the Markov Decision Process (MDP).

(5) Discount Factor

The discount factor represents the degree of emphasis on future rewards and takes a value between 0 and 1. The closer γ is to 1, the more the model values long-term returns.

3.3.2 Graph Convolutional Network-Based Model

In recent years, research on the TSP has undergone continuous and in-depth exploration, leading to a steady expansion in problem-solving scale and an increase in problem complexity. Practical engineering applications, such as material transportation path optimization, impose higher demands on both computational efficiency and solution quality of algorithms. This paper proposes a transit time-based path optimization model that integrates Graph Convolutional Networks (GCN) with Q-learning algorithms. The model capitalizes on the strengths of GCN in graph structure modeling to effectively encode adjacency relationships among stops and capture graph topological features. Utilizing a Q-learning strategy within a reinforcement learning framework, the agent sequentially selects visitation paths on the graph and iteratively updates its policy to optimize the objective function. The proposed model aims to minimize total transportation costs while ensuring that all target stops are visited, thereby offering an efficient and practical solution for large-scale, complex engineering path planning problems.

(1) Adjacency Matrix and Feature Initialization

Construct the adjacency matrix and initialize the feature encoding matrix for each stop, where N denotes the number of stops. To allow each stop to aggregate its own features while simultaneously incorporating information from its neighboring stops, the matrix must be normalized and augmented with self-loops.

$$\hat{A} = A + I \quad (10)$$

$$\tilde{A} = D^{\frac{-1}{2}} \hat{A} D^{-\frac{1}{2}} \quad (11)$$

Here, D is the degree matrix, satisfying $D_{ii} = \sum_j \hat{A}_{ij}$

(2) GCN Embedding Computation

A two-layer GCN network is used to generate embedding vectors for each stop:

$$H^{(1)} = ReLU\ (\hat{A}XW^{(1)}) \quad (12)$$

$$H^{(2)} = ReLU\ (\hat{A}H^{(1)}W^{(2)}\ ) \tag{13}$$

Here, $W^{(1)}$ and $W^{(2)}$ are the training weight matrices, ReLU is the activation function, and the output $\mathrm{H} = \mathrm{H}^{(2)} \in \mathrm{R}^{\mathrm{N}\times\mathrm{F}}$ represents the structural embedding of each stop. These vectors will be used as part of the reinforcement learning state to enhance the policy model's understanding of the graph structure.

3.3.3 Q-Learning Path Optimization Model

(1) State Definition

At each decision moment *t*, the agent's state consists of two parts: the embedding vector of the current stop $\mathrm{h_i} \in \mathrm{R^d}$ and the mask vector of the stops already visited $v_i \in \{0\ \cdot\ 1\}^N$, which together form the complete state representation:

$$s_i = [\mathrm{h_i}\ \cdot\ v_i] \in \mathrm{R^{d+N}} \tag{14}$$

(2) Space Definition

The set of actions in the current state consists of all unvisited stops:

$$A_i = \{j \in V | V_t\ (j\ ) = 0\} \tag{15}$$

This allows the agent to jump from the current position to any unvisited stop.

(3) Design of the Reward Function

The reward function is designed to minimize transportation costs. At each step, the immediate reward received for moving from the current stop i to the next stop j is:

$$r_t = -d_{ij} \tag{16}$$

(4) Update of the Q-value function

Q-learning uses the following update formula to learn the state-action value function:

$$\mathrm{Q(s_t\ \cdot\ a_t)\ \leftarrow Q(s_t\ \cdot\ a_t)\ + [r_t + rmax_{a'}Q(s_{t+1}\ \cdot\ a')\ - Q(s_t\ \cdot\ a_t)\ ]} \tag{17}$$

Where α is the learning rate, controlling the magnitude of each update. r is the discount factor, representing the weight of future rewards. s_{t+1} is the next state. α' represents all possible actions in the next state.

(5) Policy Selection: Greedy Policy

To achieve a balance between exploration and exploitation, this paper adopts the ε-greedy policy in the action selection strategy. At each decision step:

With probability ε, a random action is selected (i.e., uniformly sampled from the set of available actions in the current state) to increase the exploration of the policy; With probability 1-ε, the action with the highest current Q-value is selected (i.e., greedy selection) to exploit the currently learned optimal policy.

This mechanism balances exploration of the environment and approximation of the optimal path, helping to prevent the agent from getting stuck in local optima while accelerating the overall convergence of the strategy.

$$a_t = \begin{cases} random\ (A_t\ )\ \cdot & if\ rand < \varepsilon \\ argmax_{a\in A_t}Q(s_t\ \cdot a) & otherwise \end{cases} \tag{18}$$

3.4 MODEL TRAINING PROCEDURE

The complete training procedure of the model is outlined as follows:

(1) Initialize the graph structure, the adjacency matrix, and the distance matrix.

(2) Construct the stop feature matrix X and derive stop embeddings using a GCN.

(3) Initialize the Q-table with all entries set to zero.

(4) Iterative Training:

The process begins at the warehouse, where the current stop and visitation status are recorded. Subsequently, the next stop is selected according to the ε-greedy policy. The reward is then calculated, and the Q-table is updated accordingly. This

procedure of state transitions continues until all construction sites have been visited. Finally, the route returns to the warehouse, and the Q-value of the last transition is updated.

(5) Record the total path cost for each episode to facilitate convergence analysis.

(6) Upon completion of the training process, derive the optimal policy path utilizing the Q-table.

4. SIMULATION RESULTS AND ANALYSIS

4.1 EXPERIMENTAL SETUP

To evaluate the performance of various path optimization algorithms in engineering material transportation problems, this study establishes a standardized experimental environment and conducts a comparative analysis of three representative algorithms: the Greedy algorithm, the conventional Q-learning algorithm, and the GCN+Q-learning hybrid model.

(1) Graph Structure Construction

The experiment is conducted within the context of engineering transportation in a specific region, involving the construction of undirected graph structures comprising 11, 21, and 31 stops, respectively. Each graph includes one warehouse and 10, 20, or 30 delivery sites (stops). The paths connecting the stops are bidirectional, allowing travel in both directions. Transportation distances between stops are determined using the Euclidean distance calculated from simulated two-dimensional coordinates. All edge weights are symmetric, i.e., $d_{i,j} = d_{j,i}$.

(2) Transportation Task Setting

A single transport vehicle departs from the warehouse, sequentially delivers materials to all sites (stops), and subsequently returns to the warehouse. This model does not account for constraints such as vehicle capacity or delivery time windows. The primary objective is to determine a closed route that visits all delivery sites while minimizing the total transportation cost.

Transportation cost is defined as the total path length (unit: kilometers): $C=\sum_{i=0}^{n} d_{\pi_i,\pi_{i+1}}$, where π represents the order of the path $\pi_0 = \pi_{n+1}$.

(4) Unified Experimental Parameters

To ensure fairness, all algorithms were executed using the same graph structure and transportation tasks under the following conditions:

The route in case 1 comprises 11 stops, including one warehouse and ten construction sites. The distance for each segment ranges from 10 to 100 kilometers, based on simulated data. For reinforcement learning involving the 11 stops, the number of training episodes was set to 300 for both Q-learning and the combined GCN with Q-learning approaches.

The route in case 2 comprises 21 stops, including one warehouse and 20 construction sites. The distance between each pair of stops ranges from 10 to 100 kilometers, based on simulated data. For reinforcement learning involving the 21 stops, the number of training episodes was set to 2,000 for both Q-learning and the combined Graph Convolutional Network (GCN) with Q-learning approaches.

The route in case 3 comprises 31 stops, including one warehouse and 30 construction sites. The distance between each pair of stops ranges from 10 to 100 kilometers, based on simulated data. For reinforcement learning involving these 31 stops, both Q-learning and the combined Graph Convolutional Network (GCN) with Q-learning methods were trained over 5,000 episodes each.

Learning rate (α): 0.1, discount factor (γ): 0.9, exploration rate (ε): 0.1.

4.2 EXPERIMENTAL RESULTS AND ANALYSIS

4.2.1 Experimental Results and Analysis of 11 Stops

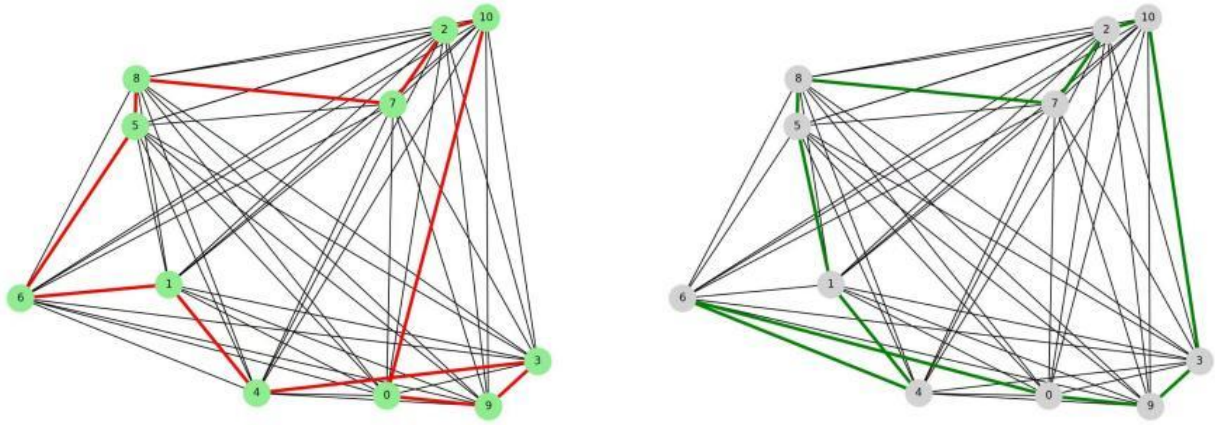
Table 2. The Comparison Table of Experimental Results Analysis for 11 stops

Algorithm	Cost (km)	Optimal path
Greedy	305.37	0,9,3,4,1,6,5,8,7,2,10,0
Q-learning	460.77	The Q-value determines the decision path, ultimately returning to the warehouse (the path is not fixed).

GCN+Q-learning	272.15	0,9,3,10,2,7,8,5,1,4,6,0

As presented in Table 2, within the engineering material transportation path optimization task established in this study, experimental comparisons among the greedy algorithm, the traditional Q-learning algorithm, and the GCN + Q-learning hybrid model reveal that the GCN + Q-learning model outperforms the others in terms of path planning quality and transportation cost control. This outcome highlights the model's robust generalization capability and its potential advantages for practical applications.

From the perspective of total transportation cost, the greedy algorithm selects the nearest unvisited stop at each step during path selection, exemplifying a typical local search method. This approach is straightforward to implement and computationally efficient; however, due to its lack of consideration for the global structure, it is prone to becoming trapped in local optima, leading to suboptimal overall path performance. Experimental results indicate that the final path length obtained by this method in the present case is 305.37 kilometers. Although efficient in execution, the method fails to adequately balance global optimality. In contrast, the traditional Q-learning algorithm maintains the state-action value function in a tabular form and can progressively learn the visitation policy through repeated interactions. Nevertheless, because its state representation primarily relies on the current stop ID and the set of visited stops—without effectively modeling stop topology or overall graph features—the policy learning process is unstable. Consequently, this model converged to a path cost of 460.77 kilometers, exceeding that of the greedy algorithm, thereby demonstrating its difficulty in achieving stable and effective policy optimization when addressing complex graph structures.



(a) (b)

Figure 1. Optimal paths determined by (a) the Greedy Algorithm and (b) the GCN combined with Q-learning Algorithm for 11 stops.

Figure 1(a) and Figure 1(b) illustrate the shortest paths generated by two distinct algorithms. It can be seen that compared to the previously mentioned methods, the GCN+Q-learning hybrid model significantly improves solution quality and optimization performance. This model effectively exploits the GCN's capacity to represent graph structures by producing structure-aware embedding vectors for each stop, which are subsequently input into the reinforcement learning module for action selection. Leveraging the representational power of GCN, the model captures the relationships among stops from a global graph perspective, thereby facilitating more globally informed path planning decisions. Experimental results indicate that after 300 training episodes, the model reduced the total path cost to 272.15 kilometers, representing a decrease of 33.22 kilometers (approximately 10.88%) relative to the greedy algorithm and a reduction of 188.62 kilometers (up to 40.93%) compared to traditional Q-learning. These findings underscore the model’s significant performance advantage.

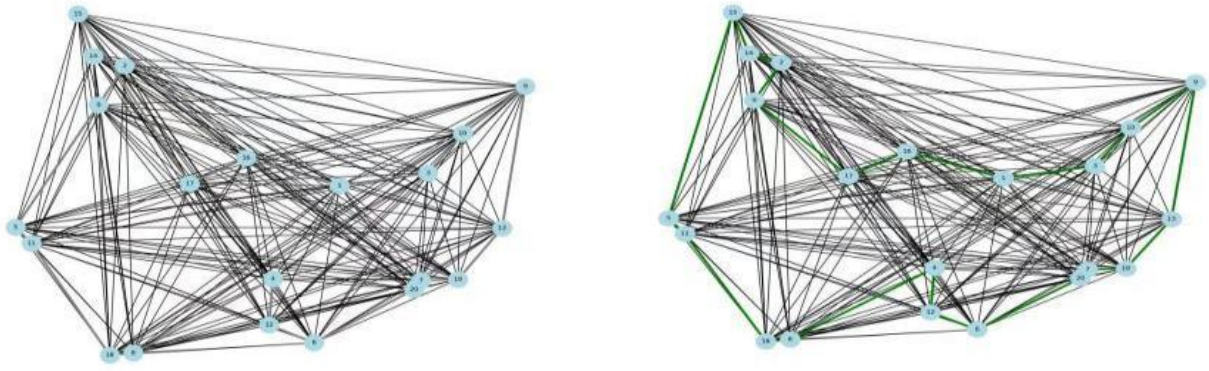
4.2.2 Experimental Results and Analysis of 21 Stops

Table 3. The Comparison Table of Experimental Results Analysis for 21 stops

Algorithm	Cost (km)	Optimal path
Greedy	379.74	0,2,14,15,17,16,1,3,10,9,13,19,7,20,6,12,4,8,18,11,5,0
Q-learning	555.46	0,14,2,15,7,20,19,1,16,17,12,4,10,3,9,13,6,8,18,11,5,0
GCN+Q-learning	378.75	0,2,14,15,5,11,18,8,4,12,6,20,7,19,13,9,10,3,1,16,17,0

As presented in Table 3, Figure 2(a), and Figure 2 (b), to more accurately represent the actual experimental procedure, this study utilizes the actual path costs from the initial training round for both the Q-learning and GCN + Q-learning models as

the baseline to reconstruct the cost trend graphs for the three algorithms. The findings indicate that the Q-learning model exhibits low path selection efficiency during the initial phase, with a first-round transportation cost of 1033.51 km, whereas the GCN + Q-learning model demonstrates an even higher initial cost, reaching 1113.44 km.



(a) (b)

Figure 2. Optimal Path of (a) the Greedy Algorithm, and (b) GCN+Q-learning Algorithm for 21 stops.

The path selection efficiency of the Q-learning model during the initial stage is relatively low; specifically, the transportation cost in the first round reaches 1033.51 km. In comparison, the initial cost associated with the GCN + Q-learning model is even higher, amounting to 1113.44 km. This can be attributed to the fact that, at the onset of training, the GCN embeddings have not yet fully captured the pertinent information within the graph structure, leading to suboptimal path selection strategies. However, as the number of training iterations increases—particularly after 2000 iterations—the performance of the GCN + Q-learning model improves markedly, ultimately converging to a cost of 378.75 km. This result successfully surpasses the fixed cost of the greedy algorithm, which stands at 379.74 km. Conversely, the traditional Q-learning model converges to a cost of 555.46 km; although this represents a clear improvement relative to its initial state, it does not exceed the performance of the greedy strategy.

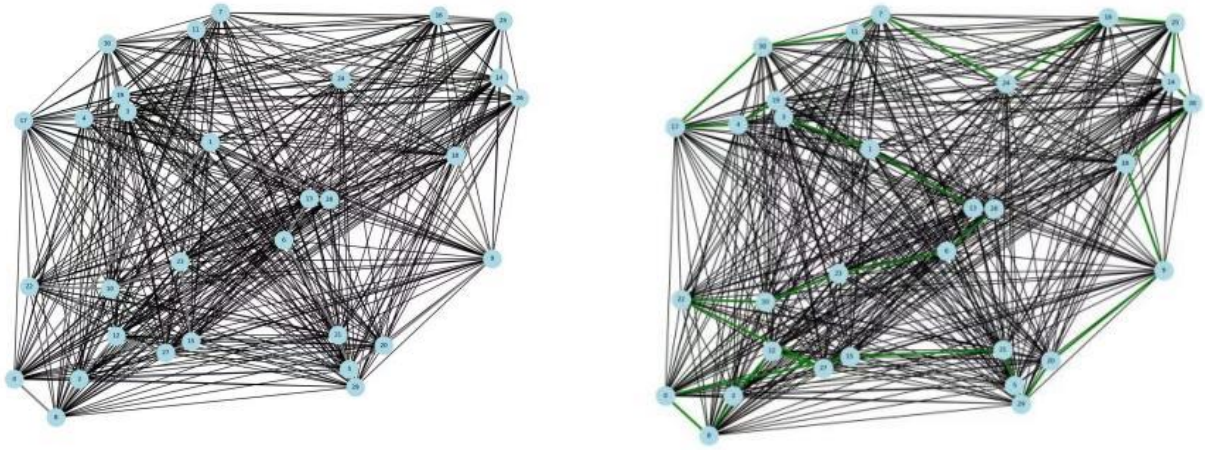
Overall, the results indicate that the GCN + Q-learning model exhibits a robust convergence capability and achieves cost reduction at a significantly faster rate compared to the standard Q-learning approach. By incorporating graph structural information to effectively perceive and represent states, the model substantially enhances the quality of path decision-making. Within the reinforcement learning framework, this model consistently outperforms the greedy algorithm, thereby demonstrating the practical applicability of graph neural networks in engineering path optimization problems.

4.2.3 Experimental Results and Analysis of 31 Stops

As presented in Table 4, Figure 3(a), and Figure 3(b), a transportation task involving one warehouse and 30 construction stops reveals notable differences in total travel distance (i.e., transportation cost) among the three algorithms evaluated. The greedy algorithm, employed as a baseline solution, yields a total travel distance of 523.93 km, which serves as a reference fixed cost. Notably, the greedy algorithm requires no training process and directly produces a relatively efficient solution. In contrast, the initial policy of the Q-learning reinforcement learning algorithm is highly inefficient, with an initial total travel distance of 1985.73 km—substantially exceeding that of the greedy algorithm. This inefficiency reflects the near-random nature of the routes generated by Q-learning prior to training. Throughout the progressive training process, the performance of Q-learning improves markedly, ultimately converging to a total distance of 861.16 km. This outcome indicates that, following training, Q-learning reduces the total cost by approximately 56.6%, thereby demonstrating a degree of optimization capability. Nevertheless, its final result remains approximately 64% higher than that of the greedy algorithm and has not yet matched the baseline solution.

Table 4. The Comparison Table of Experimental Results Analysis for 31 stops

Algorithm	Cost (km)	Optimal path
Greedy	523.93	0,8,2,12,27,15,23,10,22,17,4,3,19,30,11,7,24,16,25,14,26,18,9,20,5,29, 21 ,6,13,28,1,0
Q-learning	861.16	0,22,27,15,30,3,19,24,13,28,21,5,12,10,23,8,2,4,11,1,6,29,20,9,18,14, 26 ,25,16,7,17,0
GCN+Q-learning	507.98	0,8,2,12,27,22,10,23,6,28,13,1,3,19,4,17,30 ,11,7,24,16,25,14,26,18,9,20,29,5,21,15,



(a) (b)

Figure 3. Optimal Path of (a) the Greedy Algorithm, and (b) GCN+Q-learning Algorithm for 31 stops.

The GCN + Q-learning algorithm demonstrates superior performance in cost optimization. Initially, the total route distance is approximately 2006.15 km, which is comparable to the initial condition observed in Q-learning, indicating the absence of optimization at the outset. Following reinforcement learning facilitated by the Graph Convolutional Network (GCN), the total cost associated with the GCN + Q-learning approach decreases rapidly, ultimately reaching 507.98 km. This represents a reduction of approximately 74.7% relative to the initial value, significantly surpassing the reduction achieved by Q-learning alone and yielding a cost lower than that of the greedy algorithm. In other words, the GCN + Q-learning method identifies a transportation route shorter than that produced by the greedy strategy (reducing distance by approximately 3%) and attains the most favorable total cost outcome.

From the perspective of training convergence, the learning speed and stability of the GCN + Q-learning algorithm are demonstrably superior to those of pure Q-learning. The greedy algorithm requires no training and, consequently, does not encounter convergence issues. The primary differences between the two reinforcement learning algorithms are observed in the fluctuations of the cost curve throughout the training iterations. During the initial training phase, Q-learning employs an exploration strategy, which results in very high initial cost values and a relatively slow initial decline. As the algorithm progressively accumulates experience, the cost decreases steadily; however, this process is protracted. The convergence curve indicates that the total distance in Q-learning gradually approaches a relatively favorable value, stabilizing around 861 km after numerous iterations. This algorithm tends to become trapped in local optima during training, hindering further improvements, as evidenced by the curve flattening and failing to converge further toward the greedy solution.

The GCN + Q-learning algorithm exhibits accelerated convergence. Owing to the GCN's capacity to capture global topological information, the agent is able to make more informed decisions at an early stage, resulting in a pronounced decline in the cost curve during the initial phase. Experimental results indicate that, with fewer training iterations, the total distance associated with GCN + Q-learning rapidly decreases to a level comparable to that of the greedy algorithm and continues to decline steadily thereafter. In comparison to pure Q-learning, the convergence curve of GCN + Q-learning descends more sharply and exhibits reduced fluctuations, reflecting enhanced learning efficiency and more stable training performance. The GCN + Q-learning algorithm converges to approximately 507.98 km, achieving superior convergence values within a shorter training duration. These findings demonstrate that integrating graph convolutional networks into reinforcement learning effectively accelerates the learning process, enabling the algorithm to approach the global optimum more rapidly.

The specific transportation routes generated by the three algorithms exhibit distinct structural differences, which reflect their respective strategic characteristics and coherence. The greedy algorithm tends to select the nearest construction site at each step, resulting in routes that are generally locally coherent; that is, consecutively visited sites are typically in close proximity, thereby minimizing single-step travel distances. In most cases, the greedy strategy produces routes without excessive repetition or detours, leading to relatively straightforward overall paths. However, because it considers only the nearest neighbor at each decision point, the greedy algorithm may fail to identify a globally optimal layout. Consequently, it

may be necessary to return over long distances from the most remote construction site back to the warehouse or to take detours to visit previously skipped distant sites, which can cause a slight increase in total mileage.

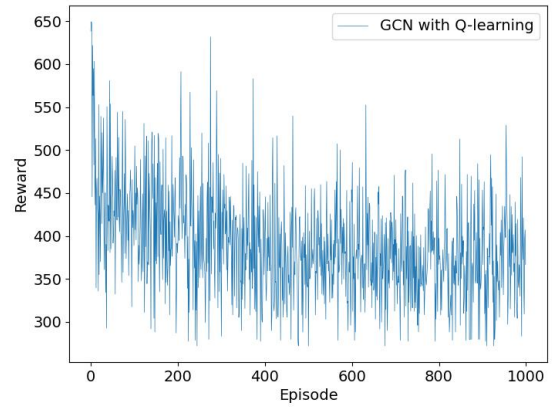
The routes generated by the Q-learning algorithm illustrate the progressive enhancement characteristic of reinforcement learning strategies. During the initial training phase, the routes produced by Q-learning are nearly random, often involving inefficient back-and-forth movements between construction sites across different regions, resulting in poor coherence and evidently suboptimal sequences. As training advances, Q-learning increasingly avoids the poorest decisions, leading to routes that are markedly improved relative to the initial state. The converged routes demonstrate a degree of route continuity, exhibiting notable improvements in the connectivity between adjacent sites compared to random strategies. Nevertheless, because Q-learning relies solely on cumulative reward signals and does not explicitly incorporate global information, the resulting route strategies may remain suboptimal. Specifically, the routes planned by Q-learning can include inefficient jumps or sequences, such as the omission of nearest-neighbor selections akin to those employed by greedy algorithms to maximize local rewards. Consequently, certain route segments are disproportionately long, and the overall coherence and compactness of the routes are inferior to the optimal solution.

The routes formed by the GCN + Q-learning algorithm exhibit the highest degree of rationality and coherence, closely approximating globally optimal paths typically achieved through manual optimization or exact algorithms. By leveraging the capabilities of the graph convolutional network, the algorithm accounts for the spatial distribution and inter-site distances of all construction locations during decision-making. This enables effective coordination in sequencing visits to both proximal and distal sites. The resulting routes generated by GCN + Q-learning generally adhere to a zonal cruising pattern, wherein deliveries are first made to clusters of nearby sites within a localized area before proceeding to other regions, thereby minimizing redundant back-and-forth travel across different zones. For distant sites, the algorithm flexibly schedules visit times to avoid positioning them at the end of the route, which would otherwise lead to increased energy consumption due to lengthy return trips. Furthermore, the GCN + Q-learning approach ensures superior distance continuity within each route segment, maintaining strong connectivity between adjacent points and reducing globally suboptimal jumps commonly observed in greedy algorithms. Overall, the route coherence achieved is optimal, reflecting a strategic emphasis on pursuing the global shortest path.

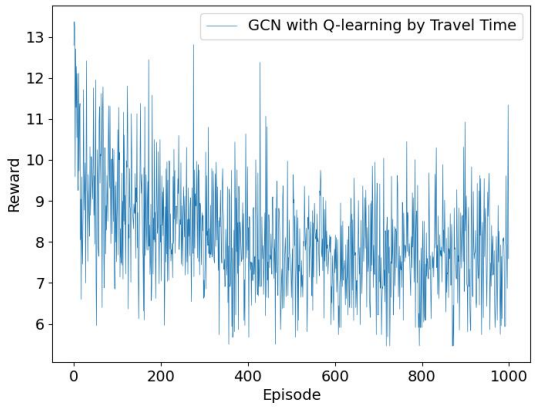
4.2.4 Analysis of Transit Time-based GCN Integrated Q-learning

This section presents a comparative analysis between the proposed method (TT based GCN integrated Q-learning) with the existing GCN integrated Q-learning under conditions of road congestion. Figure 4 illustrates the convergence of the learning curves for scenarios involving 11 to 31 stops. Notably, the conventional GCN integrated Q-learning employs distance as the reward metric, whereas the TT based GCN integrated Q-learning utilizes transit time as the reward metric.

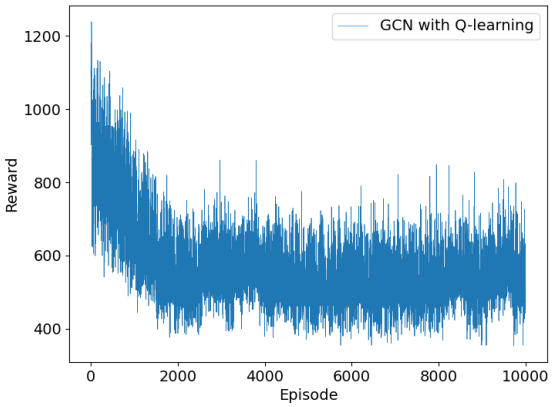
Figure 4(a) and Figure 4(b) illustrate that when the number of stops is 11, both methods achieve stable convergence at approximately 400 episodes. As the number of stops increases to 21, the GCN integrated Q-learning method attains stable convergence at around 2200 episodes (Figure 4(c)), whereas the proposed method (TT based GCN integrated Q-learning), due to its slightly higher computational complexity, reaches stable convergence at approximately 2500 episodes (Figure 4(d)). When the number of stops further increases to 31, the GCN integrated Q-learning method converges stably at about 2600 episodes (Figure 4(e)), while the TT based GCN integrated Q-learning method achieves stable convergence at roughly 3000 episodes (Figure 4(f)).



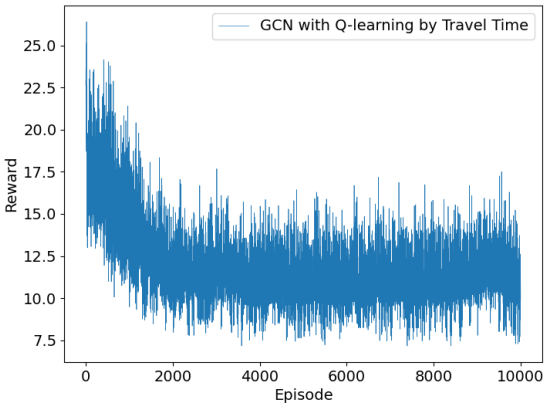
(a) 11 stops – GCN Integrated Q-learning



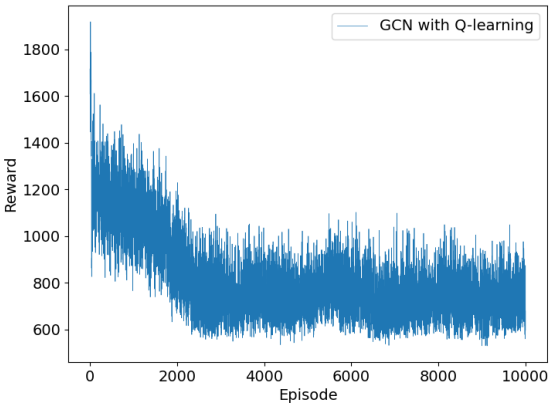
(b) 11 stops – TT based GCN Integrated Q-learning



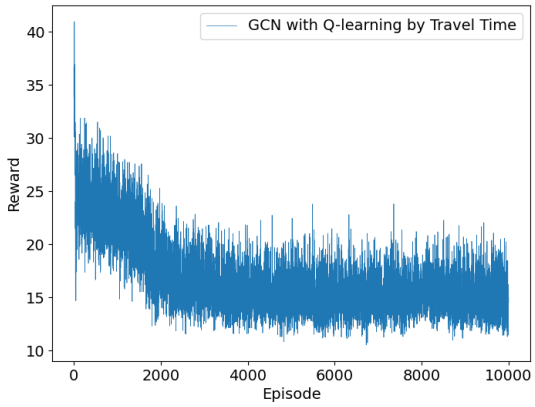
(c) 21 stops – GCN Integrated Q-learning



(d) 21 stops – TT based GCN Integrated Q-learning



(e) 31 stops – GCN Integrated Q-learning



(f) 31 stops – TT based GCN Integrated Q-learning

Figure 4. The comparison of learning curve.

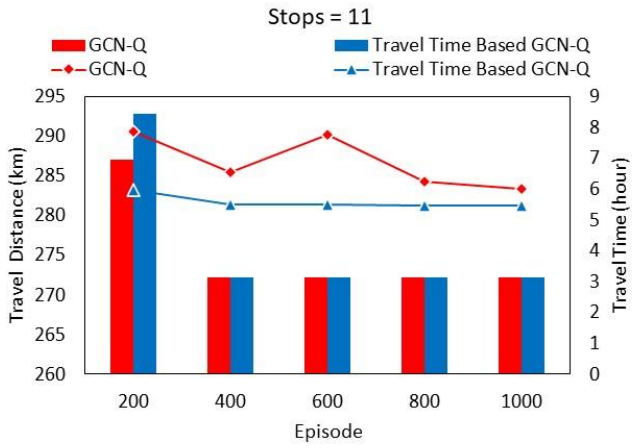
The calculation of vehicle fuel consumption follows the methodology presented by Huang Cheng [30]. This method uses the SEMTECH-D emission tester, manufactured by the American company Sensors, to measure the actual fuel consumption and emission values of heavy-duty diesel trucks during on-road operation. The test vehicle was a Dongfeng diesel truck equipped with an EQ6102 diesel engine, featuring a maximum output power of 96 kW, a curb weight of 4.8 tons, and a maximum gross weight of 9.8 tons. Fuel consumption was evaluated under both steady-speed and start-acceleration conditions on real roads. The steady-speed tests encompassed eight speed intervals ranging from 10 to 80 km/h. Acceleration tests comprised two types: normal acceleration and rapid acceleration, both conducted from 0 to 60 km/h. Normal acceleration lasted 40 seconds, with an average acceleration of 0.42 m/s², covering an estimated distance of 800 meters; rapid acceleration lasted 31 seconds, with an average acceleration of 0.54 m/s², covering an estimated distance of 600 meters. Under congested road conditions, fuel consumption was calculated at various speeds, incorporating between 0 and 38 normal accelerations

and between 0 and 15 rapid accelerations depending on the congestion level. Under smooth road conditions, fuel consumption was determined at a steady speed of 60 km/h.

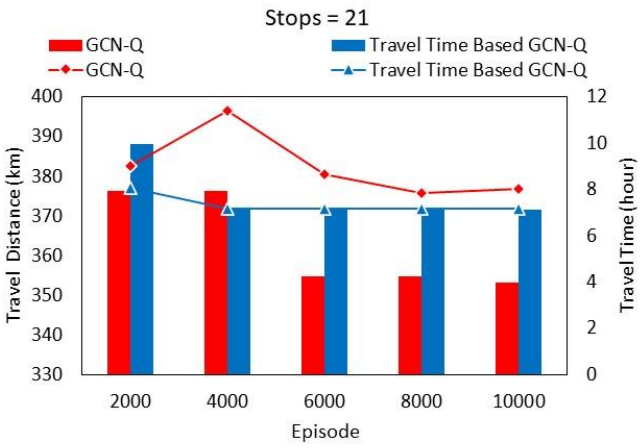
Figure 5 presents a comparison of travel distance and travel time across different scenarios. In the case of 11 stops (Figure 5(a)), after 200 episodes of iteration, the shortest path distance identified by GCN-Q is shorter than that obtained by TT-based GCN-Q. This outcome is attributed to GCN-Q's primary focus on minimizing distance; consequently, without accounting for additional factors, it consistently identifies the shortest driving route. Given the relatively small number of stops, the travel distances for both methods stabilize and converge to the shortest driving route after 400 episodes.

Due to potential road congestion, the shortest driving route does not necessarily correspond to the shortest travel time. For instance, at episode 200 in Figure 5(a), although GCN-Q identifies a route with a shorter driving distance, it requires a total travel time of 7.86 hours. In contrast, the TT-based GCN-Q, despite selecting a route with a slightly longer distance, requires only 5.96 hours. Overall, while GCN-Q consistently determines the shortest driving routes across different iteration counts, the travel time associated with TT-based GCN-Q is consistently lower than that of GCN-Q.

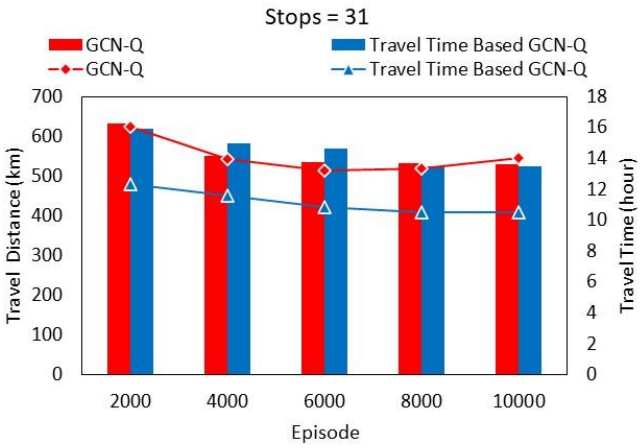
When the number of stops increases to 21 (Figure 5(b)), the shortest driving distance identified by GCN-Q is generally shorter than that found by the TT-based GCN-Q. This indicates that even after 4000 episodes of iteration and upon reaching a stable state, GCN-Q can still determine more distance-efficient routes compared to the TT-based GCN-Q. However, the travel time associated with TT-based GCN-Q remains lower than that of GCN-Q. As previously noted, when accounting for road congestion indices, the route with the shortest distance may not correspond to the shortest travel time. This phenomenon is consistently observed in the simulation results presented in Figure 5(a) through Figure 5(c), which depict scenarios with 11, 21, and 31 stops, respectively. Overall, the proposed TT-based GCN-Q method effectively identifies driving routes that minimize travel time across various conditions.



(a) 11 stops



(b) 21 stops

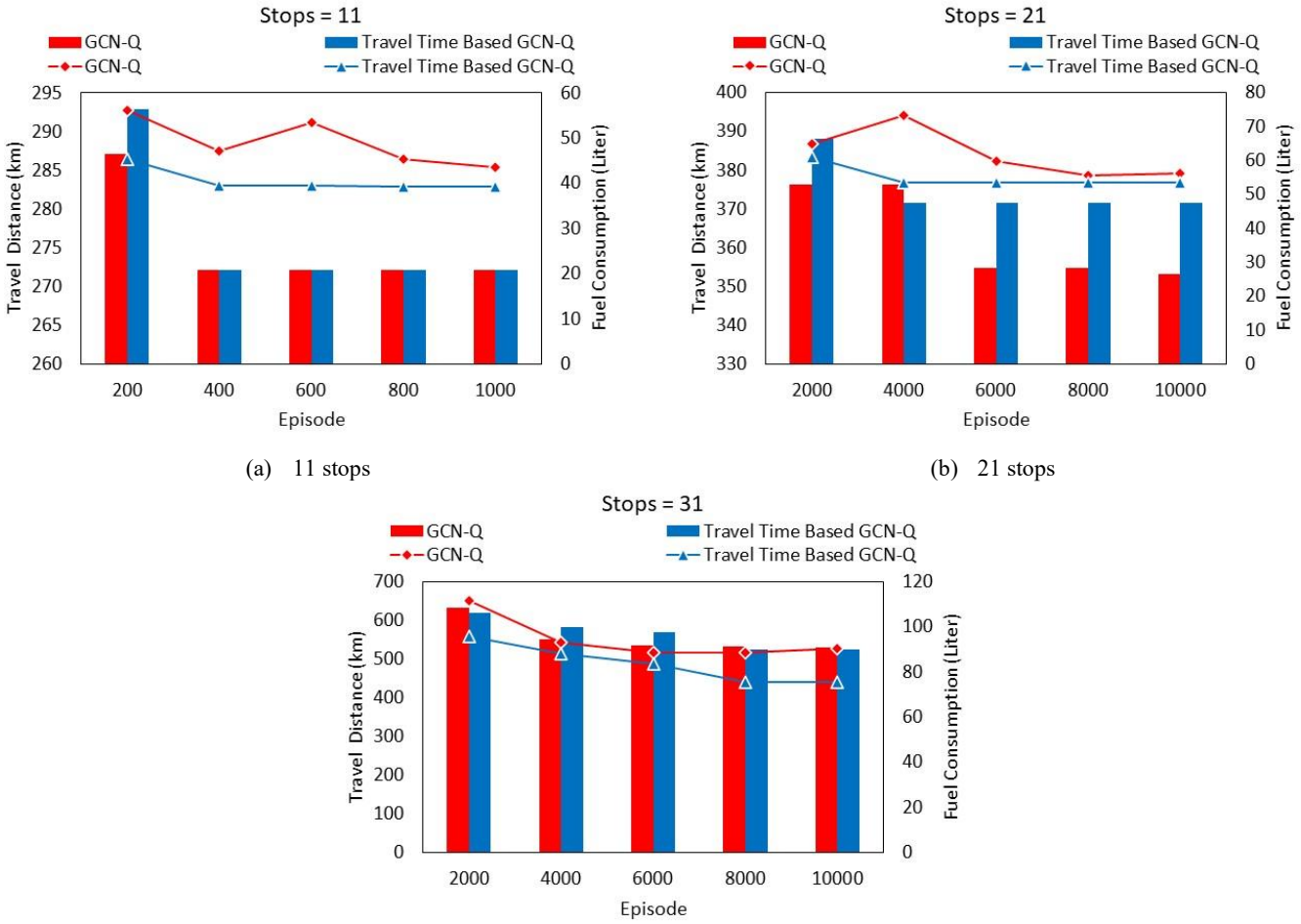


(c) 31 stops

Figure 5. The comparison of travel distance and travel time.

Generally, longer driving distances correspond to increased fuel consumption, which consequently results in higher carbon dioxide emissions. However, under real-world road network conditions, fuel consumption over the same distance may be greater in urban areas compared to suburban areas. This discrepancy arises because vehicles operating on congested city

roads experience low average speeds and frequent stop-and-go acceleration, both of which substantially elevate fuel consumption. The TT-based GCN-Q method primarily optimizes for the shortest travel time, seeking routes that minimize travel duration. According to simulation results, as shown in Figure 6(a) with stop = 11 through Figure 6 (c) with stop = 31, the travel routes found by TT-based GCN-Q under different scenarios sometimes have longer distances, but by avoiding congested road sections, their fuel consumption remains the lowest. In Figure 6(a) with stop = 11, TT-based GCN-Q found the route with the lowest fuel consumption after 400 episodes; in Figure 6(b) with stop = 21, it found the lowest fuel consumption route after 4,000 episodes; and in Figure 6(c) with stop = 31, it found the lowest fuel consumption route after 8,000 episodes.



(a) 11 stops (b) 21 stops

(c) 31 stops

Figure 6. The comparison of travel distance and fuel consumption.

The fuel consumption of the truck is directly proportional to the total volume of gas emissions. This study considers the Global Warming Potential (GWP) and employs Equation (19) to calculate the aggregate gas emissions produced by the combustion processes of transport trucks during road transportation. These emissions include carbon dioxide (CO2), methane (CH4), and nitrous oxide (N2O), which is:

$$EM_{TTW} = FC * CO_{2_{COE}} * CO_{2_{GWP}} + FC * CH_{4_{COE}} * CH_{4_{GWP}+} + FC * N_2O_{COE} * N_2O_{GWP}. \quad (19)$$

The calculation model and parameter settings of the gas emission are shown in **错误!未找到引用源。**.

Table 5. Parameters and values used in the simulation environment.

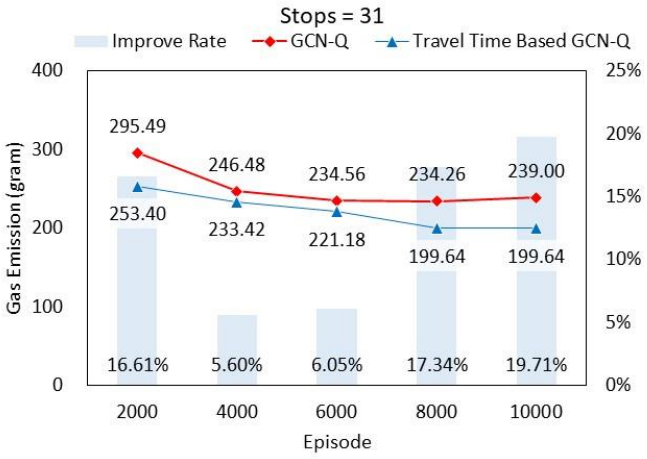
Item	Parameter
CO_2 emission factor	2.606031792($KgCO_2/L$)
CO_2 GWP*	1
CH_4 emission factor	0.000137159568 ($KgCH_4/L$)
CH_4 GWP*	25
N_2O emission factor	0.000137159568 (KgN_2O/L)
N_2O GWP*	298

* GWP is Global warming potential

Simulation results indicate that the TT-based GCN-Q method yields lower gas emissions compared to the conventional GCN-Q across various traffic scenarios, as depicted in Figure 7(a) through Figure 7(c). Although GCN-Q occasionally identifies shorter travel distances, it does not consider travel time. Consequently, selecting shorter routes that pass through congested areas may lead to increased fuel consumption and higher emissions. In Figure 7(a), at 600 episodes, GCN-Q finds the shortest route; however, congestion results in increased fuel consumption and gas emissions, as depicted in the same figure. Notably, carbon dioxide constitutes over 98% of the total gas emissions. Similarly, Figure 7(b) demonstrates that GCN-Q exhibits a comparable pattern at 4000 episodes. As the number of stops increases, the benefits of the TT-based GCN-Q become more evident. Specifically, when the number of stops is 11, as shown in Figure 7(a), the TT-based GCN-Q reduces gas emissions by 11.1% to 35.68% relative to GCN-Q. When the number of stops reaches 21, as illustrated in Figure 7(b), the TT-based GCN-Q achieves a reduction in gas emissions ranging from 3.82% to 37.06% compared to GCN-Q. Furthermore, with 31 stops, as presented in Figure 7(c), the TT-based GCN-Q identifies the optimal driving route after 8000 episodes, decreasing total gas emissions to 199.64 grams and yielding an overall improvement rate between 5.6% and 19.17%.

(a) 11 stops

(b) 21 stops



(c) 31 stops

Figure 7. The Gas Emission Comparison.

5. CONCLUSION

The VRP is one of the most extensively studied and widely applied combinatorial optimization problems within the domains of operations research and optimization. It is prevalent across numerous real-world applications and holds considerable practical significance, particularly in critical sectors such as logistics distribution, urban transportation, express delivery, sanitation scheduling, and emergency response. As a representative NP-hard problem, the VRP has been the subject of research for several decades, with both academic and industrial communities proposing a variety of effective solution approaches. These approaches are generally classified into three categories: exact algorithms, heuristic algorithms, and machine learning-based optimization algorithms.

Exact algorithms, including branch and bound and integer linear programming, are capable of identifying globally optimal solutions for small-scale problems. However, as the problem size increases, the computational time escalates exponentially, thereby impeding the ability to satisfy real-time and responsiveness requirements in practical applications. Heuristic and metaheuristic algorithms, such as genetic algorithms, ant colony optimization, and simulated annealing, provide effective search capabilities and broad applicability; nevertheless, their performance is highly contingent upon manually designed rules and parameter configurations, which limits their generalizability and transferability. Furthermore, most reinforcement learning models applied to routing tasks tend to overlook the graph structural information among stops, often treating the problem as a sequence prediction task. This approach results in notable deficiencies in structural modeling, global perception, and generalization capacity, thereby hindering the full exploitation of adjacency and path structure features inherent in the graph.

To address these challenges, the present study proposes a hybrid model that integrates a Graph Convolutional Network (GCN) with a Q-learning algorithm, incorporating a road traffic congestion index to capture real-time traffic conditions. The GCN functions as the state encoding module, effectively extracting neighborhood information of stops, edge weights (e.g., distance parameters), and the connectivity among stops. This process embeds the graph's structural features into the stop state representations, thereby enhancing the model's structural awareness. Q-learning, serving as the optimization module within reinforcement learning, incrementally learns the optimal sequence for visiting stops based on the state-value function update strategy. This integrated approach overcomes the limitations of traditional methods that rely exclusively on inherent stop attributes (such as coordinates), enabling a comprehensive understanding and dynamic construction of the entire graph structure.

The simulation results indicate that the integrated GCN and Q-learning model surpasses traditional greedy algorithms and standalone Q-learning approaches with respect to path planning quality, total cost management, and model stability. Importantly, the inclusion of edge features (distance) and the adjacency matrix enhance the model's generalization capabilities and robustness across diverse graph structures, rendering it particularly suitable for large-scale and structurally complex path planning problems.

REFERENCES

[1] Yang Shuguo and Li Chunxia. Research on an Optimization Model for a Class of Whole Vehicle Logistics Problems. *Mathematical Practice and Recognition*, vol. 48, no. 12, pp. 11-19, 2018.

[2] Yu Xinyao, Zhu Ning, Ma Yanming, et al. Research on Bus Vehicle Scheduling Plan Considering Abnormal Train Numbers. *Systems Engineering Theory and Practice*, pp. 1-18, 2022.

[3] Xu Junxiang, Zhang Jin, and Guo Jingni. Research on the Dispatching Problem of Autonomous Vehicles Considering Dynamic Travel Time. *Industrial Engineering and Management*, vol. 24, no. 05, pp. 120-126, 2019.

[4] Xu Lixia, Xu Qi, Fan Dandan. Research on Vehicle Logistics Transportation Planning Based on Two Models. *Mathematics in Practice and Theory*, vol. 45, no. 22, pp. 213-220, 2015.

[5] Chi Jushang, He Shiwei, Song Zilong. A Two-Level Car Carrier Transportation Problem Based on Branch-and-Price Algorithm. *Control and Decision*, vol. 37, no. 01, pp. 185-195, 2022.

[6] Kucukoglu I , Dewil R , Cattrysse D. The electric vehicle routing problem and its variations: A literature review. *Computers and Industrial Engineering*, vol. 161, 2021.

[7] Shuai Zhang, Mingzhou Chen, Wenyu Zhang, and Xiaoyu Zhuang. uzzy optimization model for electric vehicle routing problem with time windows and recharging stations. *Expert Systems with Applications*, vol. 145, May 2020.

[8] National Postal Administration Development Research Center. Development Report on China's Express Delivery and Logistics Industry, 2022.

[9] Toth P. and Vigo D. The vehicle routing problem. *Philadelphia: Society for Industrial and Applied Mathematics*, pp. 1-26, 2002.

[10] B. J. Li, G. H. Wu, Y. M. He, M. F. Fan, and W. Pedrycz. An overview and experimental study of learning-based optimization algorithms for the vehicle routing problem. IEEE/CAA J. Autom. Sinica, vol. 9, no. 7, pp. 1115–1138, Jul. 2022. doi: 10.1109/JAS.2022.105677.

[11] Song, H., Triguero, I. and Özcan, E. A review on the self and dual interactions between machine learning and optimization. *Prog Artif Intell* 8, 143–165 (2019). https://doi.org/10.1007/s13748-019-00185-z

[12] Mitchell T M. Machine learning. New York: Mc Graw-hill, 2007.

[13] Garg V, Jegelka S, Jaakkola T. Generalization and representational limits of graph neural networks. *Proceedings of the International Conference on Machine Learning*. PMLR, pp. 3419-3430, 2020.

[14] Yan J, Yang S, Hancock E R. Learning for graph matching and related combinatorial optimization problems. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pp. 4988-4996, 2020.

[15] Gambardella L. M., Dorigo M. Ant-Q: A reinforcement learning approach to the traveling salesman problem. *Machine Learning Proceedings*, pp. 252–260, 1995.

[16] J Zhou, et al. Graph neural networks: A review of methods and applications. *AI Open*, vol. 1, pp. 57-81, , 2020.

[17] Liu F., Zeng G. Study of genetic algorithm with reinforcement learning to solve the TSP. *Expert Systems with Applications*, vol. 36, no. 3, pp. 6995-7001, 2009.

[18] Mlejnek J and Kubalik J. Evolutionary hyper heuristic for capacitated vehicle routing problem. *Proceedings of the 15th annual conference companion on Genetic and evolutionary computation*. pp. 219-220, 2013.

[19] K. Dorling, J. Heinrichs, G. G. Messier and S. Magierowski. Vehicle Routing Problems for Drone Delivery. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 47, no. 1, pp. 70-85, Jan. 2017, doi: 10.1109/TSMC.2016.2582745.

[20] G. B. Dantzig and J. H. Ramser. The Truck Dispatching Problem. *Management science*, vol. 6, no.1, pp. 80-91, Oct. 1959.

[21] Marshall Fisher. Vehicle Routing. Handbooks in OR & MS, Vol. 8. 1995.

[22] Laporte G, Mercure H, Nobert Y. An exact algorithm for the asymmetrical capacitated vehicle routing problem. *Networks*, vol. 16, no. 1, pp. 33-46, 1986.

[23] Eilon S, Watson-Gandy C. D. T., Christofides N, et al. Distribution management- mathematical modelling and practical analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 6, pp. 589-589, 1974.

[24] Rao M R, Zionts S. Allocation of transportation units to alternative trips—A column generation scheme with out-of-kilter subproblems. *Operations Research*, vol. 16, no. 1, pp. 52-63, 1968.

[25] Cordeau J F, Laporte G. Tabu search heuristics for the vehicle routing problem. *Metaheuristic Optimization via Memory and Evolution*, US, 2005.

[26] Bishop C. M., Nasrabadi N. M. Pattern recognition and machine learning. New York: Springer-Verlag Berlin, Heidelberg, 2006.

[27] Kaelbling L. P., Littman M. L., Moore A. W. Reinforcement learning: A survey. *Journal of artificial intelligence research*, vol. 4, pp. 237-285, 1996.

[28] LI B., WU G., HE Y., et al. An overview and experimental study of learning-based optimization algorithms for the vehicle routing problem. *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 7, pp.1115-1138, 2022.

[29] Balaji B, Bell-Masterson J, Bilgin E, et al. Reinforcement Learning Benchmarks for Online Stochastic Optimization Problems. *arXiv*, 2019.

[30] Huang Cheng, Chen Changhong, Jing Qiqu, et al. n-board em ission from heavy-duty diesel vehicle and its relationship with driving behavior. *Acta Scientiae Circumstantiae*, vol. 27, no. 2. pp. 177-184, Feb. 2007.