Industrial Engineering and Innovation Management (2025)
Clausius Scientific Press, Canada

DOI: 10.23977/ieim.2025.080217
ISSN 2522-6924 Vol. 8 Num. 2

Enhancing Urban E-grocery Delivery Efficiency: Adaptive Reinforcement Learning for Dynamic Vehicle Routing

Liqiang Wu

Department of Industry Engineering, University of Shanghai for Science and Technology, Shanghai, 200093, China
232481204@st.usst.edu.cn

Keywords: Dynamic Vehicle Routing Problem; Reinforcement Learning; Stochastic Optimization; Urban E-grocery Deliveries

Abstract: Managing real-time road conditions and satisfying stochastic customer demands pose significant challenges for optimizing the Dynamic Vehicle Routing Problem with Stochastic Requests (DVRPSR) in urban grocery delivery settings. Most existing approaches generate solutions offline as static plans, which are only applicable to the specific scenarios they were optimized for, making it difficult to efficiently plan and operate a dynamic urban grocery delivery system for last-mile delivery. In this study, we introduce a new dynamic optimization model for DVRPSR. Our approach combines a multi-attention mechanism with reinforcement learning and incorporates a customer point update strategy to enhance efficiency in urban E-grocery delivery. To validate the effectiveness of our method, we conducted experiments across small (50 customers and 5 vehicles), medium (100 customers and 10 vehicles), and large (200 customers and 20 vehicles) data scales. The results demonstrate that our method outperforms current routing methods, reducing total path length, improving customer service coverage, and maintaining efficient computation time. This provides a promising strategy for enhancing the efficiency of urban E-grocery delivery and reducing operational costs.

1. Introduction

With the dramatic growth of urban E-grocery delivery during the pandemic, the competitive battleground has shifted to focus on speed and user experience, particularly for immediate and unplanned needs. The main challenge in urban grocery delivery is meeting dynamic and unpredictable customer demands while maintaining timely and efficient service. The Dynamic Vehicle Routing Problem with Stochastic Requests (DVRPSR) is a critical issue in this sector, significantly impacting both efficiency and customer satisfaction.

The conventional Vehicle Routing Problem (VRP) faces difficulties in dynamic situations due to rising unpredictability and constantly changing client needs. While most previous studies on DVRPSR concentrate on real-time routing changes, they often rely on stationary assumptions and do not adequately address the intrinsic complexity and unpredictability of E-grocery delivery [1, 2]. In

practical urban E-grocery routing, some requests (static requests) are known in advance, while others (dynamic requests) appear randomly during service. Therefore, we need a model capable of quickly reacting and adjusting to order changes. Unfortunately, many of the current models fail to adequately depict the dynamic and varied character of real-world situations [3-5]. Most existing approaches for DVRPSR still depend on heuristics that define operators handy to guide the random exploration of the solution space, which hindering efficient online decision-making.

This research aims to develop new model of DVRPSR to assign an initial route plan to delivery vehicles and dynamically modify the routes depending on developing needs simultaneously, to satisfy as many client requests as possible and thereby reduce the delivery cost. Reinforcement Learning (RL) offers new insight to handle the dynamic flexibility needed in current routing difficulties [6-8]. However, these methods often face challenges with scalability and computational efficiency, especially when dealing with complex and large-scale problems. Consequently, integrating RL with advanced techniques, such as deep neural networks become essential to improve performance and applicability in real-world scenarios. Notably, the combination of multi-head attention mechanisms from natural language processing with reinforcement learning is particularly promising[9]. This approach enhances feature identification and dynamic route optimization by providing insights into the interdependencies and relationships between various input features.

Consequently, in this work we aim to develop a new dynamic vehicle routing optimization based on the Markov Decision Processes (MDPs), which see the ideal solution as a succession of strategic choices. This allows us to use reinforcement learning to find near-optimal solutions by increasing the probability of decoding the "ideal" sequence. Furthermore, given that dynamic features cannot be adequately extracted for optimal dynamic path optimization using reinforcement learning alone, the introduction of the multi-attention mechanism allows our model to effectively explore the dependencies and relationships between input features, thus lead to the main following contributions of this paper:

- We propose a new model combining a multi-attention mechanism with reinforcement learning (MA-RL) to address the DVRPSR. This model optimizes both offline and online decisions for scheduling dynamic requests, providing a comprehensive solution for initial route planning and real-time adjustments.
- We employ a real-time update strategy for customer requests in sovling the DVRPSR, enabling dynamic adjustments to the delivery route. By continually adapting to fluctuating demand, this strategy ensures optimal routing and efficient delivery, which are essential for maintaining service quality and operational agility in fast-paced market conditions.
- The model's performance is rigorously tested through artificial benchmarks, coving a range of scenarios from small to large-scale. This thorough testing demonstrates the model's adaptability and consistent effectiveness, confirming its capability to handle diverse operational scales and complex scenarios. The results validate the model's utility and robustness in real-world applications.

2. Literature Review

VRP has long been a fundamental concern in logistics and transportation research, focusing on decreasing travel times and costs while satisfying customer needs. Historically, vehicle routing problems have been approached with fixed assumptions, starting with the innovative method proposed by Dantzig and Ramser in 1959 and evolving through modifications such as the Time-Windowed VRP (VRPTW) and the Capacitated VRP (CVRP). These presumptions are predicated on the belief that, during route planning, key variables such as customer addresses, demand levels, and traffic conditions are static and unchangeable [10-12]. However, traditional models show major flaws in terms of adaptation to changing conditions and uncertainty. This is particularly true for E-grocery

delivery, where consumer demand can fluctuate wildly and traffic patterns can change quickly. To address these challenges, recent research focuses on solving the Dynamic Vehicle Routing Problem (DVRP), which is designed to manage the dynamic nature of e-grocery delivery and align static scheduling with the needs of real-time delivery. DVRP allows for instantaneous route adjustment to manage on-demand order integration, moving beyond predefined planning assumptions. Grocery stores must be able to react fast for quick changes and fine-tune delivery routes to satisfy consumer needs. To address this real-world challenge, DVRP was introduced in the late 1970s [13,14]. Psaraftis (1988) investigate the quick route adjustments in response to new data, urgent needs, or vital events [15]. Since then, the extensive body of literature on DVRP justifies the numerous reviews dedicated to this problem [16-18]. Recently, DVRPSR have garnered growing interest [19]. Many methods within the Multiple Scenario Approach (MSA) framework still depend on tools and heuristics designed for static variants, creating sets of anticipatory plans that are updated or discarded in response to dynamic events. However, current DVRPSR models remain inadequate for managing the complexity of frequent order changes in E-grocery deliveries [20,21].

Pointer Networks paved the way for generalized solutions to combinatorial problems using Recurrent Neural Networks (RNNs) [22]. Reinforcement Learning (RL), which primarily focuses on rewards and penalties, is a strong technique in machine learning that enhances decision-making by examining the results of actions. RL is particularly effective in adapting to changing conditions and addressing the complexity of VRP [23]. By modeling and optimizing route planning strategies via experimentation and iterative improvements, RL can adapt to rapidly changing delivery needs and fluctuating road conditions [24,25]. These research studies leverage the generalization capability of artificial intelligence to develop vehicle routes with satisfactory performance. However, few studies have attempted to employ machine learning-based methods to solve VRPs with stochastic demand. Tackling this type of problem requires achieving high-quality solutions with acceptable computational effort.

Recent advances in deep learning's multi-head attention mechanisms have significantly increased data processing efficiency. Proposed by Vaswani and his colleagues for the transformer model architecture [26],this design allows the simultaneous processing of multiple data segments. This enables the model to attend multiple points along the input stream concurrently. This parallel processing capability greatly enhances the model’s ability to compile comprehensive data from multiple angles [27].Therefore, Multi-head attention approaches can be highly beneficial in the field of E-grocery delivery by fully analyzing and understanding the intricate spatial and temporal relationships between consumer locations and their needs. These techniques can simplify real delivery routes, hence improving delivery performance and customer service quality.

3. Methodology

3.1 Problem Definition

In the realm of E-grocery delivery, a primary difficulty in addressing the DVRPSR lies in adeptly handling fluctuating demands in real-time while maintaining efficiency and cost-effectiveness. In this study, we define the DVRPSR as follows: Devlivey vehicles pick up groceries from a distribution center and travel to a depot. Deliveries start and end from a depot to address customer demands that arise unpredictably over a specified timeframe (a half-day). These requests can be categorized into two types: (i) deterministic requests known before leaving the depot; and (ii) random requests that occur with a certain probability after leaving the depot. The location of each customer is known, and it is assumed that all requests must be accepted and serviced by vehicles.

3.2 Formulation of the Model

The network of DVRPSR is comprised of a set of nodes N, a set of arcs A, and a set of vehicles K. The set of nodes N is further divided into two subsets: depot (O) and customers (F), denoted as $N = O \cup F$. The demand of each customer $f \in F$ is defined by the required weight of goods u_g (kg). The probability of sending a request P_f follows a binomial delivery with K trials, and the probability of a new request occurring in period $k \in K$ is calculated as $P_{fk} = 1 - \sqrt[k]{1 - P_f}$. The maximum payload of a vehicle is U (kg), and the empty vehicle (curb) weight is W (kg).

The specific model parameters and variables are shown in **Table 1**.

TABLE 1 Notations of variables and parameters

Sets	
O	Set of depot
F	Set of customers, $F = \{F_n \cup F_c \cup F_h \cup F_{h-1} \cup F_a \cup F_u\}$
F_n	Set of new customers after dynamic update
F_c	Set of initial customers
F_h	Set of having been serviced customers when dynamically updated
F_{h-1}	Set of having just been serviced customers during this dynamic update
F_a	Set of having not been serviced customers after dynamically updated
F_u	Set of customers that are being serviced
N	Node set consist of $\{O \cup F\}$
K	Set of Vehicles visiting the $F, K = \{K_i \cup K_n \cup K_h \cup K_a \cup K_w \cup K_o\}$
K_i	Set of initial Vehicles visiting the F
K_n	Set of newly dispatched Vehicles visiting the F after the dynamic update
K_h	Set of Vehicles had completed delivery at the time of the dynamic update
K_a	Set of Vehicles visiting the F in delivery after dynamic update
K_w	Set of Vehicles visiting the Fthat are not at the customer's point at the time of the dynamic update
K_o	Set of Vehicles visiting the F that are at the customer's point at the time of the dynamic update
Deterministic parameters	
fc	Fixed cost of dispatch
wc	Cost of waiting for early arrival
pc	Penalty costs for late arrival of vehicles
c	Consumption cost per unit mile traveled by Vehicles
d_f	Demand of customer f, $f \in F$
d_{ij}	Distance needed to travel on arc (i, j)
d_{io}	Distance from customer point to depot
Q^{van}	Capacity of Vehicles
Q_k^{van}	Remaining capacity of Vehicle k responding to dynamic update, $k \in K$
v	Speed of Vehicles
ϖ	Service stop time of the depot
Stochastic parameters	
$\rho_{f_i f_j}$	Time needed to travel from customer f_i to customer f_j , $f_i, f_j \in F$
S_{fk}	Vehicle Vehicle k arrives at customer f service time, $f \in F, k \in K$
T_{max}	the maximum allowable time for a Vehicle's route
Auxiliary variables	
T_1	Point of commencement of delivery
T_2	Point of acceptance of dynamic requirements
T_{fk}	Vehicle k starts service at the customer $f, f \in F, k \in K$
t_0	Intervals for dynamic updating of requirements
LT_f	Latest time for client f to receive services, $f \in F$
ET_f	Earliest time client f receives service, $f \in F$
a_f	The moment the Vehicle leaves the customer f, $f \in F$
t_{fk}	The moment the Vehicle k arrives at the customer f, $f \in F, k \in K$
T_{fk}^{van}	Visit time at node f by vehicle k, $f \in F$
Decision variables	
y_l	$\begin{cases} 1, \text{ if Decision} - \text{making on the dispatch of new vehicles} \\ 0, \text{otherwise} \end{cases}$
y_{lw}	$\begin{cases} 1, \text{if Decision} - \text{making vehicles on their way to delivery} \\ 0, \text{otherwise} \end{cases}$
y_{lo}	$\begin{cases} 1, \text{if Decision} - \text{making vehicles at the point of customer} \\ 0, \text{otherwise} \end{cases}$
X_{fo}	$\begin{cases} 1, \text{if customer } f \textit{ back to depot} \text{ served by } van\ k, k \in K, f \in F \\ 0, \text{otherwise} \end{cases}$
$y_{f_i f_j k}$	$\begin{cases} 1, if\ van\ k \text{ serve from } customer\ f_i\ to\ customer\ f_j\ ,\ f_i, f_j \in F, k \in K \\ 0, otherwise \end{cases}$
$y_{o f_j k}$	$\begin{cases} 1, if\ van\ \ k \text{ serve from } depot\ o\ to\ customer\ f_j\ ,\ f_i, f_j \in F, k \in K \\ 0, otherwise \end{cases}$
y_{fk}	$\begin{cases} 1, if\ \ customer\ f\ is \text{ served by Vehicle k } , f \in F, k \in K \\ 0, otherwise \end{cases}$

The model establishes two objective functions: minimizing total cost and maximizing customer coverage.

The total cost includes the Vehicle's operating cost, the Vehicle's initial start-up cost, and any penalty costs incurred during delivery. The operating cost P_1 encompasses several components: the travel cost between depot and the first customer, the travel cost between customer points, the travel cost between customer points to the depot, and the travel cost of returning Vehicles to the warehouse for picking up goods. This can be expressed as:

$$P_1 = \sum_{f_i \in F_a} \sum_{f_j \in F_a} \sum_{k \in K_n} y_{f_i f_j k} \cdot c \cdot d_{ij} \cdot y_l + \sum_{f_i \in F_a} \sum_{f_j \in F_a} \sum_{k \in K - K_h} y_{f_i f_j k} \cdot c \cdot d_{ij} \cdot y_{lw} \cdot \sum_{f_i \in F_{h-1}} \sum_{f_j \in F_a} \sum_{k \in K - K_h} y_{f_i f_j k} \cdot c \cdot d_{ij} \cdot y_{lo} + \sum_{f_i \in F - F_h} \sum_{k \in K - K_h} X_{fo} \cdot c \cdot d_{io} \quad (1)$$

The initial start-up cost P_2, which includes the cost of deploying newly issued vehicles, can be expressed as:

$$P_2 = \sum_{f_j \in F_a} \sum_{k \in K_n} fc \cdot y_{of_j k} \cdot y_l \quad (2)$$

The penalty cost P_3 incurred during the Vehicle delivery process can be expressed as:

$$P_3 = wc \sum_{f \in F_a} \sum_{k \in K_a} max[0, ET_f - t_{fk}] + pc \sum_{f \in F_a} \sum_{k \in K_a} max[0, t_f - LT_f] \quad (3)$$

Then the corresponding objective function of the model is expressed as:

$$min\ P = P_1 + P_2 + P_3 \quad (4)$$

$$max \sum_{f \in F} \sum_{k \in K} y_{fk} \quad (5)$$

Flow Conservation Constraints:

Constraints (6), (7)and (8) ensure that each customer only served by one vehicle. Restriction (9) indicates that each vehicle only has a single delivery route, and each vehicle departs from the depot and returns to the depot after the delivery is completed. Restrictions (10) represents branch elimination constraint, where S indicates the collection of all the latest customer points on the whole service route after the vehicle delivery route is updated.

$$\sum_{f_i \in F_a} \sum_{k \in K_n} y_{f_i f_j k} = 1, \forall f_j \in F_a \quad (6)$$

$$\sum_{f_j \in F_a} \sum_{k \in K_n} y_{f_i f_j k} = 1, \forall f_i \in F_a \quad (7)$$

$$\sum_{f_i \in F_a} \sum_{k \in K_n} y_{f_i f_j k} \cdot y_{lw} \leq 1, \forall f_j \in F_a \quad (8)$$

$$\sum_{f_j \in F_a} y_{of_j k} = \sum_{f_j \in F_a} y_{f_j 0 k}, \forall k \in K \quad (9)$$

$$\sum_{f_i \in F_a} \sum_{f_j \in F_a} y_{f_i f_j k} \leq |S| - 1, S \in F_a, \forall k \in K \quad (10)$$

Time Constraints:

Constraint (11), (12)and (13) ensure the time window constraint of the vehicle during the delivery service. Constraint (14) Ensure that the route is feasible with respect to the vehicle's operational constraints, like maximum route length or duration.

$$T_{fk} + S_f = a_f, \forall f \in F, \ \forall k \in K \quad (11)$$

$$(T_{f_i k} + t_{f_i f_j k} + S_{f_i k}) \cdot y_{f_i f_j k} \leq T_{f_j k} \cdot y_{f_i f_j k}, \forall f_i, \forall f_j \in F_a, \forall k \in K \quad (12)$$

$$ET_f \leq T_{fk} \leq LT_f, \forall f \in F, \forall k \in K \quad (13)$$

$$\sum_{f_i \in F} \sum_{f_j \in F} \rho_{f_i f_j} \cdot y_{f_i f_j k} \leq T_{max}, \forall k \in K \tag{14}$$

Capacity Constraints:

Constraint (15) Indicates that the demand of all customers installed in each vehicle does not exceed the maximum load capacity of the vehicle. Constraint (16) indicates that at the time of the update, the remaining load of the Vehicle that has not completed the delivery task should meet the needs of the remaining customer points for the service they require after the update.

$$\sum_{f_i \in F_a} \sum_{f_j \in F_a} \sum_{k \in K_n} y_{f_i f_j k} \cdot d_f \leq Q^{van}, \forall k \in K \tag{15}$$

$$\sum_{f_i \in F_a} \sum_{f_j \in F_a} \sum_{k \in K_h} y_{f_i f_j k} \cdot d_f \leq \sum_{k \in K - K_h} Q_k^{van}, \forall k \in K \tag{16}$$

4. Solution Framework

To effectively address the DVRPSR, this paper introduces a new model, MA-RL, designed to intelligently manage the complex demands of E-grocery delivery systems. The model uses an embedding layer in a multi-dimensional vector space to uniformly process features such as customer locations, time windows, and demand volumes, thereby improving data processing efficiency and consistency. The core multi-head attention mechanism enhances the model's ability to capture dynamic changes and spatio-temporal data heterogeneity by simultaneously focusing on multiple aspects of the input data. Additionally, a Long Short-Term Memory network (LSTM) process time series data, effectively capturing historical information to support real-time decision-making. The output layer employs a fully connected layer that uses a Softmax function to output the probability delivery of the next path selection, directly supporting real-time delivery demands. Since the order of customer locations and demands does not affect the final decision, we optimize input processing by omitting the encoder RNN and directly using embedded vectors to represent inputs, simplifying the model structure and reducing computational complexity. This design enables the model to efficiently handle high-frequency order updates in E-grocery delivery, enhancing operational flexibility and system response speed. The proposed MA-RL is shown in **Figure 1**.

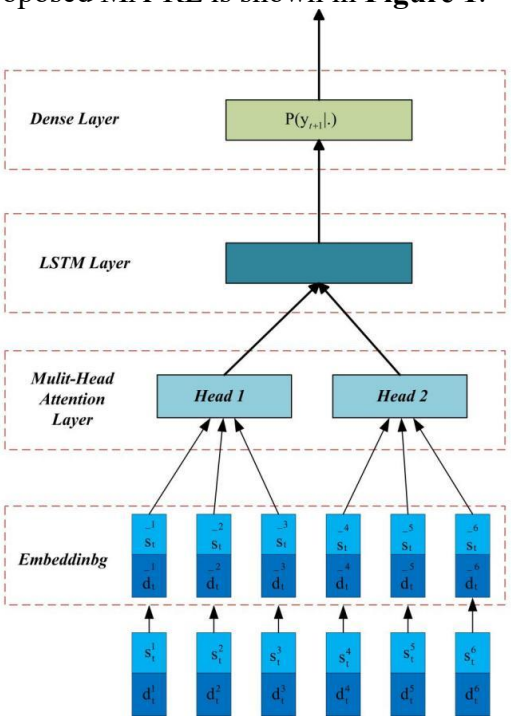

Figure 1 The proposed MA-RL.

The embedding layer maps inputs into a high-dimensional vector space. The RNN decoder stores information about the decoded sequence. Then, the RNN hidden states and embedded inputs use the attention mechanism to process time series data through LSTM, generating hidden state representations and producing probabilities for the next input.

4.1 Encode & Decode

The state of the delivery system at any time k is defined by $x_i^t = (s^i, d_t^i)$, where s^i represents the static elements of the input (location coordinates) and d_t^i represents the dynamic elements of the input (changes in customer demand). The state transition is stochastic, with the randomness of new requests resulting in an unknown probability delivery. The reward is also stochastic, but we assume its probability delivery is known so that the optimization model can make decisions based on potential costs and customer service coverage.

In this setup, the primary optimization goal is to minimize the total cost, including transportation costs, service delay costs, and potential penalty costs, while maximizing the fulfillment rate of customer requests. This requires the model to flexibly adapt to changing demands and efficiently adjust route planning to respond to immediate orders. Additionally, considering the characteristics of the E-grocery deliveries, the model also needs to handle high-frequency order peaks, ensuring efficient delivery service even during peak order periods.

In exploring the DVRPSR, we encounter a highly complex decision-making environment. The framework proposed in this study adopts an innovative approach, combining multi-head attention mechanisms and a RNN decoder to optimize vehicle route selection under dynamic conditions. This model is particularly suitable for E-grocery deliveries, where customer demands and delivery conditions may constantly change.

For this framework, we define a general combinatorial optimization problem with an input set $X = \{x^i, i = 1,\cdots, N\}$. We allow some elements of each input to change between decoding steps, which is indeed the case in VRP. Dynamic elements may be the product of the decoding process itself, or they can be imposed by the environment. In this DVRPSR, as vehicles visit customer nodes, the remaining customer demand changes over time; or we may consider a variation where new customers arrive or adjust their demand values over time, independent of vehicle decisions. Formally, we represent this with a series of tuples $x_i^t = \{(s^i, d_t^i), t = 0,1,\cdots N\}$, where s^i and d_t^i are the static and dynamic elements of the input, respectively. Static elements s^i may include customer location coordinates, while dynamic elements d_t^i such as customer demand vary over time and environmental conditions. The vector x_i^t can be viewed as a feature vector describing the state of input i at time t. For example, in VRP, x_i^t provides a snapshot of customer i, where s^i corresponds to the 2D coordinates of the customer's location, and d_t^i represents their demand at time t. We denote the set of all input states at a fixed time t as X_t.

The decoding process starts at the initial state X_0, using a pointer y_0 to indicate the initial input. At each decoding time point $t(t = 0,1,\cdots N)$, the pointer y_{t+1} selects one input from X_t as the decision input for the next step. This selection process reflects the real-time decision needs of the vehicle, such as choosing the next customer to visit or returning to the depot. The entire process continues until specific termination conditions are met, such as all customer demands being fulfilled. This process generates a sequence $Y = \{y_t, t = 0,1,\cdots N\}\}$ of possible length T, reflecting the dynamic routing solution to the problem. The sequence length may differ from the input length M because, for example, vehicles might need to return to the depot multiple times to refill. We also use the notation Y_t to represent the decoding sequence up to time t, i.e., $Y_t = y_0,\cdots y_t$.

Our goal is to develop a stochastic policy π that generates the sequence Y in a way that minimizes

the loss objective while satisfying the problem constraints. The loss function is defined based on a negative number of expected rewards, which can be expressed as:

$$L(\theta) = -\mathbb{E}_{\pi_\theta}[R_{(\tau)}] \tag{17}$$

Among them, π_θ is a strategy, parameterized by a neural network, τ stands for trajectory (a series of states and actions), $R_{(\tau)}$ is the reward for that trajectory, which can be expressed as:

$$R_{(\tau)} = \alpha \cdot (-TotalCost(\tau)) + \beta \cdot \text{NumCustomersServed}(\tau) \tag{18}$$

$TotalCost(\tau)$ is the total cost of the path τ ,$NumCustomersServed(\tau)$ Is the number of customers served in the path τ, α and β are weighting factors that adjust for the relative importance of cost and number of customers served in the reward function.

The optimal policy π^* will generate the optimal solution sequence with probability 1. Our aim is to make π as close to π^* as possible. We use the probability chain rule to decompose the probability of generating the sequence Y, denoted as $P(Y|\ X_0\ )$ as follows:

$$P(Y|\ X_0\ ) = \prod_{t=0}^{T} P(\ y_{t+1}|\ Y_t, X_t\ ) \tag{19}$$

The state transition function describes the system state update based on the current decision and the previous state.

$$X_{t+1} = \ f(\ y_{t+1,}\ X_t) \tag{20}$$

The attention mechanism calculates the probability at each step through the function g, in the form of

$$P(\ y_{t+1}|\ Y_t, X_t) = softmax\big(g(\ h_t, X_t)\big) \tag{21}$$

where g is an affine function outputting a vector of the input size, and h_t is the state of the RNN decoder, summarizing the information from the previous decoding steps $y_0, \cdots\ y_t$.

4.2 Training Method

To train the network, we use the well-known strategy gradient methods. To use these methods, we parameterize the stochastic policy π with the parameter θ. The policy gradient method iteratively improves the policy using a gradient estimate of the expected returns associated with the policy parameters. In principle, the policy gradient model consists of two networks: (i) an actor network that predicts the probability delivery of the next action in any given decision step, and (ii) a critic network that estimates the reward of any problem instance from a given state.

4.3 Update Strategy

For the generation of dynamic demands, the main idea is to transform the dynamic problem into multiple static problems for solution. We adopt a customer point update strategy. In this strategy, when any vehicle arrives at a customer point and completes the service, the system updates the dynamic information upon leaving the customer point and determines if new customer demands have arisen. If new demands arise, the system first checks if the remaining capacity of the vehicle can satisfy the new customer point's demands. If it can, the vehicle's delivery route is re-planned; otherwise, the vehicle continues with the initial delivery plan, and the new demand point is deferred for arrangement in the next update. Under this update strategy, the vehicle's delivery process is illustrated in **Figure 2.**

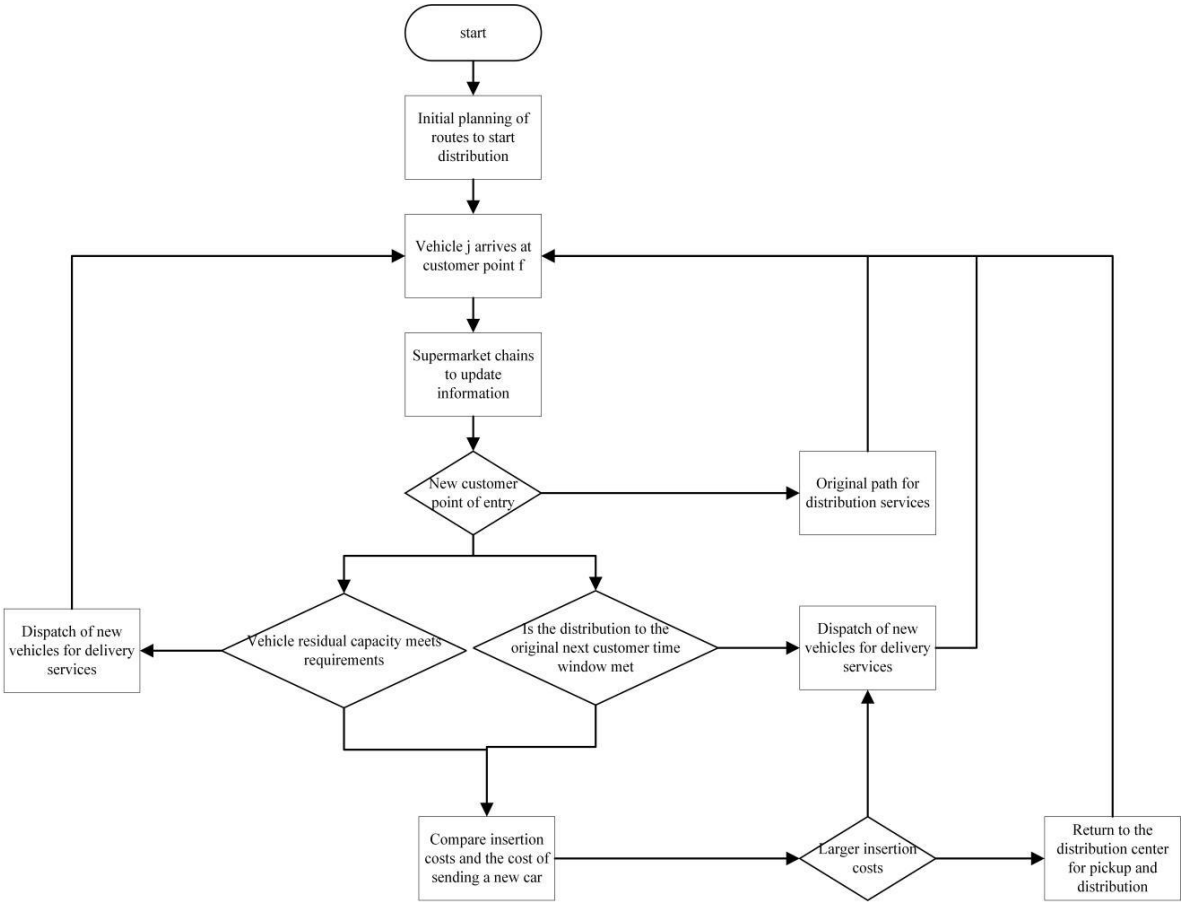

Figure 2 Vehicle routing with real-time customer request updates

5. Experimental Analyses and Results

5.1 Data Sets and Parameters

To comprehensively evaluate our proposed dynamic vehicle routing optimization model based on multi-head attention mechanisms and reinforcement learning, we designed a series of simulation experiments (parameters are shown in **Table 2**). These experiments cover small-scale (50 customers and 5 vehicles), medium-scale (100 customers and 10 vehicles), and large-scale (200 customers and 20 vehicles) scenarios to simulate various situations in real-world logistics delivery environments. Our goal is to verify the performance and adaptability of the model across a wide range of application scenarios through these tests of different scales.

TABLE 2 Characteristics of the Data Sets

Data Sets		
No. of depots (*O*), customers (*F*) and vehicles K		
Small datasets Depot = 1, customers =50, Vehicles = 5	Medium datasets Depot = 1, customers = 100, Vehicles = 10	Large datasets Depot = 1, customers = 200, Vehicles = 20
Parameters	*Model*	
Fixed cost of dispatch *fc*	100	
Cost of waiting for per unit minutes early arrival	0.5	

wc	
Penalty costs for per unit minutes late arrival of vehicles pc	1
Consumption cost per unit mile traveled by Vehicle c	20
Capacity of Vehicles Q^{van}	60
Speed of Vehicles v	20
the maximum allowable time for a Vehicle's route T_{max}	8
Demand of customer d_f	$Lognormal\ (\mu_{d_f}, \frac{1}{3}\ \mu_{d_f})$
Travel time of Vehicles c	$Lognormal\ (u_{\rho_{f_i f_j}}, \frac{UB-u_{\rho_{f_i f_j}}}{100} u_{\rho_{f_i f_j}})$ *
Time windows for each customer	$Lognormal\ (T_{fk}^{van} + \rho_{f_i f_j} + S_{fk}, \varpi - \rho_{f_i f_j} - S_{fk})$
* UB is the upper bound value, which is greater than the longest arc in the graph.	

In the experimental setup, we compared our method MA-RL, with traditional static optimization model, single-head attention mechanism model, dynamic model without attention mechanisms, genetic algorithm, particle swarm optimization algorithm, and ant colony optimization algorithm. All models were run under the same conditions to ensure fairness in comparison. For this purpose, we randomly generated the location (latitude and longitude) and demand of each customer, with demand volumes following a normal delivery with a mean between 300 and 1800, and a standard deviation range of 0 to 1/3 of the average demand. Additionally, we set time windows for each customer, referring to the benchmark data generation method of the vehicle routing problem with time windows (VRPTW) proposed by Solomon (1987), with appropriate adjustments [28].

When calculating the distance between customer points, we chose to use the straight-line distance due to the complexity and difficulty of obtaining real-world road network conditions. These distances were calculated using the Haversine formula based on the latitude and longitude of the customer points. Through this method, we could evaluate the performance of different algorithms in handling real vehicle routing optimization problems in a controlled environment, particularly their ability to handle dynamic changes and time window constraints.

The study addresses the DVRPSR using reinforcement learning methods and multi-head attention mechanisms. With 10 attention heads, we set the model to guarantee computational efficiency while sufficiently collecting and analyzing intricate input information. The hidden layers' dimensionality was first set at 512, then changed depending on experimental requirements. A learning rate of 0.002 was found inside the reinforcement learning framework to help to stabilize the learning process; the discount factor was set at 0.95, therefore enabling the model to balance instantaneous rewards with long-term advantages. To inspire the search of a larger solution space—which is progressively limited to support the convergence of the algorithm—a somewhat high initial exploration rate was used. Configured at 64, the sample count would improve memory management and batch processing performance all through the training cycle. The unique use cases and data characteristics guide the choice and improvement of these criteria. By means of repeated experimental testing, they undergo a process of continuous adaptation to attain the greatest potential learning outcomes and efficacy in fast changing environments.

5.2 Perfermence of the MA-RL

We designed and conducted a comprehensive series of experiments to precisely evaluate the

impact of MA-RL on DVRPSR. The experiments were structured into three different levels based on the number of customers and vehicle configurations: micro-level (50 clients and 5 vehicles), intermediate level (100 clients and 10 vehicles), and macro-level (200 clients and 20 vehicles). This layered experimental design allowed us to thoroughly investigate how effectively multi-head attention processes adapt and perform across systems of varying sizes.

Table 3 presents the results of our studies in detail. The first column illustrates different scales of the experiment, such as A-n50-k5, indicating 50 customers served by 5 vehicles. The second column shows the performance of seven different methods (Static Optimization, Single-Head Attention, MA-RL, Dynamic Without Attention Genetic Algorithm, Genetic Algorithm, Particle Swarm Optimization, and Ant Colony Optimization) across three scales. The third, fourth, and fifth columns display the total delivery route lengths, customer service coverage, and run times obtained by the various methods at different scales. Our findings demonstrate that MA-RL significantly outperformed other models in terms of both total and mean route lengths across all scales.

To highlight the optimal values of the mes at different levels, we compared the performance of the seven methods across the three scales and bolded the best results. Specifically, the MA-RL achieved a cumulative route length of 1830.07 in a limited-scale test, which is less than half of the 3660.14 path length reported by the stationary optimization model. This MA-RL also exhibited considerably shorter route lengths in both medium-sized and large-scale studies, consistently outperforming other methods. Additionally, MA-RL achieved the highest rate of meeting customer requests among all tested scales, further demonstrating its effectiveness in real-world scenarios. In the limited trial, the model reached a customer request satisfaction rate of 91%, significantly superior to the 71% achieved by the static optimization model. While MA-RL requires additional computational time, the extra workload is well-justified given the significant improvements it brings in optimizing routes and expanding customer service capabilities.

These experimental results not only validate the superiority of MA-RL in DVRPSR, but also demonstrate their potential practical value, especially in logistics and delivery fields that require dynamic responses and efficient route decisions. Through comparative analysis and practical performance evaluation, we further confirmed the powerful capabilities of the dynamic optimization model with multi-head attention mechanisms in handling complex routes and time window constraints.

TABLE 3 Comparison of Total Path Lengths, Computation Times, and Customer Service Coverage Rates for Different Models Across Various Scales

Scale	Model	Total Path Length	Customer Service Coverage Rate	Computation Times(S)
A-n50-K5	Static Optimization	3660.14	71%	**12**
A-n50-K5	Single-Head Attention	2200.1	85%	21
A-n50-K5	MA-RL	**1830.07**	**91%**	26
A-n50-K5	Dynamic Without Attention	2510.16	79%	14
A-n50-K5	Genetic Algorithm	2012.21	88%	31
A-n50-K5	Particle Swarm Optimization	2108.31	87%	28
A-n50-K5	Ant Colony Optimization	1950.68	90%	34
A-n100-K10	Static Optimization	4930.14	65%	**14**
A-n100-	Single-Head Attention	3000.1	79%	25

K10				
A-n100-K10	MA-RL	**2465.07**	**88%**	31
A-n100-K10	Dynamic Without Attention	3410.76	75%	23
A-n100-K10	Genetic Algorithm	2830.76	85%	39
A-n100-K10	Particle Swarm Optimization	2968.48	83%	37
A-n100-K10	Ant Colony Optimization	2753.53	86%	45
A-n200-K20	Static Optimization	8510.14	60%	**17**
A-n200-K20	Single-Head Attention	5500.1	78%	29
A-n200-K20	MA-RL	**4255.07**	**85%**	36
A-n200-K20	Dynamic Without Attention	5002.56	70%	27
A-n200-K20	Genetic Algorithm	4512.86	82%	49
A-n200-K20	Particle Swarm Optimization	4724.74	80%	46
A-n200-K20	Ant Colony Optimization	4351.84	84%	55
* A-n50-k5 indicates that this scale involves 5 vehicles delivering to 50 customers.				

5.3 Comparision of Nomalized Outcomes

To thoroughly explore and emphasize the importance of the experimental findings, we normalized the total path length, processing duration, and customer service coverage rate for each model across various experimental scales. This normalization process establishs a fair standard for more accurately evaluating the effectiveness of each model. Figure 3 through 5 illustrate the relative performance of multiple models at different scales using this approach.

Figure 3 shows that MA-RL consistently achieved remarkable performance across all experimental scales, significantly reducing path lengths and highlighting its efficiency in route optimization. In contrast, the stationary optimization model lagged behind, exhibiting the greatest increase in route length among all models. This stark contrast underscores the superior capability of the multi-head attention mechanism in solving route optimization problems in dynamic environments.

In **Figure 4**, MA-RL proved its strong capacity to satisfy consumer needs, achieving the highest satisfaction ratings across all test criteria. Conversely, the stationary optimization method showed the worst performance in customer service coverage, underscoring its inability to handle complex consumer demands in constantly changing circumstances.

Figure 5 shows that the stationary optimization framework outperformed other models in terms of conventional computation duration due to its simpler calculation technique, resulting in faster processing times. However, because MA-RL manages more complex data handling and decision-making tasks, it often requires longer processing times. This implies that whie MA-RL is more

efficient than its alternatives, there is still room for improvement in its computational efficiency.

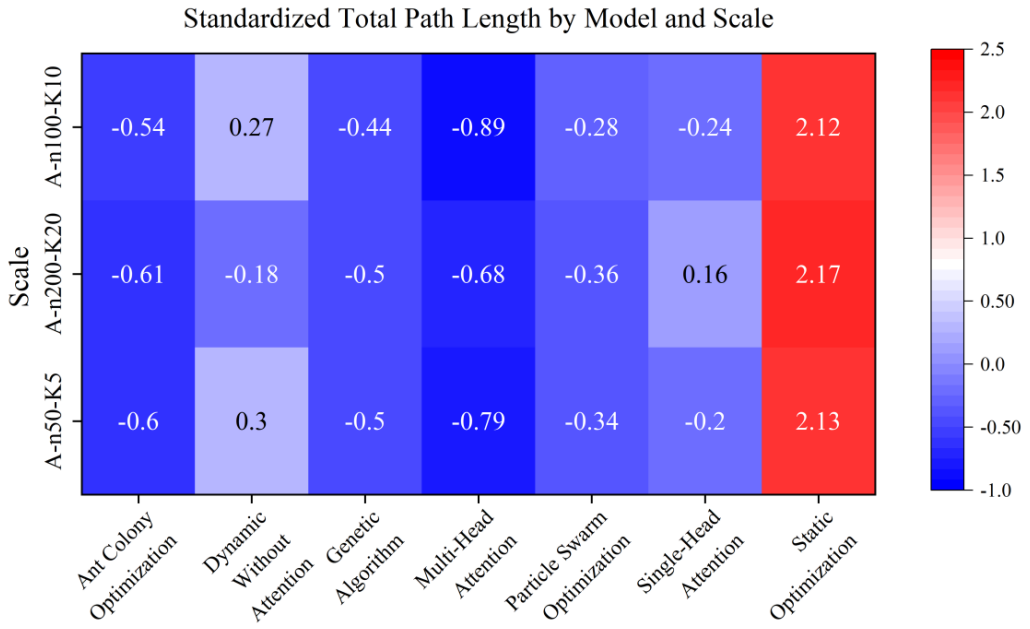

Figure 3 Comparison chart of standardized total path lengths of different models across various scales

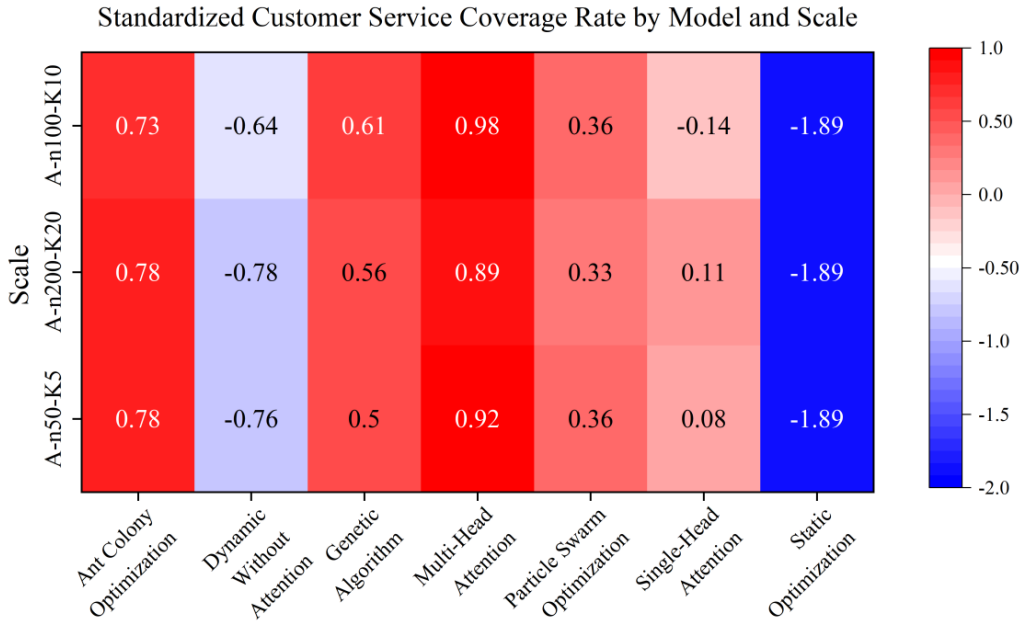

Figure 4 Comparison chart of standardized customer service coverage rates of different models across various scales

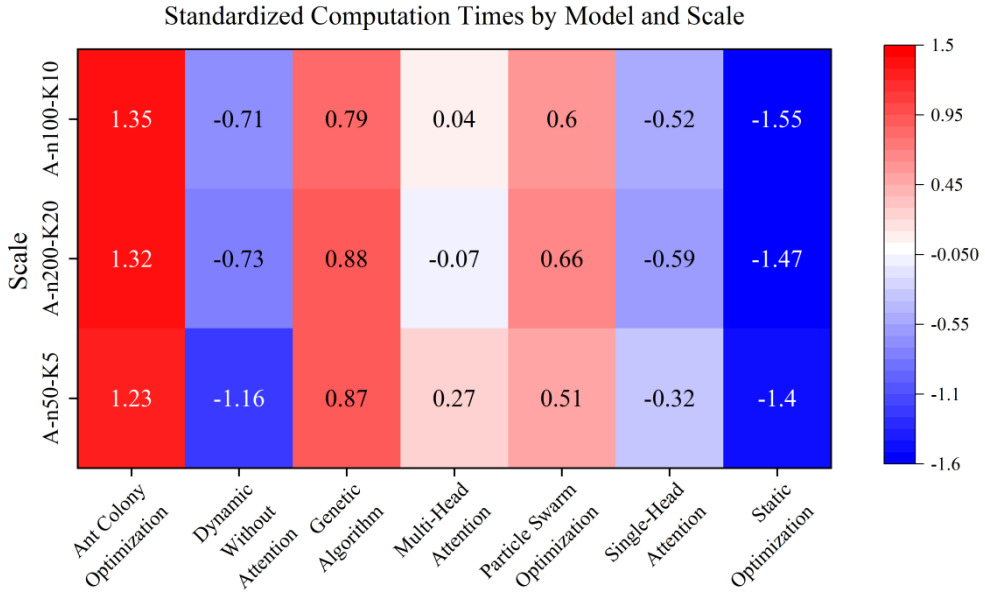

Figure 5 Comparison chart of standardized computation times of different models across various scales

6. Discussion and Conclusion

This study developed a novel adaptive route optimization method by integrating reinforcement learning with multi-head attention mechanisms, specifically targeting real-time customer request updates in the DVRPSR. Experimental results demonstrate significant advantages of MA-RL in reducing total path lengths and improving customer service coverage rates, compared to other models. These enhancements are particularly meaningful for practical applications in the logistics industry, especially in terms of increasing transportation efficiency and reducing energy consumption.

Future research will focus on optimizing the computational efficiency of the multi-head attention mechanisms and exploring predictive update strategies to further enhance the model's adaptability and efficiency in complex logistics networks. Efforts will also be dedicated to refining the model's architecture to reduce implementation costs and improve performance, ensuring rapid and effective responses to customer needs in dynamically changing market environments.

Additionally, the progress of this study signals a deeper integration of artificial intelligence, machine learning, and modern logistics management, driving digital transformation in the logistics industry. This integration enhances the flexibility and response speed of logistics services, helping businesses maintain a competitive edge in the fiercely competitive market by attracting and retaining a broader customer base. In the long term, these technological advancements will optimize logistics infrastructure at a societal level, promoting the construction of more efficient and sustainable logistics networks.

References

[1] Rahmanifar, G., M. Mohammadi, A. Sherafat, M. Hajiaghaei-Keshteli, G. Fusco, C. Colombaroni. Heuristic approaches to address vehicle routing problem in the Iot-based waste management system. Expert Systems with Applications. 2023. 220:119708.

[2] Mo, P., Y. Yao, A. D'Ariano, Z. Liu. The vehicle routing problem with underground logistics: Formulation and algorithm. Transportation Research Part E: Logistics and Transportation Review. 2023. 179:103286.

[3] Liu, M., Q. Zhao, Q. Song, Y. Zhang. A Hybrid Brain Storm Optimization Algorithm for Dynamic Vehicle Routing

Problem With Time Windows. IEEE Access. 2023. 11:121087-121095.
[4] Wang, S., W. Sun, M. Huang. An adaptive large neighborhood search for the multi-depot dynamic vehicle routing problem with time windows. Computers & Industrial Engineering. 2024. 191:110122.
[5] Teng, Y., J. Chen, S. Zhang, J. Wang, Z. Zhang. Solving dynamic vehicle routing problem with time windows by ant colony system with bipartite graph matching. Egyptian Informatics Journal. 2024. 25:100421.
[6] Liu, T., B. Zou, M. He, Y. Hu, Y. Dou, T. Cui, P. Tan, S. Li, S. Rao, Y. Huang, S. Liu, K. Cai, D. Wang. LncReader: identification of dual functional long noncoding RNAs using a multi-head self-attention mechanism. Briefings in Bioinformatics. 2023. 24(1):bbac579.
[7] Zahin, M. A. Multi-Headed Self-Attention Mechanism-Based Transformer Model for Predicting Bus Travel Times Across Multiple Bus Routes Using Heterogeneous Datasets [Master's Thesis]. Columbia, MO: University of Missouri-Columbia; 2023.
[8] Chen, J., J. Luo, editors. Enhancing Vehicle Routing Solutions Through Attention-Based Deep Reinforcement Learning. 2023 5th International Conference on Data-driven Optimization of Complex Systems (DOCS); 2023 22-24 Sept. 2023; Tianjin, China: IEEE.
[9] Sun, Y., J. Platoš. Abstractive text summarization model combining a hierarchical attention mechanism and multiobjective reinforcement learning. Expert Systems with Applications. 2024. 248:123356.
[10] Dantzig, G. B., J. H. Ramser. The Truck Dispatching Problem. Management Science. 1959. 6(1):80-91.
[11] Maroof, A., B. Ayvaz, K. Naeem. Logistics Optimization Using Hybrid Genetic Algorithm (HGA): A Solution to the Vehicle Routing Problem With Time Windows (VRPTW). IEEE Access. 2024. 12:36974-36989.
[12] Legrand, C., D. Cattaruzza, L. Jourdan, M.-E. Kessaci, editors. Investigation of the Benefit of Extracting Patterns from Local Optima to Solve a Bi-objective VRPTW; 2024; Cham: Springer Nature Switzerland.
[13] Speidel, V., editor EDP-assisted fleet scheduling in tramp and coastal shipping. Proceedings of the 2nd International Ship Operation Automation Symposium, Washington, DC, August 30-September 2, 1976 Proceedings expected to be available about December; 1976; 1976.
[14] Wilson, N. H. M., N. J. Colvin. Computer control of the Rochester dial-a-ride system: Massachusetts Institute of Technology, Center for Transportation Studies; 1977.
[15] Psaraftis, H. N. Vehicle routing: Methods and studies. Dynamic Vehicle Routing Problems. 16. Amsterdam: North Holland Publishing; 1988. p. 223-248.
[16] Ojeda Rios, B. H., E. C. Xavier, F. K. Miyazawa, P. Amorim, E. Curcio, M. J. Santos. Recent dynamic vehicle routing problems: A survey. Computers & Industrial Engineering. 2021. 160:107604.
[17] Pillac, V., M. Gendreau, C. Guéret, A. L. Medaglia. A review of dynamic vehicle routing problems. European Journal of Operational Research. 2013. 225(1):1-11.
[18] Bekta?s, T., P. P. Repoussis, C. D. Tarantilis. Chapter 11: Dynamic Vehicle Routing Problems. Vehicle Routing. MOS-SIAM Series on Optimization: Society for Industrial and Applied Mathematics; 2014. p. 299-347.
[19] Psaraftis, H. N., M. Wen, C. A. Kontovas. Dynamic vehicle routing problems: Three decades and counting. Networks. 2016. 67(1):3-31.
[20] Garrido, P., M. C. Riff. DVRP: a hard dynamic combinatorial optimisation problem tackled by an evolutionary hyper-heuristic. Journal of Heuristics. 2010. 16(6):795-834.
[21] Hanshar, F. T., B. M. Ombuki-Berman. Dynamic vehicle routing using genetic algorithms. Applied Intelligence. 2007. 27(1):89-99.
[22] Vinyals, O., M. Fortunato, N. Jaitly. Pointer networks. Advances in neural information processing systems. 2015. 28.
[23] Nazari, M., A. Oroojlooy, L. Snyder, M. Takác. Reinforcement learning for solving the vehicle routing problem. Advances in neural information processing systems. 2018. 31.
[24] Sutton, R. S., A. G. Barto. Reinforcement learning: An introduction. Cambridge: MIT press; 2018.
[25] Mnih, V., K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis. Human-level control through deep reinforcement learning. Nature. 2015. 518(7540):529-533.
[26] Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin. Attention is all you need. Advances in neural information processing systems. 2017. 30.
[27] Zhang M, Li P. Nested graph neural networks. Advances in Neural Information Processing Systems 34 (2021): 15734-15747.
[28] Solomon, M. M. Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations research. 1987. 35(2):254-265.