

Multi-Objective Deep Reinforcement Learning for the Optimization of Dynamic Large-Scale Vehicle-Cargo Matching Problem

Ran Tian*, Jian Yang, Qingxia Tang, Jiangtao Guo, Ruihan Yuan, Jitong Lian

College of Software, Northwest Normal University No.967, Anning East Road, Lanzhou 730070, China

Keywords: Vehicle cargo matching; Deep reinforcement learning; Deep Q-Learning; Generalization ability

Citation: Tian R., Yang J., Tang Q.X., et al.: Multi-Objective Deep Reinforcement Learning for the Optimization of Dynamic Large-Scale Vehicle-Cargo Matching Problem. *Data Intelligence*, 2026, Vol. 8, Art. No.: 20250306. DOI: <https://doi.org/10.3724/2096-7004.di.2025.0306>

ABSTRACT

This study presents a novel deep reinforcement learning algorithm, Deep Q-Learning with Long Short-Term Memory (LSTM) and a Multi-Head Self-Attention Mechanism (DLSAM), for the complex vehicle-cargo matching problem in logistics. The problem is formulated as a Markov Decision Process, guiding the selection of optimal vehicles for cargo shipments at each decision stage to meet dynamic and potentially large-scale operational demands. The framework of the proposed algorithm employs Deep Q-Learning to optimize the matching strategy through an action-value function, approximated by a deep neural network. LSTM networks are integrated to capture long-range temporal dependencies, essential for modeling the dynamic characteristics of vehicle-cargo matching. Concurrently, a Multi-Head Self-Attention mechanism enables the model to identify and assign varying levels of importance to crucial input features, thereby enhancing matching precision. An encoder-decoder architecture further aids in extracting global features from input data and generating effective decision sequences. Extensive experiments on diverse datasets, including scenarios of varying operational scales, compared this methodology with several algorithms such as Policy Gradient and standard Deep Q-Learning. The results demonstrate the superior performance of the developed algorithm across multiple objectives: type matching, time matching, loading rate, and empty running rate. Notably, this model achieves significantly improves type matching and time matching accuracy, while reducing the rates of empty running and empty loading. These evaluations underscore the algorithm's robustness, stability, and adaptability for diverse and large-scale vehicle-cargo matching scenarios.

*Corresponding author: Ran Tian (E-mail: tianran@nwnu.edu.cn; ORCID: 0000-0003-4435-580X).

1. INTRODUCTION

With the rapid expansion of the logistics industry, vehicle-cargo matching has become a critical factor of transportation efficiency and cost control—particularly in the context of heterogeneous cargo types, diverse service requirements, and multi-objective optimization demands. Efficient matching is essential not only for enhancing market competitiveness but also for ensuring the long-term sustainability of logistics enterprises [1]. Operationally, it requires balancing cargo characteristics, vehicle capacity, and delivery constraints to optimize fleet deployment while meeting customer expectations and improving service satisfaction. Intelligent matching strategies help mitigate key operational inefficiencies such as high deadhead mileage, suboptimal scheduling, and mismatched vehicle-cargo assignments, thereby strengthening fleet coordination and system adaptability. This is especially critical in sectors like express delivery, where thousands of daily orders—varying in origin, delivery windows, and cargo specifications—create highly heterogeneous demand. Effective matching must also account for external factors including traffic conditions, regulatory requirements, and payload limits, while ensuring safety and reliability for specialized shipments (e.g., temperature-sensitive or hazardous goods) [2]. Thus, vehicle-cargo matching transcends mere operational optimization; it serves as a foundational mechanism for guaranteeing service quality, regulatory compliance, and risk mitigation in modern logistics systems.

Figure 1 presents the three-stage framework of an integrated vehicle-cargo matching process. During the initialization phase, cargo and vehicle queues are systematically structured in accordance with predefined matching criteria prior to algorithmic processing. The second stage comprises three sequential operations within the vehicle-cargo matching platform: (1) Acquisition and preprocessing of fundamental goods and vehicle parameters. (2) Application of multi-objective optimization algorithms to evaluate potential matches. (3) Generation of optimal matching solutions through rapid historical data retrieval and dynamic

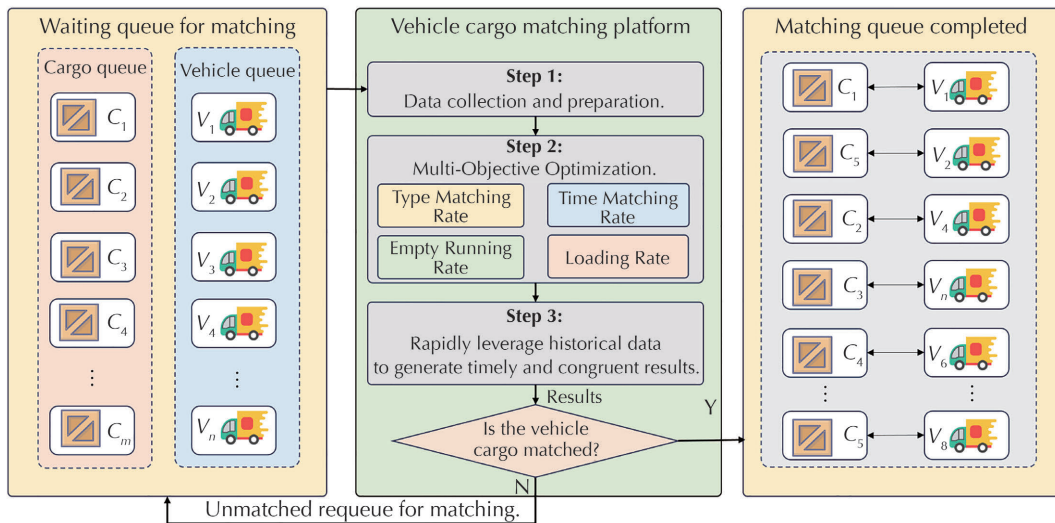


Figure 1. Vehicle cargo matching process.

reallocation of residual resources. In the final operational phase, successfully matched pairs are transferred to the confirmed queue, while the algorithm undergoes iterative optimization based on performance feedback. This establishes a closed-loop learning system that progressively enhances both matching precision and operational effectiveness. Despite significant progress, critical challenges remain in vehicle-cargo matching systems, particularly regarding:

Challenge 1: Within logistics and distribution networks, enhancing vehicle-cargo matching performance is fundamentally dependent on robust large-scale data management capabilities, particularly for: Foremost, matching precision fundamentally determines transportation system efficacy and cost-performance ratios in logistics operations. Optimal matching achieves three critical improvements: (1) minimization of empty-run mileage (2) maximization of load factor utilization, and (3) systemic enhancement of logistical throughput. Nevertheless, contemporary matching algorithms often struggle to preserve solution optimality when confronted with multidimensional real-world constraints, particularly in dynamic operating environments. Second, matching speed is crucial to meet the real-time demands of logistics distribution. As data volume increases, many algorithms face bottlenecks in computational speed, hindering their ability to deliver timely responses. Moreover, algorithmic scalability becomes imperative as logistics networks grow exponentially and dataset dimensions increase polynomially. These systems must optimally allocate computational resources and temporal constraints while maintaining robust generalization performance across heterogeneous, high-dimensional data distributions, thereby ensuring both operational stability and dynamic adaptability under variable conditions.

Challenge 2: The core challenge in vehicle-cargo matching for logistics operators stems from the inherent complexity of Pareto-optimal multi-criteria decision making, as opposed to conventional single-objective optimization problems. This challenge necessitates a dual approach: (1) minimization of deadhead mileage to enhance asset utilization, and (2) multi-criteria optimization incorporating temporal efficiency, load factor maximization, and vehicle-cargo compatibility constraints. Unlike single-objective formulations that isolate individual metrics (e.g., empty-run reduction or capacity utilization), the multi-objective framework requires dynamic trade-off analysis among competing Pareto-optimal solutions. Temporal performance, a key determinant in logistics system operations, exerts direct influence on both client satisfaction metrics and aggregate operational productivity. Furthermore, load factor optimization enables maximal utilization of both vehicle cubic capacity and weight-bearing potential, thereby achieving significant transportation cost reductions through improved asset productivity.

To address these challenges, we propose DLSAM (Deep Q-Learning with Long Short-Term Memory and Self-Attention Mechanism), an integrated framework combining Deep Q-Learning, Long Short-Term Memory (LSTM) networks, and self-attention mechanisms. The key innovations of this study are fourfold:

- This study introduces an innovative multi-objective dynamic weight reinforcement learning mechanism, which effectively balances multiple optimization objectives in vehicle-cargo matching, such as type matching, time matching, load rate matching, and empty running rate matching. This approach enables global optimization, enhancing overall efficiency and resource utilization in logistics systems.
- A dynamic vehicle-cargo matching framework based on the Markov Decision Process (MDP) is developed in this study. This framework dynamically adjusts matching strategies according to the real-time changes in the environment and demands, thereby optimizing transportation efficiency and resource allocation, improving the responsiveness and flexibility of the logistics system.
- A spatiotemporal feature joint encoder, combining LSTM networks and multi-head attention mechanisms, is implemented in this research. This encoder effectively captures the complex spatiotemporal dependencies between vehicles and cargo, significantly improving the matching accuracy and efficiency, particularly in handling time-series data and long-term dependencies.
- The proposed algorithm efficiently handles large-scale datasets, ensuring rapid execution and scalable performance in real-world logistics scenarios. This capability enables the method to manage the growing volume of data and complex matching requirements, adapting to logistics environments of varying sizes.

The paper is organized as follows: Section 2 reviews conventional approaches, deep learning architectures, and training paradigms relevant to vehicle-cargo matching. Section 3 formalizes the problem through mathematical modeling and optimization formulations. Section 4 details our proposed reinforcement learning framework and deep neural network designs. Section 5 presents experimental results with comparative performance analysis. Section 6 examines practical applications and operational impacts. Section 7 concludes with research contributions and future directions in intelligent logistics optimization.

2. RELATED WORK

The issue of vehicle-cargo matching has attracted significant attention from researchers, leading to substantial advancements. Research in this area primarily focuses on three key areas: vehicle-cargo matching and optimization algorithms, vehicle-cargo matching and intelligent systems, and vehicle-cargo matching and supply chain management. This section provides a summary of the research dedicated to optimizing vehicle-cargo matching, as presented in Table 1.

In the field of vehicle-cargo matching problem and optimization algorithms, various techniques and technologies have been extensively explored to enhance matching efficiency and reduce logistics costs. Bayesian networks and probabilistic models, as predictive tools, have demonstrated their potential in addressing uncertainty and dynamically changing environments in early research on vehicle cargo matching ([3-4]). However, a dependency that hinders their ability to meet real-time requirements and adapt to dynamic environmental changes, which results in limited prediction accuracy in complex scenarios. Despite these limitations, they provide valuable insights for decision-makers by analyzing

Table 1. Overview of Research Paper Classification in the Field of Vehicle Cargo Matching Problem.

Question	Algorithm	Source
Vehicle Cargo Matching and Optimization Algorithm	Bayesian Networks and Probability Models	[3]
		[4]
	Genetic and Evolutionary Algorithms	[5]
		[6]
	Optimization and Scheduling Model	[7]
		[8]
	Stable Matching and Decision Theory	[9]
		[10]
	Quantum Evolution and Intelligent Algorithms	[11]
		[12]
Vehicle Cargo Matching and Intelligent System	Deep Learning and Intelligent Matching	[13]
		[14]
	Application of Blockchain Technology	[15]
		[16]
	Intelligent Optimization Algorithm	[17]
		[18]
	Internet of Vehicles and Logistics IoT	[19]
		[17]
	Multimodal Matching and Resource Sharing Platform	[20]
		[21]
Vehicle Cargo Matching and Supply Chain Management	Supply Chain Risk and Resilience	[22]
		[23]
	Cold Chain Logistics Management	[24]
		[25]
	Vehicle Cargo Matching Platform and Strategy Selection	[26]
		[27]
Fair and Transparent Supply Chain	[28]	
	[29]	

historical data and estimating the probability of successful vehicle cargo matching. Additionally, genetic and evolutionary algorithms, particularly intelligent genetic algorithms (IGA), have played a significant role in addressing complex vehicle cargo matching challenges. The studies referenced in [5-6], and [7] illustrate methods for optimizing the vehicle cargo matching problem by enhancing genetic algorithms. They also demonstrate how to improve the fairness and efficiency of matching through adaptive interaction algorithms and three-party evolutionary games. These contributions extend beyond algorithmic performance enhancements to establish both a theoretical foundation and practical guidance for the stable operation of the online freight industry. Optimization and scheduling models are equally vital in vehicle cargo matching, as they minimize transportation costs and time through precise mathematical models and algorithms. However, the optimization process can be time-consuming, often becoming trapped in local optima, and it is challenging to efficiently manage multidimensional objectives and dynamic constraints. The studies referenced in [8-9], and [10] have proposed various optimization models, including mixed-integer nonlinear programming and simulated annealing algorithms. These models have achieved significant results in reducing the distance for empty vehicle pick-up and drop-off, as well as lowering transportation costs. Particularly in the multi-delivery point services of freight O2O platforms, the application of these optimization models greatly enhances the efficiency and market competitiveness of the platform. Stable matching and decision theory play a crucial role in vehicle cargo matching. They focus on achieving stable matching relationships among multiple stakeholders; however, they often overly emphasize matching stability while neglecting flexibility and multi-objective optimization in dynamic environments. The study referenced in [11] ensured the stability of the matching process and balanced the interests of all parties by constructing a stable matching model. Quantum evolution and intelligent algorithms represent cutting-edge technologies in the field of vehicle cargo matching. The research cited in references [12], and [13] harnesses the potential of quantum evolutionary algorithms to address vehicle cargo matching problems, demonstrating superior performance in managing large-scale and complex issues. Although these algorithms show promise in handling extensive problems, their practical application is constrained by implementation complexity and technological maturity.

In the field of vehicle cargo matching and intelligent systems, reference [14] investigates the application of deep learning technology to enhance the intelligence of vehicle cargo matching. By employing the Deep Belief Network model in conjunction with the Chaos Krill Herd Algorithm (CKHA) to optimize hyperparameters, this study achieved significant improvements in matching accuracy, precision, and recall rates, offering innovative approaches for developing efficient and secure vehicle cargo matching systems. However, while the optimization accuracy is high, the adaptability to complex dynamic environments is limited, and there is a notable lack of real-time adjustment capabilities. Reference [15] integrates blockchain technology with vehicle cargo matching and proposes an Intelligent Autonomous Transportation System (IATS) based on blockchain. By utilizing the Ethereum blockchain to securely store sensitive data such as orders and goods, and optimizing vehicle cargo matching through smart contracts and the LightGBM algorithm, these studies not only enhance data security but also improve logistics efficiency. Nevertheless, their computational complexity and real-time requirements restrict the enhancement of matching efficiency. References [16-17], and [18] focus on the application of

intelligent optimization algorithms in vehicle cargo matching. Through adaptive parameter control and colony optimization algorithm, mixed integer programming model, and adaptive large neighborhood search (ALNS) algorithm, these studies demonstrate how to improve the efficiency and accuracy of vehicle cargo matching through algorithm innovation. The optimization performance is strong, but it is difficult to simultaneously balance efficiency and matching accuracy in multiobjective optimization scenarios. References [19], and [17] discussed the role of the Internet of Vehicles and Logistics Internet of Things (L-IoT) in vehicle cargo matching. Through IoT technology, real-time monitoring and intelligent matching of vehicle and cargo information have been achieved, providing essential technical support for the intelligence and automation of the logistics industry. However, its primary application remains in information collection and sharing, lacking the optimization capabilities necessary for addressing complex matching problems. References [20-21], and [22] focus on the development of multimodal matching and resource-sharing platforms. These studies have optimized the routing of goods in urban logistics distribution from central warehouses to satellite points or self-pickup cabinets using mixed-integer programming models and Adaptive Large Neighborhood Search (ALNS) heuristic algorithms, significantly reducing transportation costs and vehicle travel distances. Nevertheless, these approaches struggle to accommodate large-scale dynamic matching requirements and exhibit limited optimization capabilities for global objectives.

In the field of vehicle cargo matching and supply chain management, references [23], and [24] conducted an in-depth analysis of risk factors within the supply chain, including the “ripple effect” caused by supplier interruptions. They proposed corresponding risk assessment and management strategies. These studies offer significant theoretical and practical guidance for enhancing supply chain stability and responding to potential disruptions by integrating discrete-time Markov chain (DTMC) and dynamic Bayesian network (DBN) models. Reference [28], and [27] examine the influence of equitable distribution attention on the quality of information disclosure behavior in binary supply chains. The research findings indicate that fair behavior among supply chain members can lead to an overall Pareto improvement, offering a new perspective on balancing fairness concerns with economic benefits in supply chain management. Reference [25] proposed an integrated multi-criteria decision-making (MCDM) method for evaluating and selecting pharmaceutical cold chain logistics suppliers. By combining the SWARA and CoCoSo methods and introducing a probabilistic language environment, this study provides a reliable evaluation tool for minimizing transportation and storage risks while ensuring drug quality. References [26] examined the bilateral matching and user policy selection issues in freight resource-sharing platforms. By constructing a bilateral matching model and incorporating behavioral economics theory, these studies offer management recommendations for optimizing matching efficiency and success rates, which can enhance the operational efficiency of the platform.

Although methods such as Bayesian networks, genetic algorithms, and optimization models have been widely applied in vehicle-cargo matching in existing research, several challenges remain unresolved. These methods typically rely on historical data, exhibit weak adaptability to dynamic environments, and struggle to achieve real-time adjustments and multi-objective balance, particularly under complex environmental changes. To address these issues, this paper proposes a deep Q-network-based reinforcement learning

framework ([29]), which dynamically optimizes the matching strategy through continuous interaction with the environment. This framework not only meets real-time demands but also comprehensively considers multiple key indicators, including type matching degree, time matching degree, loading rate, and empty running rate, to design a multi-objective optimization mechanism for efficient multi-objective balance. This modeling approach aims to overcome the limitations of traditional algorithms in multi-objective optimization and dynamic constraints, providing an effective solution for improving logistics distribution efficiency ([30]).

3. PRELIMINARIES

In this section, we present the mathematical formulation of the vehicle-cargo matching problem. When addressing the vehicle-cargo matching problem, it is essential to consider various factors. To systematically represent these factors and their interrelationships, we use a set of symbols to denote key variables and provide a table (see Table 2) that outlines the physical meanings and definitions of each symbol. By defining these symbols, we can more effectively articulate the characteristics of both vehicles and goods, thereby establishing a comprehensive mathematical model that forms the basis for subsequent optimization scheduling.

In the problem of vehicle-cargo matching, our objective is to achieve dual optimization by minimizing transportation costs and maximizing resource utilization efficiency through the selection of an effective matching scheme between vehicles and cargoes. To this end, we introduced the matching utility value as the evaluation criterion and established an objective function for quantitative analysis (as shown in Formula 1). This objective function takes into account four primary indicators: Type Matching Degree (TM), Time Matching Degree (TiM), Empty Running Rate Matching Degree (ER), and Loading Rate Matching Degree (LR). By applying appropriate weighting to these indicators, we aim to identify the current optimal vehicle-cargo matching strategy under various constraint conditions. The weight assigned to each matching degree can be adjusted based on actual business needs, allowing for the reflection of the relative importance of different optimization objectives within the overall scheduling strategy.

$$\max \sum_{i=0}^n \sum_{j=0}^m (w^{TM} \cdot r_{i,j}^{TM} + w^{TiM} \cdot r_{i,j}^{TiM} + w^{ER} \cdot (1 - r_{i,j}^{ER}) + w^{LR} \cdot r_{i,j}^{LR}) \quad (1)$$

Where:

$$i \in V, j \in C, w^{TM} + w^{TiM} + w^{ER} + w^{LR} = 1 \quad (2)$$

In optimizing vehicle-cargo matching, these indicators assess the compatibility between vehicles and cargos from multiple perspectives during transportation, providing a basis for enhancing scheduling strategies. In the above formulas, $r_{i,j}^{TM}$, $r_{i,j}^{TiM}$, $r_{i,j}^{ER}$, and $r_{i,j}^{LR}$ represent the specific matching degree indicators between vehicle i and cargo j . To explicitly formulate the calculation of these metrics, their detailed mathematical definitions are provided in the subsequent sections.

Table 2. Explanation of Variable Meaning.

Notation	Physical Description
V_i	Represents a vehicle indexed by i
C_j	Represents cargo or a customer indexed by j
V_i^{type}	The type of vehicle i , such as vehicle type (e.g., flatbed, refrigerated).
V_i^{load}	represents the load capacity of vehicle i , defined as the maximum weight of cargo it is permitted to carry.
V_i^{volume}	The volume capacity of vehicle i , which determines how much cargo it can hold.
V_i^{pos}	The current position of vehicle i , typically in a coordinate format (e.g., GPS location).
$V_i^{start_time}$	Refers to the time point when vehicle i can start executing tasks.
$V_i^{end_time}$	The end time when vehicle i must complete its tasks or return to its depot.
C_j^{type}	The type of cargo j , such as the nature of the goods (e.g., perishable, hazardous).
$C_j^{quality}$	The quality or condition requirements of cargo j , which may affect its handling or vehicle selection.
$C_j^{start_pos}$	The starting position or origin of cargo j , where the goods need to be picked up.
$C_j^{end_pos}$	The destination or drop-off position of cargo j .
$C_j^{start_time}$	The earliest time that cargo j can be picked up.
$C_j^{end_time}$	The latest time by which cargo j must be delivered.
$R_{i,j}^{TM}$	The type matching degree between vehicle i and cargo j , evaluating how well the vehicle type fits the cargo requirements.
$R_{i,j}^{TiM}$	The time matching degree between vehicle i and cargo j , evaluating the compatibility of time windows for pickup and delivery.
$R_{i,j}^{ER}$	The empty running rate between vehicle i and cargo j , reflecting the efficiency in minimizing empty travel.
$R_{i,j}^{LR}$	The loading rate or capacity utilization rate between vehicle i and cargo j , indicating how well the vehicle's capacity is used for the cargo's volume.
S	The state S in the Markov process represents all vehicle and cargo information at a given time, serving as the input for decision-making.
A	" A represents the set of actions in a Markov process, which are the choices an agent can make in a state, such as selecting a vehicle for specific cargo."
τ	τ represents the state transition rule in Markov processes, used to describe the subsequent situation of an agent after performing a certain action.
R	R represents the reward function in Markov processes, which is the feedback obtained by the system after taking a certain action in a certain state

3.1 Type Matching Degree

TM is a crucial indicator for assessing the compatibility between vehicle and cargo types in logistics transportation. It evaluates the degree of match by considering the alignment between the vehicle type and the cargo type. The formula is:

$$r_{i,j}^{\text{TM}} = f(V_i, C_j) \quad (3)$$

In the field of logistics and transportation, the compatibility between vehicle types and cargo types is one of the critical factors influencing transportation efficiency and safety. To quantify this compatibility, we introduce an evaluation function $f(V_i, C_j)$ that assesses the degree of match between a specific vehicle type and a cargo type. The computation of this function is based on the definition provided in reference [3], but we have provided more types of vehicles and cargos on this basis, making the matching more diverse. The function's value is constrained within the range of 0 to 1, where a value of 1 indicates a perfect match between the vehicle and cargo types, while a value of 0 denotes a complete lack of compatibility.

3.2 Time Matching Degree

TiM is a crucial factor that assesses the alignment between the time windows of vehicles and cargo. This metric is quantified by calculating the overlap between the time windows of the vehicle and the cargo, as illustrated in Equation (4). This method offers a quantitative measure for evaluating and optimizing the temporal coordination between vehicles and cargo, ultimately enhancing transportation efficiency.

$$r_{i,j}^{\text{TiM}} = \frac{\min\{V_i^{\text{end_time}}, C_j^{\text{end_time}}\} - \max\{V_i^{\text{start_time}}, C_j^{\text{start_time}}\}}{C_j^{\text{end_time}} - C_j^{\text{start_time}}} \quad (4)$$

Equation (4) determines the time matching degree by calculating the overlap between the time windows of vehicles and cargoes, thereby assessing their temporal compatibility. The numerator of the equation represents the length of the overlapping time window, which is the intersection of the vehicle's and cargo's time windows, while the denominator denotes the total length of the cargo's time window. This quantitative approach provides decision-making support for logistics platforms in vehicle scheduling and cargo transportation, optimizing the temporal coordination between vehicles and cargo and, consequently, enhancing transportation efficiency.

3.3 Empty Running Rate Matching Degree

In the field of logistics and transportation, ER is a critical metric for evaluating transportation efficiency [31]. It reflects the proportion of distance or time that a vehicle travels without carrying any cargo. A high empty running rate typically indicates inefficient utilization of vehicle resources, which not only increases fuel consumption and maintenance costs but also diminishes the overall economic efficiency of transportation. Therefore, reducing the empty running rate is significant for enhancing transportation efficiency and minimizing resource waste. The formula for calculating the empty running rate is as follows:

$$r_{i,j}^{\text{ER}} = \frac{L_{i,j}^1}{L_{i,j}^1 + L_{i,j}^2} \quad (5)$$

Among them, $r_{i,j}^{\text{ER}}$ represents the empty running rate between vehicle i and cargo j . $L_{i,j}^1$ represents the distance from the current position V_i^{pos} of the vehicle to the starting position C_j^{start} of the cargo. $L_{i,j}^2$ represents the distance of the goods from the starting position C_j^{start} to the ending position C_j^{end} . The values of $L_{i,j}^1$ and $L_{i,j}^2$ in the formula can be calculated as follows:

$$L_{i,j}^1 = \text{distance}(V_i^{\text{pos}}, C_j^{\text{start_pos}}) \quad (6)$$

$$L_{i,j}^2 = \text{distance}(C_j^{\text{start_pos}}, C_j^{\text{end_pos}}) \quad (7)$$

Within this formulation, $\text{distance}(\text{pos}_1, \text{pos}_2)$ represents the distance traveled from position pos_1 to position pos_2 .

3.4 Loading Rate Matching Degree

Both weight and volume are critical factors in vehicle-cargo matching, as vehicle payload capacities and volumetric limits are fixed. When the weight of the cargo to be assigned is significantly less than the vehicle's rated payload, such a discrepancy may indicate suboptimal matching or result in excessive operational costs. Conversely, if the cargo weight exceeds the vehicle's payload capacity, it poses significant safety risks to both the driver and the vehicle. In such instances, transportation may be directly rejected.

To differentiate between weight-sensitive and volume-sensitive cargo, a quantitative criterion must be established. Specifically, when the weight of the cargo is significant, calculations should be based on its mass. Conversely, if weight is not the dominant factor, volume should serve as the primary basis for computation. Various modes of transportation may employ different criteria to make this distinction. To address this issue, we adopt the definition of loading rate matching degree from reference [3] as the foundation for the definition utilized in this study. The corresponding formulas are presented in Equations (8) and (9).

$$\theta = C_j^{\text{volume}} / 0.006 \quad (8)$$

$$r_{ij}^{\text{LR}} = \begin{cases} C_j^{\text{volume}} / V_i^{\text{volume}}, & \text{if } \theta \leq C_j^{\text{quality}} \\ C_j^{\text{quality}} / V_i^{\text{load}}, & \text{otherwise} \end{cases} \quad (9)$$

Equation (8) defines the threshold θ , where C_j^{volume} represents the volume of cargo j . The sensitivity type of the cargo is determined by the ratio of C_j^{volume} to the constant 0.006, which serves as a critical density threshold (equivalent to $0.006 \text{ m}^3/\text{t}$) to distinguish between weight-sensitive (heavy) and volume-sensitive (light/bulky) cargo. Specifically, this constant implies that if a cargo's effective density falls below this value, it is treated as volume-constrained; otherwise, it is considered weight-constrained. Equation (9)

defines the loading rate matching degree $r_{i,j}^{LR}$, which represents the relationship of loading rate compatibility between vehicle i and cargo j .

Meanwhile, the vehicle cargo matching problem satisfies the following constraints:

$$V_i^{\text{load}} \geq C_j^{\text{quality}}, V_i \in V, C_j \in C \quad (10)$$

$$V_i^{\text{volume}} \geq C_j^{\text{volume}}, V_i \in V, C_j \in C \quad (11)$$

$$\min\{V_i^{\text{end_time}}, C_j^{\text{end_time}}\} - \max\{V_i^{\text{start_time}}, C_j^{\text{start_time}}\} > 0, V_i \in V, C_j \in C \quad (12)$$

$$V_i^{\text{end_time}} \geq V_i^{\text{start_time}}, V_i \in V \quad (13)$$

$$C_j^{\text{end_time}} \geq C_j^{\text{start_time}}, C_j \in C \quad (14)$$

$$\frac{L_{i,j}^1}{L_{i,j}^1 + L_{i,j}^2} \leq \delta \quad (15)$$

$$f(V_i, C_j) > 0, V_i \in V, C_j \in C \quad (16)$$

The constraints provided in the formulas are designed to ensure the rationality and efficiency of the vehicle-cargo matching process. Formula (10) specifies that the load capacity of vehicle i must be greater than or equal to the weight of cargo j , where V and C represent the sets of vehicles and cargoes, respectively. Formula (11) ensures that the loading volume of vehicle i must be greater than or equal to the volume of cargo j . Formula (12) ensures that the available time window of the vehicle overlaps with the required transportation time of the cargo, i.e., the vehicle must be available during the time when the cargo needs to be transported. Formula (13) ensures that the end time of a vehicle is later than its start time, preventing logical errors in scheduling. Similarly, Formula (14) ensures that the end time of the cargo is later than its start time. Formula (15) places a limit on the empty running rate, ensuring that the proportion of empty running during the transportation process does not exceed a maximum threshold δ . Here, δ represents the maximum allowable empty running rate, a constraint designed to minimize empty running and thereby enhance transportation efficiency. Finally, Formula (16) ensures that there is a form of compatibility or matching between vehicles and cargoes, meaning their types must align to facilitate effective transportation.

4. METHODOLOGY

To address the multi-objective vehicle-cargo matching problem, we propose the DLSAM algorithm.

4.1 Markov Decision Process for Vehicle Cargo Matching

When addressing the vehicle-cargo matching problem, we model the issue using a Markov Decision Process (MDP) [32]. The MDP framework provides a systematic approach to formalizing decision-making

in vehicle-cargo matching as a dynamic relationship among states, actions, and rewards. Figure 2 illustrates the MDP for the vehicle-cargo matching problem. In this model, each state represents the current distribution of vehicles and cargoes, each action corresponds to a matching decision, and the reward reflects the outcomes of those decisions (e.g., transportation efficiency or profit). This approach enables a clear representation of the dynamic changes within the vehicle-cargo matching problem and facilitates the identification of optimal decision paths, thereby offering theoretical support for scheduling optimization. In order to effectively handle the dynamic changes in requirements, this paper adopts a multi time slice modeling/solving strategy.

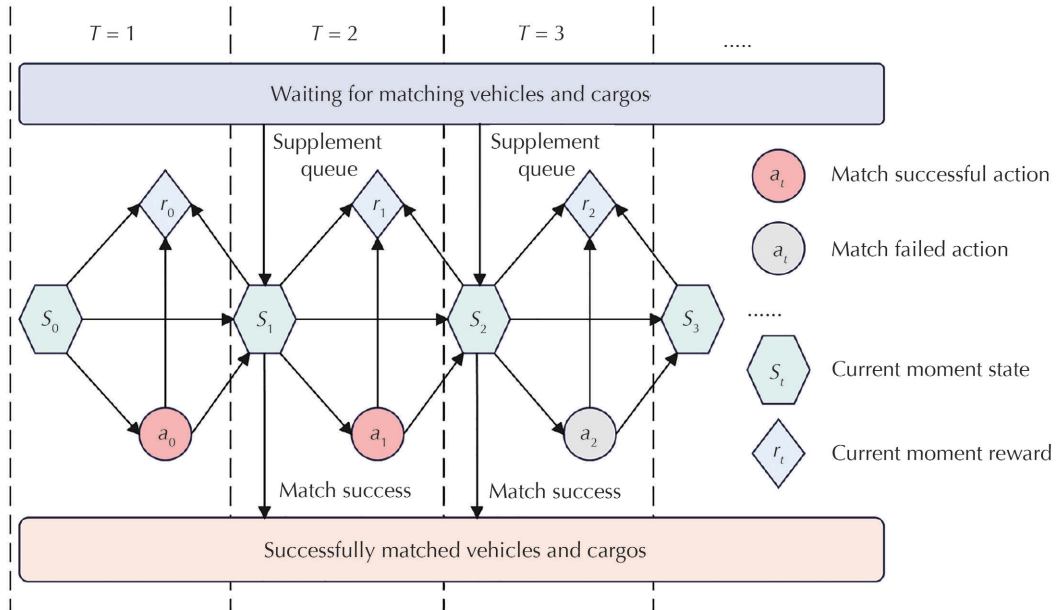


Figure 2. Markov Decision Process Diagram for Vehicle Cargo Matching Process.

To capture the system's dynamic behavior and enable effective decision-making, we partition the entire decision cycle into multiple discrete time slices, denoted as $T = 1, 2, \dots, t$. The length of each time slice is Δt .

Figure 2 presents a schematic diagram of MDP, which includes states and actions such as vehicles and cargo waiting for matching, queue replenishment, successful matching actions, and unsuccessful matching actions. An MDP is typically represented by a four-tuple, which consists of the following components:

- s : In our MDP model, each state is composed of information about both the cargo and the vehicles. Specifically, it is expressed as follows: $s = (V, C, M)$, The fundamental attributes of the vehicle information are represented as $V = [V_1, V_2, \dots, V_n]$, The fundamental attributes of the cargo information are represented as $C = [C_1, C_2, \dots, C_m]$. M represents the current matching status between vehicles and cargo, $M_{i,j}$ indicates whether vehicle i and cargo j are successfully matched. A successful match is represented by 1, and a failure is represented by 0. The calculation formula is as follows:

$$M_{i,j} = \begin{cases} 1, & \text{if } C(i, j) \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

Similarly, the weight vector $w(t) = [w^{TM}(t), w^{TiM}(t), w^{ER}(t), w^{LR}(t)]$ is dynamically determined by the environmental state s . For example, when the proportion of time sensitive tasks exceeds the threshold, the system automatically increases the weight of $w(t)$ to force to priority matching of vehicles and goods with more urgent time windows.

$$C(i, j) = (r_{ij}^{TM} \cdot r_{ij}^{TiM} \cdot \dots) \wedge (r_{ij}^{LR} > \delta^{LR}) \wedge (r_{ij}^{ER} < \delta^{ER}) \quad (18)$$

Where, δ^{LR} represents the threshold of loading rate, and δ^{ER} represents the threshold of empty running rate. In practical applications, these values are usually set based on specific business needs, cost considerations, and operational goals. A vehicle V_i and cargo C_j are considered successfully matched only when all specified conditions are met, at which point $M_{i,j} = 1$. Otherwise, the match is considered unsuccessful, and $M_{i,j} = 0$. This matching mechanism ensures efficient and reasonable pairing between vehicles and cargoes through strict criteria, thereby optimizing the overall operation of the logistics system.

- a : The action space A is a critical component that defines all possible actions the agent can take in a given state s . In the context of the vehicle-cargo matching problem, the action space A includes all possible vehicle-cargo matching attempts. An action a is represented by an integer encoding, where $a \in A$. This encoding is unique and can be decoded into a tuple (i, j) , where i represents the index of the vehicle and j represents the index of the cargo. Performing action a in state s means that the agent instructs vehicle V_i to depart from its initial location, travel to the location of cargo C_j , and then transport the cargo to its destination. Each action is encoded as a unique integer through bijective mapping $\phi: (i, j) \mapsto a$, which facilitates integration with deep reinforcement learning architectures.
- τ : The state transition rules not only dictate the evolution of the system's state after performing a specific action but also encompass the updates of key elements such as the matching matrix, the locations of vehicles and cargo, and their idle statuses. With precise state transition rules, the agent can more accurately predict the evolution of future states, thereby enabling better decision-making. The implementation of state transitions must comprehensively account for the dynamic characteristics of the system and the uncertainties of the environment to ensure that the agent can efficiently learn and adapt in complex scenarios.

Whenever a vehicle completes a delivery task, the environment must be updated according to the new state. The update process involves three key steps.

- 1) Action Validity Update: Determine whether the currently selected vehicle and cargo have already been chosen in previous assignments to avoid redundancy.
 - 2) Matching Status Update: Adjust the matching status of vehicles and cargo in realtime based on the completion of the current transport task. For instance, when a piece of cargo is successfully assigned and loaded, its status should be updated to “in transit” or “assigned.”
 - 3) Vehicle and Cargo Information Update: Reassess the remaining states of vehicles and cargo, such as the remaining carrying capacity of the vehicle and the outstanding delivery volume of the cargo. These updates support subsequent scheduling and decision-making.
- R : For the vehicle-to-cargo matching problem, Equation (19) provides a method to more comprehensively evaluate the impact of different matching performance criteria.

$$R = w^{\text{TM}} \cdot r_{i,j}^{\text{TM}} + w^{\text{TIM}} \cdot r_{i,j}^{\text{TIM}} + w^{\text{ER}} \cdot (1 - r_{i,j}^{\text{ER}}) + w^{\text{LR}} \cdot r_{i,j}^{\text{LR}} \quad (19)$$

Equation (19) defines the reward function for the vehicle-cargo matching problem, establishing the reward R as a weighted sum of four matching criteria: type compatibility, time efficiency, empty travel rate, and load utilization. Each criterion is assigned a specific weight to reflect its importance in the overall reward. This design enables the algorithm to evaluate matching performance comprehensively by considering multiple factors and allows for the adjustment of weights to adapt to the requirements of various logistics scenarios, thereby optimizing the decision-making process. The weights w_t are not fixed, but are adaptively adjusted online based on system performance. Specifically, we adopt a mechanism based on a mechanism inspired by proportional-integral (PI) control:

$$w_t^k = \alpha \cdot \bar{r}_t^k + (1 - \alpha) \cdot w_{t-1}^k, \quad (20)$$

where $\bar{r}_t^k$ is the moving average score of objective k over the past H steps, and $\alpha \in (0, 1)$ controls the adaptation speed. If a certain objective consistently performs poorly (e.g., $\bar{r}_t^{\text{TIM}}$ is low due to frequent time-window violations), its weight will be automatically increased, thereby guiding the policy to assign higher priority to satisfying that objective. This mechanism effectively alleviates conflicts among multiple objectives and avoids manual hyperparameter tuning.

The design of dynamically adjusting α is inherently intended to mitigate the sensitivity issue associated with fixed parameters: when the system reward fluctuates significantly, α will automatically decrease to smooth weight updates and avoid convergence oscillations; when the system tends to stabilize, α will appropriately increase to accelerate the convergence rate. This dynamic characteristic allows the model to adaptively maintain the convergence stability of multi-objective rewards without relying on sensitivity tests for fixed α . Compared with the fixed α mechanism, the dynamic α does not require manual presetting of the trade-off between convergence speed and stability via parameters. Instead, it automatically adapts to the system state through real-time feedback, fundamentally reducing the model's sensitivity to this parameter.

4.2 Deep Q-Learning with LSTM and MHA Mechanism

4.2.1 Model Architecture

After establishing MDP, we proposed a deep reinforcement learning algorithm DLSAM, as shown in Figure 3, to solve the multidimensional optimization problem of transportation task scheduling and resource allocation in the vehicle cargo matching scenario. This framework consists of two main parts: environment and agent.

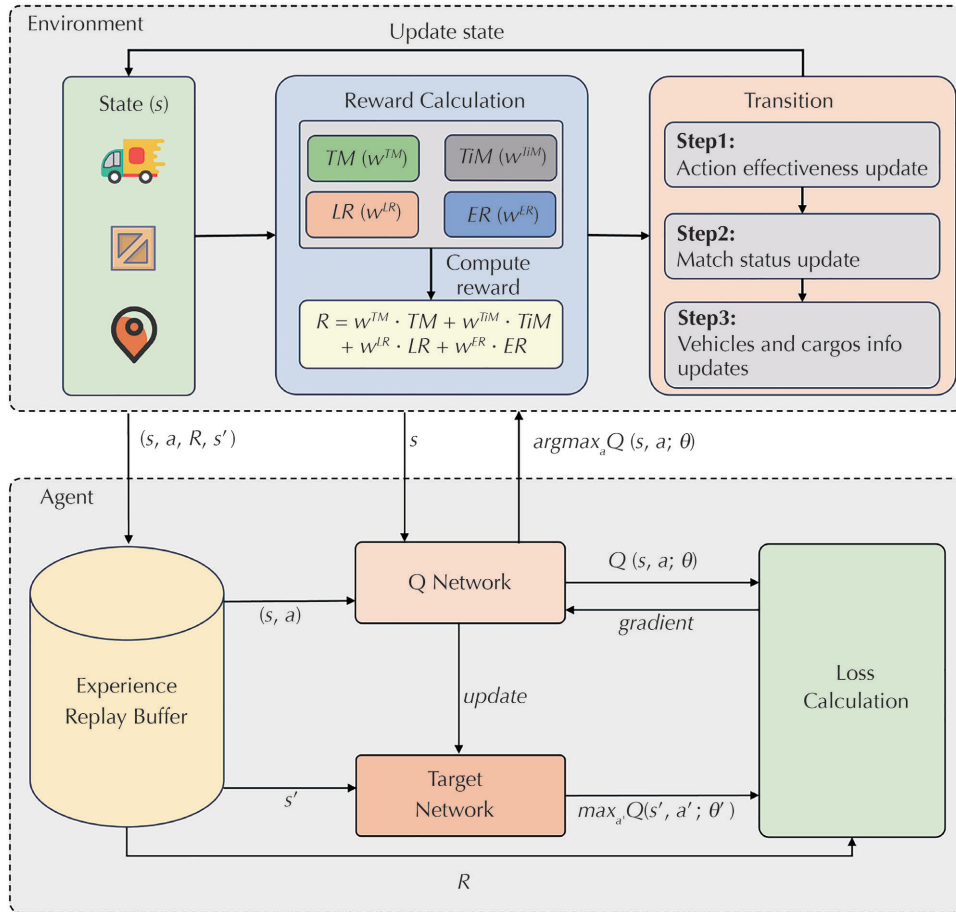


Figure 3. Deep Q-Learning with LSTM and MHA Mechanism.

First, the environmental component encompasses the state information of all vehicles, cargo, and locations. After an action is performed, the system updates the environmental state through the state transition module. Second, within the environment's reward module, a comprehensive reward value is calculated using multiple weighted criteria, such as time efficiency and the rate of empty travel.

Finally, the agent continuously optimizes the Q-network by utilizing experience replay and target network update strategies. This enables the agent to select the most effective scheduling strategy in each state, facilitating dynamic resource allocation and optimization.

In Figure 3, the input state of the environment module consists of dynamic information, including vehicles, cargo, and locations. In practical applications, this reflects the real-time status of various resources within the logistics delivery system, encompassing the locations of vehicles, their load capacities, the delivery demands of cargo, and geographic location data.

The algorithm employs a Deep Q-network (DQN) within the framework of reinforcement learning, integrating a Long Short-Term Memory (LSTM) network and a Multi-Head Attention (MHA) mechanism to optimize the matching process efficiently. During the initialization phase, the Q-network Q_θ and the target network $Q_{\theta'}$ are initialized with random weights. An experience replay buffer D is created to store samples generated throughout the matching process, ensuring that the training is not adversely affected by correlations between samples. Additionally, key hyperparameters, including the maximum number of training episodes E , the discount factor γ , the learning rate α , the exploration rate ϵ , and the exploration decay factor ϵ_{decay} , are configured to effectively guide the learning process.

During each training episode, the system begins in the initial state s_0 . At each time step t , actions for matching are chosen using an ϵ -greedy strategy. Specifically, the model randomly selects an action with a probability of ϵ , or, with a probability of $1 - \epsilon$, it selects the action that maximizes the output of the current Q-network.

$$a_t = \begin{cases} a_{\text{random}}, & \text{if } \phi < \epsilon \\ \operatorname{argmax}_a Q_\theta(s_t, a; \theta), & \text{otherwise} \end{cases} \quad (21)$$

During training episodes, the action a_t at time step t is determined according to specific rules. A random number ϕ is compared with the probability value ϵ to decide whether to select a random action a_{random} or the action that maximizes the output of the current Q-network, i.e., $\operatorname{argmax}_a Q_\theta(s_t, a; \theta)$. This mechanism helps to balance exploration and exploitation, thereby enhancing the overall performance of the system. The selected action a_t is used to match vehicle V_i with cargo C_j . After the matching is completed, the system receives an immediate reward R_t and transitions to the next state s_{t+1} .

The samples (s_t, a_t, r_t, s_{t+1}) generated during the matching process are stored in the experience replay buffer D . During the training process, the model randomly samples a batch from the buffer to compute the target values:

$$y_t = \begin{cases} r_t, & s_{t+1} \in S_{\text{terminal}} \\ r_t + \gamma \max_{a'} \hat{Q}_{\theta'}(s', a'; \theta), & s_{t+1} \notin S_{\text{terminal}} \end{cases} \quad (22)$$

This formula computes the target value during training process of the matching task. When the state s_{t+1} belongs to the terminal state set ($s_{t+1} \in S_{\text{terminal}}$), the target value y_t is equal to the immediate reward r_t .

Conversely, when the state s_{t+1} does not belong to the terminal state set ($s_{t+1} \notin S_{\text{terminal}}$), the target value y_t is computed as the immediate reward r_t plus the discounted future reward, which is represented by the discount factor γ multiplied by the maximum Q -value of all possible actions in state s_{t+1} . By considering both the immediate reward and the potential future value of states, this formula enhances the learning process and optimizes strategy development for the model.

The loss is computed using the Mean Squared Error (MSE), which is defined as:

$$L(\theta) = \frac{1}{N} \sum_{(s_t, a_t, r_t, s_{t+1}) \in D} (y_t - Q_\theta(s_t, a_t; \theta))^2 \quad (23)$$

Equation (23) defines the loss $L(\theta)$ utilizing the Mean Squared Error (MSE). During the training phase of the matching task, samples, denoted as (s_t, a_t, r_t, s_{t+1}) , are extracted from the experience replay buffer D . For each such sample, the discrepancy between the target value y_t and the corresponding predicted Q -value, $Q_\theta(s_t, a_t; \theta)$, is calculated. These squared discrepancies are then summed and subsequently averaged over the total number of samples N to compute the loss. This metric quantifies the discrepancy between the model's predicted Q -values and the target values, thereby guiding the iterative adjustment of the parameters θ to minimize this loss. Consequently, reducing the loss enhances the model's learning performance and optimizes its strategy.

The model updates its network parameters utilizing the gradient descent algorithm. Periodically, every C steps, the parameters of the Q -network are replicated to the target network, $Q_{\theta'}$. Furthermore, at the conclusion of each episode, the exploration rate ε is progressively decreased.

$$\varepsilon \leftarrow \max(\varepsilon_{\min}, \varepsilon \cdot \varepsilon_{\text{decay}}) \quad (24)$$

Equation (24) governs the adjustment of the exploration rate ε at the conclusion of each episode. During the training process, to facilitate the model's gradual transition towards exploiting nearoptimal strategies, the exploration rate is progressively reduced. Within this equation, $\varepsilon_{\min}$ represents the minimum exploration rate, while $\varepsilon_{\text{decay}}$ denotes the decay factor. Initially, the product of the current exploration rate ε and the decay factor $\varepsilon_{\text{decay}}$ is computed. This resultant value is subsequently compared against the minimum exploration rate $\varepsilon_{\min}$, and the maximum of these two values is designated as the new exploration rate. Such an approach prevents the exploration rate from decreasing excessively rapidly, thereby mitigating premature convergence to a suboptimal solution, while concurrently promoting a gradual reduction in exploration during later training stages. This mechanism enables the model to preferentially select more refined strategies.

4.2.2 Q-Network

To address the challenges of dynamic, temporal, and multi-objective optimization in the vehicle-cargo matching process, this study proposes a novel hybrid network architecture integrating Long Short-Term Memory (LSTM) and Multi-Head Attention (MHA) mechanisms. This architecture is utilized as the

Q-network and target network in the proposed DLSAM. The DLSAM introduces several innovations to overcome the limitations of traditional vehicle-to-cargo matching methods. The core of this hybrid network is a composite structure based on LSTM [33] and MHA [34], which comprises an encoder-decoder module, multi-layer LSTM networks, multi-head self-attention mechanisms, and activation functions.

Specifically, an encoder-decoder module is employed to process input information, enabling the extraction and integration of global features to predict near-optimal vehicle-to-cargo matching solutions. The LSTM network captures the temporal dynamics of vehicle and cargo states as well as historical matching records, effectively identifying long-term dependencies within the data sequences. Furthermore, the multi-head self-attention mechanism efficiently extracts complex dependencies between vehicles and cargo, such as competition among vehicles for the same cargo or the matching priority of a single vehicle for multiple cargoes. Finally, the Softmax function converts the model's output into a probability distribution, thereby enabling the strategic selection of near-optimal matching combinations.

To address the inherent complexity of vehicle-cargo matching, the proposed model is engineered to serve as the core architectural module for both the Q-network and the target network within the DQN framework. The architecture of this model is depicted in Figure 4.

This section offers a detailed introduction to the Input Module, Encoder-Decoder Module, and Output Module of the policy network.

Input: The input dataset for this study comprises vehicle information, cargo information, and associated location data. To quantify the matching relationships between vehicles and cargo, a matching state matrix, denoted as M , is defined with dimensions $n \times m$. Here, n and m denote the respective counts of vehicles and cargo items. Each element $M_{i,j}$ in this matrix represents the matching status between vehicle V_i and cargo C_j . Within this matrix, row indices i correspond to vehicles, and column indices j correspond to cargo items. This matrix enables the systematic management and analysis of vehicle-cargo matching relationships.

Encoder: Within the Q-network, the encoder transforms input state information into a featurerich, high-dimensional representation, thereby facilitating subsequent processing and analysis. The encoder comprises E layers (where $E = 2$), as illustrated. The input state information, denoted as $X_{\text{input}} = (C, V, M)$, is typically represented as a vector. Through linear transformations, X_{input} is projected into a higher-dimensional space. This projection facilitates feature extraction, thereby enhancing the model's representational capacity for the input data. Consequently, this transformation augments the model's expressive power, allowing for the capture of more subtle features and underlying patterns within the data. The specific linear projection is detailed in Equation (24).

$$X = W \cdot X_{\text{input}} + b \quad (25)$$

The input vectors undergo linear projections to generate the query (Q), key (K), and value (V) matrices, as defined by Equation (25).

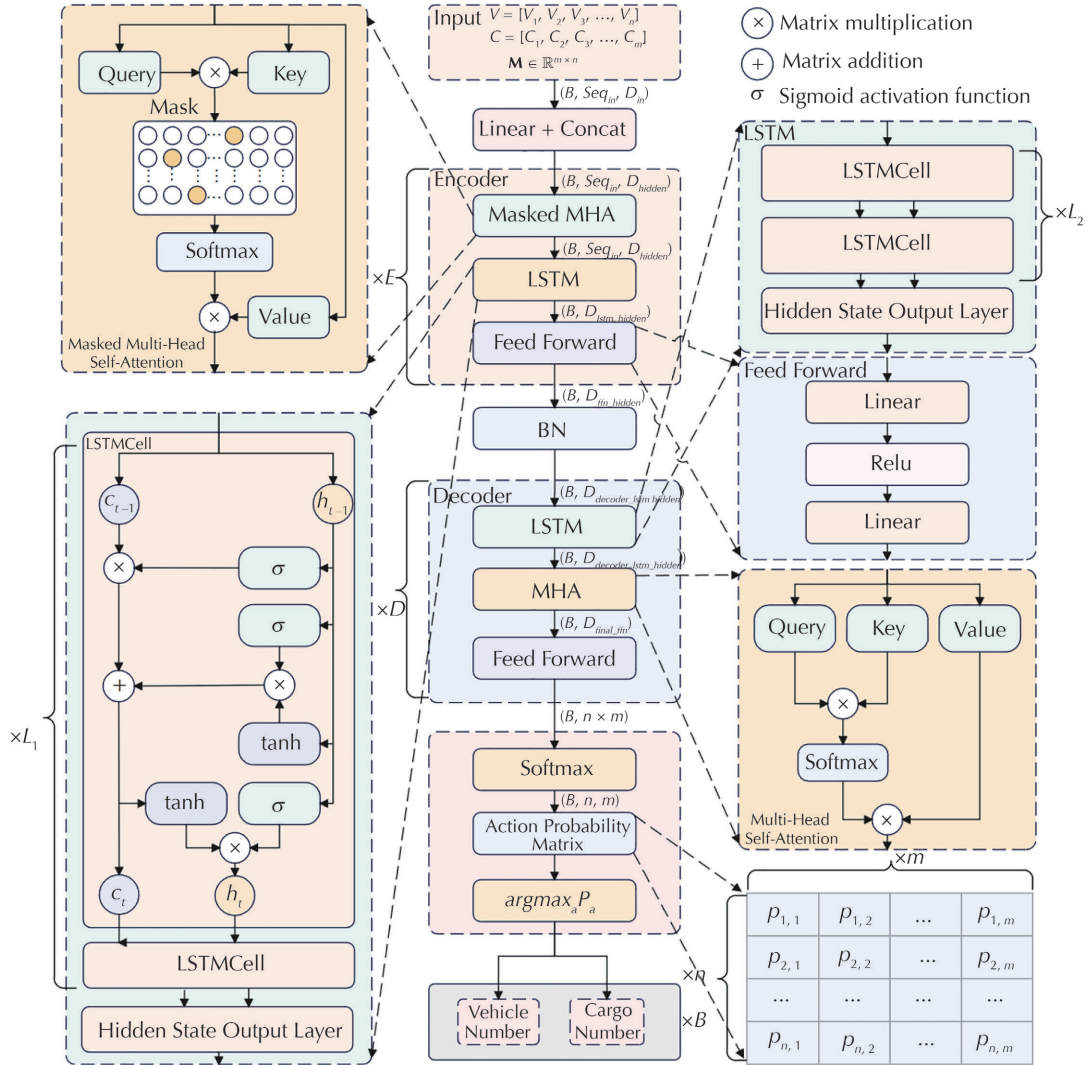


Figure 4. Encoder-Decoder Decision Model with LSTM and Multi-Head Self-Attention.

$$Q = X \cdot W^Q, \quad K = X \cdot W^K, \quad V = X \cdot W^V \quad (26)$$

Where, W^Q , W^K , and W^V denote trainable weight matrices that enable the effective projection of input features into their respective query, key, and value spaces.

The process commences with computing the dot product of the query matrix Q and the transposed key matrix K^T to derive initial attention scores. These scores are subsequently scaled by a factor related to the

dimension of the key vectors. The scaled scores are then normalized via the Softmax function to produce a probability distribution. Finally, this distribution is used to weight the value matrix V , thereby generating the final output.

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{Q \cdot K^T}{\sqrt{d_k}}\right) \cdot V \quad (27)$$

The parameter d_k denotes the dimensionality of the key vectors, and the term $\frac{1}{\sqrt{d_k}}$ functions as a scaling coefficient. This scaling is applied to the dot product results to mitigate the risk of excessively large values, thereby improving numerical stability.

The Multi-Head Attention (MHA) mechanism empowers the model to analyze complex interrelationships between vehicles and cargo from diverse perspectives. This is achieved by utilizing multiple independent attention heads, each focusing on different relational aspects or features. The outputs from these attention heads are subsequently concatenated to form a comprehensive feature representation, integrating information gleaned from the various perspectives. Finally, this consolidated representation is processed via a linear transformation to generate the ultimate output.

$$\text{MHA}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h) \cdot W^O \quad (28)$$

In this context, an individual attention head computes $\text{head}_i = \text{Attention}(Q_i, K_i, V_i)$; the matrix W^O is also a key component, representing the linear transformation applied to the aggregated output of all heads. The feature representation generated by the multi-head attention mechanism thus integrates information captured from diverse perspectives, yielding a more comprehensive and informative representation. This resulting enhanced representation serves as a robust input for subsequent temporal modeling tasks.

Subsequently, the output from the attention mechanism undergoes additional temporal modeling via multiple LSTM network layers. The specific computational formulation for the first layer of this multi-layer LSTM structure is presented as follows:

$$h_t, c_t = \text{LSTMCell}(\text{MHA}(Q, K, V)) \quad (29)$$

This equation details how the hidden state (h_t) and cell state (c_t) are updated within the LSTM cell, utilizing the output from the Multi-Head Attention mechanism. It should be noted that the input to the LSTM Cell is not only the attention output, but also includes the hidden state of the previous step. This is a simplified representation.

Finally, in Equation (30), the variable h_t denotes the output of the preceding layer.

$$h_{t+1}, c_{t+1} = \text{LSTMCell}(h_t, c_t, x_{t+1}) \quad (30)$$

Herein, h_t denotes the output of the preceding layer, which subsequently serves as input to the current LSTM cell. The LSTM cell then calculates the updated hidden state (h_{t+1}) and cell state (c_{t+1}).

Subsequently, the hidden state h_{t+1} from the final LSTM layer serves as input to a feedforward neural network. This additional processing step aims to extract and model more intricate nonlinear relationships within the data. The corresponding mathematical formulation is presented as follows:

$$z_1 = h_t^i \cdot W_1 + b_1 \quad (31)$$

$$z_2 = \text{ReLU}(z_1) \quad (32)$$

$$z_3 = z_2 \cdot W_2 + b_2 \quad (33)$$

These equations delineate how the output from the LSTM layer is processed. Initially, as shown in Equation (31), the hidden state h_t^i is multiplied by a weight matrix and augmented with a bias vector to yield z_1 . Subsequently, Equation (32) applies the ReLU activation function to z_1 to introduce nonlinearity, resulting in z_2 . Finally, Equation (33) details z_2 undergoing a further linear transformation (multiplication by another weight matrix and addition of another bias vector) to produce z_3 . This sequence of operations enables the model to learn more intricate patterns within the data.

To augment the model's adaptability to input features and its generalization capability, a Batch Normalization (BN) layer is introduced subsequent to the output from the feedforward neural network.

Decoder: The input to the decoder component comprises the features z_3 , previously processed by the feedforward neural network and the Batch Normalization (BN) layer. To further model dynamic changes in matching relationships between vehicles and cargo, the decoder employs an L_2 -stacked Long Short-Term Memory (LSTM) network. This network is specifically designed to capture temporal dependencies within historical matching records and vehicle states. The computational details of the multi-layer LSTM have been described previously.

The primary objective of this decoder is to dynamically capture changes in the matching relationships between vehicles and cargo, ultimately yielding a robust temporal feature representation. Within the decoder, the output from the final LSTM layer is subsequently inputted to the multi-head attention mechanism.

Within the decoder, the output from the final LSTM layer, $h_t^{(L_2)}$, serves as input to the multihead attention (MHA) mechanism. This allows for the derivation of contextual information most pertinent to the current decoding target. The MHA computation entails linear transformations to generate queries (Q), keys (K), and values (V), followed by relevance calculations performed independently across multiple attention heads. Finally, the outputs from these heads are concatenated and subjected to a further linear transformation, yielding the context-enhanced feature representation denoted as MHA_D .

Subsequently, a feedforward neural network is utilized to enable the model to effectively capture complex interrelationships among input features and convert them into feature representations optimized for subsequent processing. This transformation establishes a foundation for the ensuing decision-making process.

$$\text{FFN}(\text{MHA}_D) = (\text{ReLU}(\text{MHA}_D \cdot W_3 + b_3)) \cdot W_4 + b_4 \quad (34)$$

Output: The output from the preceding layer is processed via a Softmax layer within the context of vehicle and cargo matching scenarios. The model generates multiple potential matching combinations, with each combination being assigned a score. During the deep reinforcement learning process, the Softmax function allows the model to more effectively address multi-objective optimization problems. This function evaluates the complex interrelationships between vehicles and cargo, subsequently identifying the most suitable matching combination. Through this approach, the Softmax function not only facilitates efficient decision-making but also augments the model's adaptability to diverse input scenarios.

$$P_{i,j} = \text{Softmax}(\text{FFN}(\text{MHA}_D)) \quad (35)$$

Equation (35) details the application of the Softmax function, which transforms the neural network output, $\text{FFN}(\text{MHA}_D)$, into a probability distribution. Specifically, $\text{FFN}(\text{MHA}_D)$ represents the output from the final layer of the deep reinforcement learning model, comprising a series of scores. The Softmax function converts these scores into probabilities, denoted as $P_{i,j}$, such that their sum totals one. Each probability $P_{i,j}$ corresponds to a potential vehicle-cargo matching combination, wherein a higher probability signifies a greater likelihood that the corresponding combination represents the most suitable matching solution.

To provide a concise view of the computational cost of the proposed LSTM-MHA hybrid Q-network, we analyze the dominant terms under a sequence length T and feature dimension d with h attention heads ($d_k = d/h$). In one MHA layer, the linear projections to Q , K , and V incur $\mathcal{O}(T \cdot d^2)$, while the attention computation $\text{Softmax}(Q \cdot K^T / \sqrt{d_k})V$ introduces the quadratic interaction across time steps, yielding $\mathcal{O}(T^2 \cdot d)$; thus the overall time complexity is $\mathcal{O}(Td^2 + T^2d)$ and the main extra memory comes from storing attention scores of size $T \times T$ (up to a constant factor of h). In contrast, an LSTM layer scales linearly with the sequence length, with time complexity $\mathcal{O}(T(d \cdot d_h + d_h^2))$ for hidden size d_h and training memory on the order of $\mathcal{O}(T \cdot d_h)$. Consequently, the computational bottleneck typically arises from the MHA term when T grows, and the proposed framework remains practical for dynamic matching by constraining the effective history window (limiting T) and restricting the candidate set (e.g., Top-K filtering), which bounds the number of tokens processed per decision and preserves real-time feasibility.

4.2.3 Training Algorithm

Algorithm 1 details the DLSAM training process for vehicle-cargo matching. This process incorporates the Q-network update mechanism, the utilization of an experience replay buffer, and an exploration rate decay strategy.

Algorithm 1. DLSAM for Vehicle-Cargo Matching Problem.

Initialize:

Initialize Q-network $Q(s, \alpha; \theta)$ and target network $Q'(s, \alpha; \theta')$

Initialize experience replay buffer D

Set hyperparameters: maximum episodes E , discount factor γ , learning rate α , exploration rate ϵ , exploration decay factor ϵ_{decay}

Input:

Vehicle dataset: Contains vehicle information (e.g., ID, capacity, type, location)

Cargo dataset: Contains cargo information (e.g., ID, weight, volume, type, destination)

for $e = 1 \rightarrow E$ **do**

Obtain initial environment state S_0

for $T = 1 \rightarrow N$ **do**

if with probability ϵ (exploration) **then**

└ Select action a_t randomly

else

└ Select $a_t = \operatorname{argmax}_a Q(s, \alpha; \theta)$

Execute action q_t , observe reward r_t and next state S_{t+1}

Store transition (S_t, q_t, r_t, S_{t+1}) in buffer D

Sample mini-batch (s_i, a_i, r_i, s'_i) from D

Compute target Q-values, refer to equation (22)

Calculate loss $L(\theta)$, refer to equation (23)

Update Q-network parameters:

└ $\theta \leftarrow \theta - \alpha \cdot \nabla_{\theta} L(\theta)$

if Reward reaches threshold **then**

└ Save current model parameters

if Every c steps **then**

└ Synchronize target network: $\theta' \leftarrow \theta$

└ Decay exploration rate: ϵ , refer to equation (24)

Output: Vehicle-cargo matching results (vehicle IDs paired with cargo IDs)

5. EXPERIMENTS

This section presents a detailed discussion of the practical application and experimental results of the DLSAM algorithm applied to vehicle-cargo matching. The performance of the proposed method is evaluated across diverse scenarios utilizing multiple datasets. Specifically, the algorithm's advantages are demonstrated concerning key metrics such as empty vehicle rate, time matching accuracy, type matching accuracy, load rate, and matching speed. Moreover, a comparative analysis against conventional methods affirms the efficacy and novelty of the proposed approach.

5.1 Benchmark Data

In this paper, the detailed data of vehicles and cargoes were obtained from the logistics industry big data platform, and the data were cleaned up and organized to obtain specific attributes such as type, weight, volume, origin, destination, matching start time and cut-off time respectively. The vehicle and cargo datasets, along with the algorithm implementation code, have been made publicly available under an open-source license (accessible at: <https://github.com/laiwang1999/DLSAM>). The experimental dataset comprises two distinct files:

(1) Cargo data, as shown in Table 3. The cargo data includes attributes such as cargo type, weight, volume, origin and destination, and expected delivery time window. The selected cargo types encompass fresh produce, machinery parts, general merchandise, and bulk goods such as sand and gravel, reflecting the broad coverage of cargo categories. The weight range is set between 4,000 kg and 10,000 kg, while the volume range is set between 20 m³ and 60 m³ to simulate the realistic weight and volume of cargo. In order to simplify the transportation process and algorithm training process, this study represents the starting and ending points of goods as coordinate points on a two-dimensional plane based on real logistics data. This approach not only helps to simplify the complex problem of transportation route planning but also provides a clear and well-defined foundation for our analysis and algorithm design. For the time window of the cargo, we define it as an interval with two endpoints, representing the start and end times, denoted as $(C_j^{\text{start_time}}, C_j^{\text{end_time}})$. Here, $C_j^{\text{start_time}}$ represents the start time of the time window, and $C_j^{\text{end_time}}$ represents its end time. Within this time window, the cargo is in a ready state, during which operations such as loading, transportation, or other activities can take place, and the setting of the time window follows a normal distribution around peak hours.

Table 3. Cargoes Data Sheet.

Type	Quality	Volume	SP	EP	Time
Machine parts	10000	50	(48, 44)	(44, 16)	(5, 20)
Fresh fruits and vegetables	6809	27	(9, 38)	(37, 6)	(5, 19)
...	...	...	...	...	...
Hardware Machinery	7899	48	(27, 42)	(32, 24)	(8, 19)

(2) Vehicle data, as shown in Table 4, includes attributes such as vehicle type, load capacity, volume capacity, location, and working time intervals. The vehicle types include flatbed trucks, refrigerated trucks, and low-sided trucks, reflecting the wide range of vehicle coverage in our study. The load capacity is generated within the range of 8,000 kg to 12,000 kg, and the volume capacity is set between 30 m³ and 80 m³ to simulate real-world vehicle capacities. To simplify the complexity of vehicle scheduling, we used the same simplification method as the cargo data. This method not only helps simplify complex route scheduling problems, but also provides a clear foundation for our analysis and algorithm design. The working time intervals of vehicles are defined as $(V_i^{\text{start_time}}, V_i^{\text{end_time}})$, where $V_i^{\text{start_time}}$ represents the start time of the working interval, and $V_i^{\text{end_time}}$ represents the end time. During this time interval, vehicles are in a working state, capable of performing operations such as loading and transportation, also following a normal distribution.

Table 4. Vehicles Data Sheet.

Type	Load	Volume	Position	Time
Flat car	10937	45	(10, 42)	(8, 24)
Refrigerated car	11641	53	(40, 23)	(19, 23)
...	...	...	...	...
Low barrier car	9619	53	(49, 5)	(10, 22)

To address the vehicle-cargo matching problem, we selected seven datasets, as shown in Table 5. Each dataset features varying numbers of vehicles and cargo, with the map size progressively increasing as the scale grows. These datasets are designed to simulate logistics networks and transportation scenarios of different scales. We used 80% of each dataset as the training set and the remaining 20% as the test set. The selection of datasets covers a wide range of scenarios, from small-scale to large-scale cases, ensuring the broad applicability of the research findings.

Table 5. Parameter table of vehicle cargo matching experimental datasets.

Datasets	Number of vehicles	Number of cargos	Map size
CV50	50	50	50 × 50
CV100	100	100	100 × 100
CV500	500	500	200 × 200
CV1k	1000	1000	300 × 300
CV2k	2000	2000	500 × 500
CV5k	5000	5000	1000 × 1000

All experiments were conducted on a server equipped with an Intel(R) Xeon(R) Platinum 8352V CPU featuring 12 virtual CPUs (vCPUs) with a clock speed of 2.10 GHz, and an NVIDIA RTX 4090 GPU with 24 GB of memory.

5.2 Ablation Experiment

To ascertain the efficacy of the individual modules in DLSAM, a series of ablation experiments were designed to evaluate the impact of removing or replacing key components on matching performance. These experiments illuminate the contribution of each module to the overall model. The DLSAM variants used in the ablation study are summarized in Table 6, which delineates the specific modifications made to each variant—such as removing the LSTM module or replacing the attention mechanism. This summary provides foundational support for the subsequent performance analysis. Detailed experimental settings are also provided in Table 6.

Table 6. DLSAM Experimental Variants and Descriptions.

Model Variant	Description
DLSAM-NL	Remove the LSTM in DLSAM.
DLSAM-NSA	Remove all attention module in DLSAM.
DLSAM-FC	Replace the entire module with fully connected layers.
DLSAM-BiLSTM	Replace the LSTM in DLSAM with BiLSTM.
DLSAM-GRU	Replace the LSTM in DLSAM with Gated Recurrent Unit.
DLSAM-CAM	Replace the multi-head attention module in DLSAM with channel attention.
DLSAM-SAM	Replace the multi-head attention module in DLSAM with spatial attention.
DLSAM-NoMaskEncoder (NME)	Remove the mask from the MHA module in the DLSAM encoder.

To validate the contribution of each module in DLSAM to the performance of vehicle-cargo matching, we designed and conducted a series of ablation experiments. In this chapter, we evaluate the impact of sequentially removing or replacing key components of DLSAM on the overall matching performance. By comparing the performance of various model variants, we can identify which modules are critical for improving matching accuracy, loading rate, unloading rate, and overall operational efficiency. The results of the ablation experiments provide valuable insights for optimizing DLSAM, helping us better understand the specific role of each module in different task environments.

Table 7 summarizes the performance of DLSAM and its various ablation variants on six key metrics across four datasets of different scales (CV50, CV100, CV500, CV1k). The results demonstrate that the DLSAM algorithm outperforms other comparative models across all metrics, particularly excelling in key indicators such as type matching degree and load rate degree. In the experiments, spatial attention

Table 7. Comparison of Performance in Vehicle-Cargo Matching Tasks.

Dataset	Target	DLSAM-NL	DLSAM-NSA	DLSAM-FC	DLSAM-BiLSTM	DLSAM-GRU	DLSAM-CAM	DLSAM-SAM	DLSAM-NME	DLSAM
CV50	TM	0.585 ± 0.03	0.556 ± 0.04	0.897 ± 0.02	0.450 ± 0.03	0.440 ± 0.03	0.424 ± 0.02	0.636 ± 0.02	0.633 ± 0.02	0.646* ± 0.02
	LR	0.729 ± 0.02	0.768 ± 0.02	0.845 ± 0.01	0.588 ± 0.01	0.577 ± 0.01	0.584 ± 0.01	0.831 ± 0.01*	0.812 ± 0.01	0.821 ± 0.01
	ER	0.307 ± 0.02	0.316 ± 0.02	0.389* ± 0.02	0.337 ± 0.01	0.537 ± 0.01	0.522 ± 0.01	0.343 ± 0.01	0.356 ± 0.01	0.285 ± 0.01
	TiM	0.819 ± 0.03	0.825 ± 0.02	0.390 ± 0.04	0.664 ± 0.02	0.668 ± 0.02	0.696 ± 0.02	0.888* ± 0.01	0.775 ± 0.02	0.970 ± 0.01
	RT	0.010s ± 0.00s	0.010s ± 0.00s	0.010s ± 0.00s	0.010s ± 0.00s	0.010s ± 0.00s	0.010s ± 0.00s	0.010s ± 0.00s	0.010s ± 0.00s	0.010s ± 0.00s
	R	0.853 ± 0.02	0.852 ± 0.02	0.806 ± 0.03	0.895* ± 0.01	0.877 ± 0.02	0.894 ± 0.01	0.879 ± 0.02	0.810* ± 0.03	0.930 ± 0.02
CV100	TM	0.575 ± 0.03	0.593 ± 0.03	0.741 ± 0.02	0.857* ± 0.02	0.834 ± 0.02	0.851 ± 0.02	0.697 ± 0.03	0.810 ± 0.02	0.884 ± 0.02
	LR	0.755 ± 0.02	0.733 ± 0.02	0.714 ± 0.02	0.804 ± 0.01	0.810* ± 0.01	0.791 ± 0.02	0.804 ± 0.01	0.774 ± 0.02	0.815 ± 0.01
	ER	0.324 ± 0.02	0.287 ± 0.01	0.176 ± 0.01	0.367 ± 0.01	0.367 ± 0.01	0.269 ± 0.01	0.376 ± 0.01	0.356 ± 0.01	0.365 ± 0.01
	TiM	0.880 ± 0.01	0.854* ± 0.02	0.355 ± 0.03	0.684 ± 0.02	0.689 ± 0.02	0.697 ± 0.02	0.667 ± 0.02	0.688 ± 0.02	0.707 ± 0.02
	RT	0.060s* ± 0.01s	0.060s* ± 0.01s	0.050s ± 0.00s	0.130s ± 0.01s	0.060s ± 0.01s	0.080s ± 0.01s	0.100s ± 0.01s	0.070s ± 0.01s	0.070s ± 0.01s
	R	0.859 ± 0.02	0.874* ± 0.01	0.858 ± 0.02	0.864 ± 0.02	0.871 ± 0.01	0.857 ± 0.02	0.821 ± 0.02	0.810 ± 0.02	0.878 ± 0.01
CV500	TM	0.569 ± 0.02	0.579 ± 0.02	0.587* ± 0.02	0.643 ± 0.02	0.633 ± 0.02	0.646 ± 0.02	0.696 ± 0.02	0.656 ± 0.02	0.693* ± 0.02
	LR	0.722 ± 0.01	0.741 ± 0.01	0.718 ± 0.02	0.879 ± 0.01	0.878 ± 0.01	0.876 ± 0.01	0.852 ± 0.01	0.875 ± 0.01	0.874 ± 0.01
	ER	0.303 ± 0.01	0.314 ± 0.01	0.285 ± 0.01	0.293 ± 0.01	0.289 ± 0.01*	0.296 ± 0.01	0.321 ± 0.01	0.316 ± 0.01	0.296 ± 0.01
	TiM	0.854 ± 0.01	0.828* ± 0.02	0.826 ± 0.02	0.809 ± 0.02	0.809 ± 0.02	0.801 ± 0.02	0.803 ± 0.02	0.790 ± 0.02	0.808 ± 0.02
	RT	0.220s ± 0.02s	0.210s ± 0.02s	0.230s ± 0.02s	0.300s ± 0.02s	0.220s ± 0.02s	0.220s ± 0.02s	0.240s ± 0.02s	0.200s ± 0.01s	0.200s ± 0.01s
	R	0.858 ± 0.02	0.851 ± 0.02	0.865 ± 0.02	0.904 ± 0.01	0.903* ± 0.01	0.900 ± 0.01	0.885 ± 0.01	0.847 ± 0.02	0.912 ± 0.01
CV1k	TM	0.665 ± 0.02	0.669 ± 0.02	0.601 ± 0.03	0.698* ± 0.02	0.694 ± 0.02	0.692 ± 0.02	0.625 ± 0.03	0.685 ± 0.02	0.710 ± 0.02
	LR	0.792 ± 0.01	0.784 ± 0.01	0.773 ± 0.01	0.784 ± 0.01	0.782 ± 0.01	0.777 ± 0.01	0.775 ± 0.01	0.776 ± 0.01	0.789* ± 0.01
	ER	0.337 ± 0.01	0.354 ± 0.02	0.370 ± 0.01	0.305 ± 0.01	0.309 ± 0.01	0.304 ± 0.01	0.370 ± 0.01	0.356 ± 0.01	0.300 ± 0.01
	TiM	0.672* ± 0.02	0.676 ± 0.02	0.648 ± 0.02	0.619 ± 0.02	0.617 ± 0.02	0.626 ± 0.02	0.665 ± 0.02	0.645 ± 0.02	0.636 ± 0.02
	RT	0.330s ± 0.02s	0.300s ± 0.01s	0.300s ± 0.01s	0.350s ± 0.02s	0.340s ± 0.02s	0.340s ± 0.02s	0.360s ± 0.02s	0.320s* ± 0.02s	0.320s* ± 0.02s
	R	0.835 ± 0.02	0.821 ± 0.02	0.792 ± 0.02	0.843* ± 0.01	0.842 ± 0.01	0.843* ± 0.01	0.800 ± 0.02	0.773 ± 0.02	0.857 ± 0.01
Total	Total (Opt/Subopt)	3-9	3-7	6-5	2-5	1-3	1-3	2-3	2-1	12-4

Opt-Subopt represent the number of optimal and suboptimal values achieved by this algorithm across all metrics and datasets.
* denotes suboptimal (second-best) values. Bold values indicate the best (optimal) mean performance.

(SAM) and channel attention (CAM) were used as replacements for the multi-head attention mechanism (resulting in DLSAM-SAM and DLSAM-CAM, respectively). While these alternatives achieved comparable performance on smaller datasets (e.g., CV100), they generally fell short of the original multi-head attention mechanism. For example, on the CV50 dataset, DLSAM achieved a TM of 0.646, notably higher than the 0.636 of spatial attention and the 0.424 of channel attention. This result highlights the superior ability of the multi-head attention mechanism to capture global features and critical patterns in complex matching relationships.

Moreover, when the LSTM module or the multi-head attention mechanism was removed (resulting in DLSAM-NL and DLSAM-NSA, respectively), the algorithm exhibited significant declines in TM and LR, further validating the importance of these modules for capturing temporal dependencies and extracting global features. Additionally, the DLSAM-NME variant, which removed the mask from the Multi-Head Attention module in the encoder, also showed consistently poorer performance compared to the standard DLSAM across most metrics and datasets, particularly in the overall reward (R) and often in empty running rate (ER). This indicates that the masking mechanism is crucial for the encoder to effectively focus attention on relevant features within the input state, thereby improving matching quality. Overall, DLSAM achieved the best or second-best performance in most experiments, far exceeding other models. This outcome underscores the irreplaceable role of the Multi-Head Attention mechanism (including its masking strategy) and the effectiveness of the module combination in the DLSAM design.

This superior performance can be attributed to the synergistic integration of temporal modeling and global relational reasoning. Specifically, the LSTM module enables the agent to maintain a coherent memory of dynamic system states over time—such as vehicle availability, cargo arrival patterns, and historical matching outcomes—thereby supporting informed decisions in non-stationary environments. Meanwhile, the masked multi-head attention mechanism allows the model to simultaneously evaluate all feasible vehicle–cargo pairs while respecting problem-specific constraints (e.g., type compatibility, time windows), effectively filtering out invalid matches and prioritizing high-value assignments. Together, these components empower DLSAM to achieve not only higher matching accuracy and loading efficiency but also lower operational costs, as evidenced by its consistently strong performance in reward maximization and empty running rate reduction—especially in large-scale and complex scenarios such as CV500 and CV1k.

To further investigate the performance of vehicle-cargo matching, we conducted a series of critical ablation experiments. These experiments were designed to precisely evaluate the contribution of each component in DLSAM, identifying the parts that play a crucial role in its accuracy, convergence speed, and stability. The following image (Figure 5) visually illustrates the training process of the ablation experiments. Each set of experiments was conducted in training environments of increasing complexity, ranging from CV50 to CV1k. After 2,000 training episodes, we were able to observe the performance of various strategies, providing deeper insights into their strengths and weaknesses under different conditions.

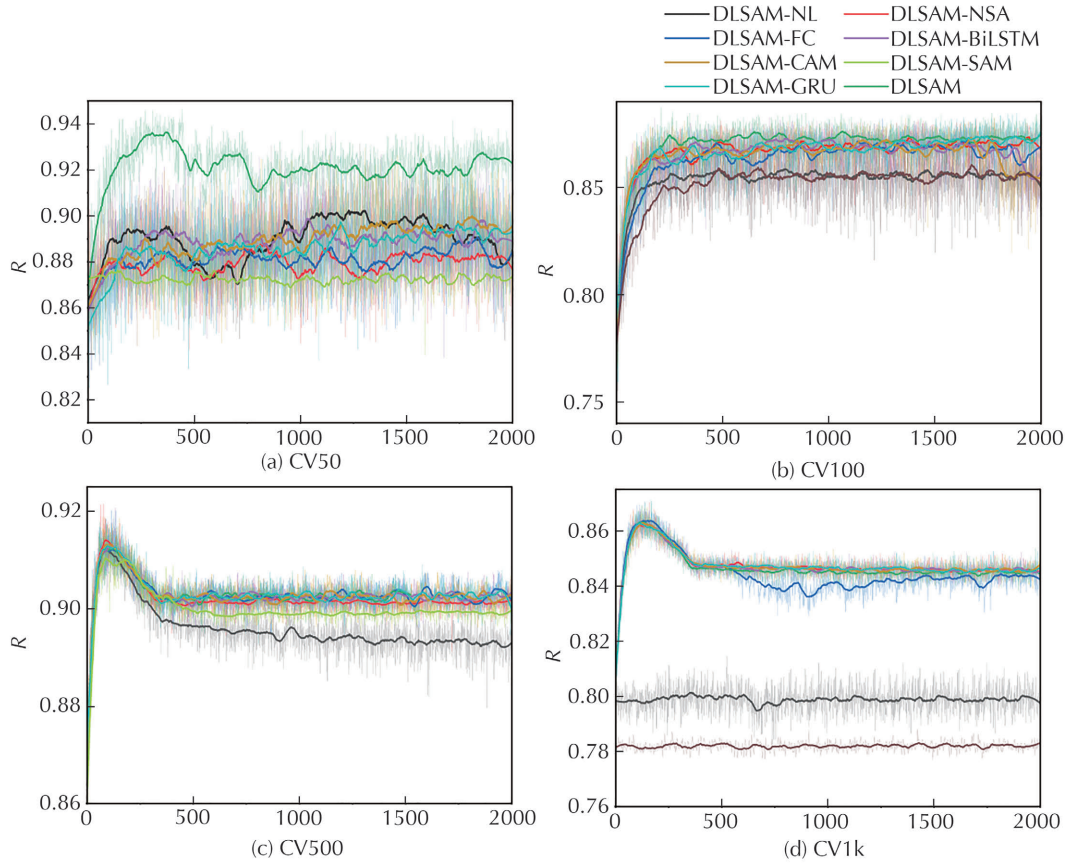


Figure 5. Process Diagram of Ablation Experiment Training.

Figure 5 illustrates the training process for the ablation experiments, which aim to evaluate the impact of individual components in DLSAM on vehicle-cargo matching performance. By sequentially removing or replacing specific modules within DLSAM, we identified the parts that are critical to its accuracy, convergence speed, and stability. Each experiment was conducted in training environments of increasing complexity, ranging from CV50 to CV1k.

The experiments were trained for 2,000 episodes, and the performance of different strategies was observed. Results from CV50 and CV100 demonstrate that the DLSAM strategy significantly outperformed other strategies in terms of initial convergence speed and final matching accuracy. In the more complex environments of CV500 and CV1k, while the convergence of all strategies tended to stabilize, DLSAM consistently maintained a high matching accuracy, highlighting its robustness in more challenging scenarios.

In contrast, DLSAM-NL showed poorer performance, particularly in more complex environments (CV500 and CV1k), exhibiting higher volatility and lower matching accuracy. This indicates that the standalone application of the basic strategy is insufficient to handle complex scenarios effectively.

5.3 Comparative Experiment

In the comparative experiments, five algorithms were selected to evaluate the performance of the proposed model. These include three baseline algorithms: Policy Gradient (PG), Proximal Policy Optimization (PPO), and DQN, as well as two existing algorithms: DBN and IDBN. The specific parameter configurations for each baseline algorithm are presented in Table 8.

Table 8. Comparison Experiment Parameter Settings Table for Vehicle-Cargo Matching.

Algorithm	Parameter Settings
DQN	learning rate = 1e-2, hidden size = 128, gamma = 0.85, optimizer = Adam, num episodes = 2000, exploration rate = 0.01
PPO	actor learning rate = 1e-3, critic learning rate = 1e-2, hidden size = 128, gamma = 0.98, optimizer = Adam, num episodes = 2000, exploration rate = 0.01
PG	learning rate = 5e-2, hidden size = 128, gamma = 0.75, optimizer = Adam, num episodes = 2000, exploration rate = 0.01
DBN & IDBN	Number of experts = 5, consistency threshold < 0.1, number of time slices = 4, the weight of attribute matching degree = 0.5
DLSAM	learning rate = 1e-3, hidden size = 128, gamma = 0.98, optimizer = Adam, num episodes = 2000, exploration rate = 0.01

As shown in Table 8, we evaluated the performance of the proposed DLSAM model and all comparative algorithms on six datasets. For each test set, 100 validation results were generated, and their average was calculated. For the reinforcement learning algorithms among the comparative models, testing was conducted using the model that achieved the optimal objective value during training. For the DBN and IDBN algorithms, the original methods proposed by their authors were used without modification.

The performance of our model and the comparative algorithms across all datasets is summarized in a table, using six metrics to evaluate differences between the algorithms: Type Matching Degree (TM), Time Matching Degree (TiM), Load Rate (LR), Empty Vehicle Rate (ER), Runtime (RT), and the Average Reward of Successful Vehicle-Cargo Matching (R).

The comprehensive experimental results demonstrate that the proposed DLSAM algorithm outperforms other algorithms in overall performance, exhibiting exceptional robustness and adaptability. In comparisons across all six datasets, DLSAM achieved optimal or near-optimal results on most key metrics. Notably, on large-scale datasets (e.g., CV500 and CV2k), DLSAM showed significant advantages in critical performance metrics such as Type Matching Accuracy (TMR), Runtime (RT), and Total Reward (TR).

For instance, on the CV500 dataset, DLSAM achieved a TiM of 0.808 and a total reward (TR) of 0.912, both of which were the highest among all algorithms. Similarly, on the CV2k dataset, DLSAM excelled in important metrics such as runtime (RT) and total reward (TR), with an RT of just 0.60 seconds and a R of 0.857, significantly surpassing the performance of other algorithms. These results indicate that while DLSAM does not exhibit a significant advantage in time efficiency on small-scale datasets, it demonstrates substantial superiority on larger-scale data. Furthermore, its ability to effectively maximize cumulative rewards makes it particularly well-suited for meeting the demands of dynamic vehicle-cargo matching tasks.

In contrast, the performance of other algorithms was relatively weaker. Traditional baseline algorithms, such as PG, PPO, and DQN, failed to achieve comparable performance to DLSAM on most datasets, particularly on large-scale datasets, where they exhibited significant volatility and instability. Although the IDBN algorithm demonstrated some competitiveness on smaller datasets (e.g., CV50), its performance was less stable on larger datasets (e.g., CV500 and CV2k) compared to DLSAM. The DBN algorithm, on the other hand, showed inconsistent performance across multiple datasets, making it difficult to demonstrate sustained advantages.

5.4 Generalization Research

In this section, we conduct a detailed analysis of the generalization performance of DLSAM in the vehicle-cargo matching problem. To evaluate the effectiveness of the DLSAM method across different datasets, we utilized five datasets of varying scales: CV50, CV100, CV500, CV1k, and CV2k. The evaluation focused on four key metrics: Type Matching Rate (TM), Time Matching Rate (TiM), Empty Running Rate (ER), and Loading Rate (LR). The experiments are divided into two parts: (1) training and testing the model separately on each dataset to obtain the optimal results, and (2) training the model on the CV5k dataset and testing it on the other datasets to evaluate its generalization performance. Table 9 presents a comparison of the optimal results and generalization results across four key metrics on different datasets (CV50, CV100, CV500, CV1k, CV2k). The table also includes the “Gap,” which represents the difference between the generalization results and the optimal results, providing a measure of the model’s generalization capability.

The data presented in Table 10 collectively demonstrate that the DLSAM algorithm exhibits robust and stable generalization performance. This is evidenced by the consistently small gap values across the different metrics and datasets. For instance, the Type Matching Rate (TM) shows minimal discrepancies between optimal and generalization results, with gaps typically ranging from 0% to 2%. Similarly, the Loading Rate (LR) and Time Matching Rate (TiM) also show strong alignment, with gaps generally not exceeding 2%, and often being 0% or 1%, particularly for larger datasets like CV1k and CV2k. The Empty Running Rate (ER) also reflects stable generalization, with gaps being 0% for the largest datasets and a maximum of 3% for the CV50 dataset. These findings underscore DLSAM’s proficiency in maintaining its performance when applied to datasets different from its primary training set, thereby validating its scalability and adaptability, especially in large-scale logistics scenarios.

Table 9. Comparison of Performance in Vehicle-Cargo Matching Tasks.

Dataset	Target	PG	PPO	DQN	DBN	IDBN	DLSAM
CV50	TM	0.250 ± 0.03	0.558 ± 0.04	0.650 ± 0.02	0.700 ± 0.00	0.690* ± 0.01	0.646 ± 0.02
	LR	0.751 ± 0.02	0.826 ± 0.02	0.826 ± 0.01	0.630 ± 0.01	0.650 ± 0.01	0.821 ± 0.01
	ER	0.451 ± 0.03	0.630 ± 0.03	0.320* ± 0.01	0.460 ± 0.00	0.530 ± 0.01	0.285 ± 0.01
	TiM	0.360 ± 0.05	0.993 ± 0.00	0.935 ± 0.01	0.290 ± 0.01	0.300 ± 0.01	0.970* ± 0.01
	RT	0.010 ± 0.00	0.030 ± 0.01	0.010 ± 0.00	0.200 ± 0.01	0.620 ± 0.02	0.010 ± 0.00
	R	0.847 ± 0.03	0.870 ± 0.02	0.890* ± 0.02	0.853 ± 0.01	0.860 ± 0.01	0.930 ± 0.02
CV100	TM	0.617 ± 0.03	0.305 ± 0.04	0.662* ± 0.02	0.565 ± 0.01	0.570 ± 0.01	0.884 ± 0.02
	LR	0.960 ± 0.01	0.735 ± 0.03	0.837* ± 0.02	0.529 ± 0.01	0.549 ± 0.01	0.815 ± 0.01
	ER	0.379 ± 0.02	0.454 ± 0.02	0.372* ± 0.01	0.366 ± 0.01	0.366 ± 0.01	0.365 ± 0.01
	TiM	0.137 ± 0.03	0.392 ± 0.04	0.685* ± 0.02	0.251 ± 0.02	0.273 ± 0.02	0.707 ± 0.02
	RT	0.090 ± 0.01	0.060 ± 0.00	0.070* ± 0.01	0.720 ± 0.02	0.710 ± 0.03	0.070* ± 0.01
	R	0.720 ± 0.03	0.866* ± 0.02	0.821 ± 0.02	0.760* ± 0.02	0.780 ± 0.02	0.879 ± 0.01
CV500	TM	0.745 ± 0.02	0.588 ± 0.04	0.631 ± 0.03	0.618 ± 0.01	0.650 ± 0.01	0.693* ± 0.02
	LR	0.730 ± 0.03	0.808 ± 0.02	0.853* ± 0.01	0.583 ± 0.01	0.560 ± 0.01	0.874 ± 0.01
	ER	0.363 ± 0.02	0.362 ± 0.02	0.304* ± 0.01*	0.366 ± 0.01	0.590 ± 0.01	0.296 ± 0.01
	TiM	0.907* ± 0.02	0.920 ± 0.01	0.800 ± 0.03	0.318 ± 0.02	0.150 ± 0.02	0.808 ± 0.02
	RT	0.250 ± 0.03	0.240 ± 0.03	0.180 ± 0.01	1.950 ± 0.22	1.820 ± 0.12	0.200* ± 0.02
	R	0.864 ± 0.02	0.857 ± 0.02	0.885* ± 0.01	0.785 ± 0.02	0.804 ± 0.02	0.912 ± 0.01
CV1k	TM	0.548 ± 0.03	0.505 ± 0.04	0.610 ± 0.02*	0.600 ± 0.01	0.610* ± 0.01	0.710 ± 0.02
	LR	0.812* ± 0.02	0.815 ± 0.01	0.779 ± 0.02	0.560 ± 0.01	0.580 ± 0.01	0.789 ± 0.01
	ER	0.370 ± 0.02	0.356* ± 0.02	0.374* ± 0.01	0.560 ± 0.01	0.560 ± 0.01	0.300 ± 0.01
	TiM	0.631* ± 0.02	0.600 ± 0.03	0.655 ± 0.02	0.140 ± 0.01	0.150 ± 0.01	0.636 ± 0.02
	RT	0.350 ± 0.04	0.330 ± 0.03	0.310 ± 0.02	3.700 ± 0.21	3.600 ± 0.31	0.320* ± 0.02
	R	0.787 ± 0.02	0.778 ± 0.02	0.792* ± 0.02	0.530 ± 0.02	0.538 ± 0.02	0.857 ± 0.01
CV2k	TM	0.330 ± 0.04	0.512 ± 0.04	0.658 ± 0.02	0.618 ± 0.01	0.612 ± 0.01	0.630* ± 0.02
	LR	0.784* ± 0.02	0.812 ± 0.01	0.788 ± 0.02	0.583 ± 0.01	0.583 ± 0.01	0.776 ± 0.02
	ER	0.680 ± 0.03	0.356 ± 0.02	0.348* ± 0.01	0.365 ± 0.01	0.361 ± 0.01	0.275 ± 0.01
	TiM	0.614 ± 0.03	0.608 ± 0.03	0.646 ± 0.02*	0.318 ± 0.02	0.320 ± 0.02	0.663 ± 0.02
	RT	0.550 ± 0.05	0.520 ± 0.03	0.540* ± 0.04	6.330 ± 0.30	11.300 ± 0.40	0.600 ± 0.05
	R	0.757 ± 0.02	0.782 ± 0.02	0.804 ± 0.02	0.784 ± 0.02	0.812* ± 0.02	0.857 ± 0.01

Table 9. *Continued.*

Dataset	Target	PG	PPO	DQN	DBN	IDBN	DLSAM
CV5k	TM	0.140 ± 0.03	0.508 ± 0.04	0.657 ± 0.02	0.650 ± 0.01	0.660* ± 0.01	0.730 ± 0.02
	LR	0.902 ± 0.01	0.810 ± 0.02	0.782 ± 0.02	0.600 ± 0.01	0.590 ± 0.01	0.842* ± 0.01
	ER	0.359 ± 0.02	0.355 ± 0.02	0.367* ± 0.01	0.590 ± 0.01	0.590 ± 0.01	0.358* ± 0.01
	TiM	0.986 ± 0.01	0.605 ± 0.04	0.673 ± 0.03	0.140 ± 0.01	0.140 ± 0.01	0.748* ± 0.02
	RT	1.660 ± 0.10	1.450* ± 0.08	1.360 ± 0.05	38.400 ± 1.20	36.600 ± 1.00	1.770 ± 0.12
	R	0.800 ± 0.02	0.779 ± 0.02	0.821* ± 0.02	0.532 ± 0.02	0.532 ± 0.02	0.865 ± 0.01
Total	Total (Opt/ Subopt)	5–4	8–3	6–18	0–2	0–5	19–9

Opt-Subopt represent the number of optimal and suboptimal values achieved by this algorithm across all metrics and datasets.

* denotes suboptimal (second-best) values. Bold values indicate the best (optimal) mean performance.

Table 10. Generalization Performance and Optimization Gap Across Different Datasets in Vehicle-Cargo Matching.

Dataset	TM			LR			TiM			ER		
	Opt	Gen	Gap	Opt	Gen	Gap	Opt	Gen	Gap	Opt	Gen	Gap
CV50	0.52	0.51	1%	0.84	0.84	1%	0.94	0.93	1%	0.62	0.64	3%
CV100	0.56	0.55	2%	0.81	0.80	1%	0.61	0.60	2%	0.60	0.61	1%
CV500	0.71	0.70	1%	0.83	0.84	1%	0.76	0.75	2%	0.65	0.65	0%
CV1k	0.62	0.62	0%	0.77	0.77	1%	0.66	0.66	0%	0.63	0.63	0%
CV2k	0.62	0.61	1%	0.78	0.77	0%	0.66	0.66	0%	0.63	0.63	0%

Opt represents the optimal value in the current dataset.

Gen represents the generalization result.

Gap represents the difference between the generalization value and the optimal value.

5.5 Pareto Optimality Experiment

The vehicle-cargo matching problem is a critical multi-objective optimization task in the field of logistics and distribution. Its goal is to efficiently allocate goods to vehicles in a manner that maximizes resource utilization and customer satisfaction. This problem typically involves various constraints, such as the type of cargo, the capacity of the vehicle, and the time requirements, and is further complicated by datasets of varying scales. Therefore, there is an urgent need for an efficient multi-objective optimization algorithm to balance these diverse demands. Figure 6 presented demonstrates the optimization performance of the vehicle-matching problem across datasets of different scales (CV50, CV100, CV500), specifically visualizing the distribution of Pareto optimal solutions through a three-dimensional projection

of three matching degree indicators [30]. A Pareto optimal solution represents a set of non-dominated solutions, where no solution can improve on all objectives simultaneously without sacrificing the performance of at least one objective. The optimization objectives for this problem include: 1) Type matching degree, which measures the compatibility between vehicle types and cargo types. For example, cold-chain transport requires refrigerated vehicles, while high-value goods demand vehicles with enhanced security features; 2) Time matching degree, which assesses the alignment between vehicle schedules and cargo delivery time windows. 3) Empty running rate degree, which represents the distance or time traveled by vehicles without cargo. A high empty mileage rate leads to resource waste and increased transportation costs; ensuring timely delivery of goods. 4) Empty loading rate degree, which reflects the utilization of vehicle space and load capacity. A high load factor indicates efficient resource use and minimizes idle space.

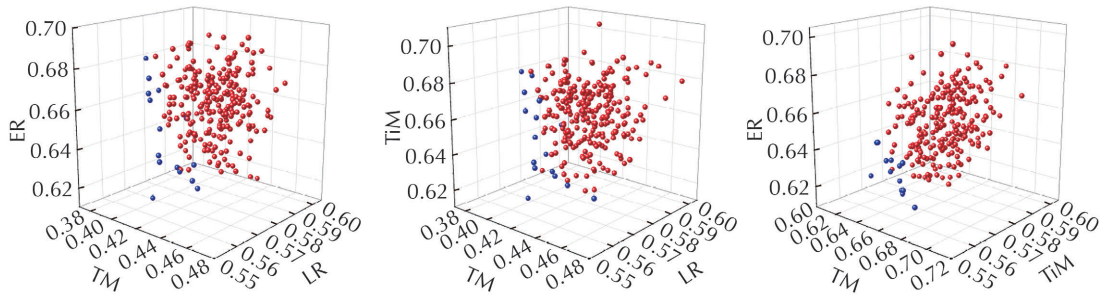


Figure 6. The Distribution of Pareto Optimal Solutions for Multi-Objective Vehicle-Cargo Matching on the CV50 Dataset.

By implementing these multi-objective optimizations, the efficiency and cost-effectiveness of logistics distribution systems can be significantly enhanced. Figure 6 illustrates the distribution of Pareto optimal solutions for the CV50 dataset under multi-objective optimization. The figure clearly presents the distribution differences between the obtained solutions and the ideal Pareto front, providing valuable insights into the impact of data scale on the effectiveness of multiobjective optimization.

As observed in Figure 6, for the CV50 dataset, the relatively small data scale and limited number of goods and vehicles result in a more concentrated distribution of optimal solutions. When optimizing type matching degree, loading rate, and empty running rate, the distribution of Pareto optimal solutions is relatively uniform, indicating that multi-objective optimization is less complex in small-scale datasets and that optimal solutions are more easily identified. However, when time matching degree is introduced as an optimization objective, the distribution of solutions becomes more dispersed, reflecting the increased complexity of the matching process due to the time constraints. Therefore, it can be inferred that, in smaller datasets, time matching degree becomes one of the key factors influencing optimization performance.

In multi-objective optimization problems, as the data scale increases, the complexity of the problem also escalates. Figure 7 illustrates the distribution of Pareto optimal solutions for the vehicle-cargo matching task on the CV100 dataset. Compared to smaller datasets, as the number of vehicles and goods increases, the distribution of optimal solutions becomes more dispersed, reflecting the diversity and complexity of the optimization task. This observation provides valuable insights for the performance evaluation of subsequent algorithms on medium-scale datasets.

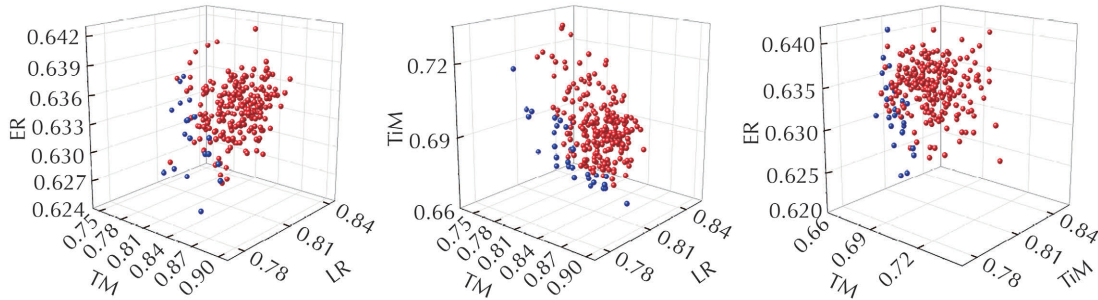


Figure 7. The Distribution of Pareto Optimal Solutions for Multi-Objective Vehicle-Cargo Matching on the CV100 Dataset.

As seen in Figure 7, as the dataset size increases to CV100, the complexity of the matching task increases significantly. At this stage, the distribution of Pareto optimal solutions becomes more dispersed, particularly in the combination of loading rate and time matching degree. This indicates that, with an increase in the number of vehicles and goods, the diversity of optimization tasks rises, and different solutions may achieve better trade-offs between certain objectives. For the combination of type matching degree and empty running rate, the distribution of optimal solutions is broader, suggesting that multi-objective optimization in medium-scale datasets requires more refined algorithms to balance the weights of various objectives effectively.

As the data scale further increases, the complexity of the multi-objective optimization problem also rises. Figure 8 illustrates the distribution of Pareto optimal solutions for the vehicle-cargo matching task on the CV500 dataset. Compared to smaller datasets, the distribution of optimal solutions in large-scale datasets exhibits a noticeable clustering phenomenon, particularly in the combination of loading rate and time matching degree. This indicates that, in large-scale datasets, the complexity of the optimization problem increases, and algorithms must perform a more detailed balance across multiple objectives to effectively handle the complexity and diversity in the optimization process.

As observed in Figure 8, in the CV500 dataset, as the data scale further increases, the complexity of the matching task rises significantly. The distribution of optimal solutions exhibits a noticeable clustering phenomenon, particularly in the optimization of loading rate and time matching degree, indicating that multi-objective optimization becomes notably more challenging in large-scale datasets. For the combination of type matching degree, empty running rate, and time matching degree, the distribution

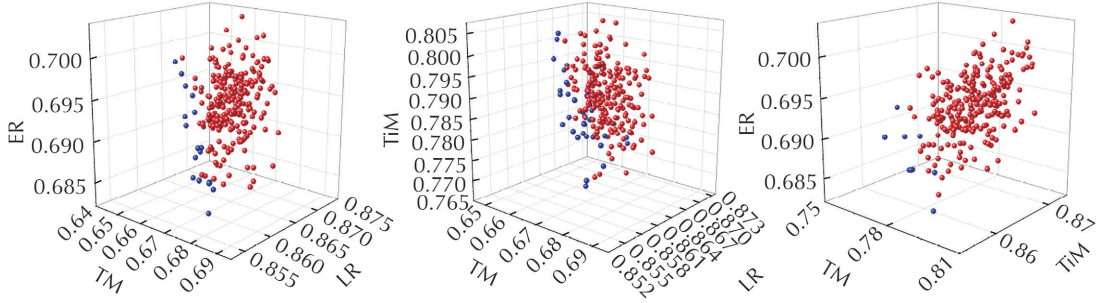


Figure 8. The Distribution of Pareto Optimal Solutions for Multi-Objective Vehicle-Cargo Matching on the CV500 Dataset.

of optimal solutions is much more dispersed, making it difficult to identify clear clustering centers. This suggests that, in large-scale datasets, the multi-dimensional optimization requirements lead to greater diversity in solutions, necessitating a more detailed trade-off among multiple objectives by the algorithm.

Analysis of the Pareto front diagrams reveals that, as the dataset size increases, the distribution of Pareto optimal solutions becomes more diverse and complex. This indicates that the scale of the dataset has a significant impact on multi-objective optimization in the vehicle-cargo matching problem, particularly when multiple matching degree dimensions are optimized simultaneously. In large-scale datasets, the distribution of optimal solutions is broader and more intricate. This observation provides crucial insights for the design of subsequent optimization algorithms, highlighting the need for more refined and adaptable strategies to handle the complexity of optimization problems in large-scale datasets.

5.6 Parametric Study

In this study, the CV100 dataset was selected as the experimental subject to analyze the impact of various key parameters on the performance of DLSAM. The following important parameters are discussed: batch size, learning rate, discount factor (γ), memory capacity, the number of encoder and decoder layers, and the number of LSTM layers. Table 11 summarizes the configuration of these parameters.

To ensure realistic matching constraints in the simulation, the loading rate threshold (δ^R) and empty running rate threshold (δ^{ER}) in Eq. (18) were both set to 0.5. This choice follows industry benchmarks in less-than-truckload logistics, where vehicle utilization below 50% is generally regarded as inefficient, and an empty running ratio above 50% significantly reduces profit margins. In Eq. (20), the window size parameter H used to compute the historical average score was set to 50. This configuration balances stability and responsiveness: it smooths short-term stochastic fluctuations in reward signals while preserving the model's ability to adapt to environmental changes. Moreover, $\epsilon_{\min}$ was set to 0.05 to retain a minimum level of exploration in later training stages and prevent convergence to local optima. The decay factor ϵ_{decay} was set to 0.995, enabling a gradual transition from exploration to exploitation as training progresses.

Table 11. DLSAM parameter experimental design.

Parameter name	Parameter quantity
Batch size	8, 32, 64, 128
Learning rate	0.001, 0.01, 0.05, 0.1
Gamma	0.7, 0.75, 0.8, 0.85
Memory size	5000, 10000, 20000, 50000
Number of Encoder Layers	1, 2, 3
Number of Decoder Layers	1, 2, 3
Number of LSTM Layers in Encoder	1, 2, 3, 4
Number of LSTM Layers in Decoder	1, 2, 3, 4

This study analyzes the impact of different encoder-decoder layer configurations and the number of Long Short-Term Memory (LSTM) layers on the performance of the DLSAM model, as illustrated in Figure 9.

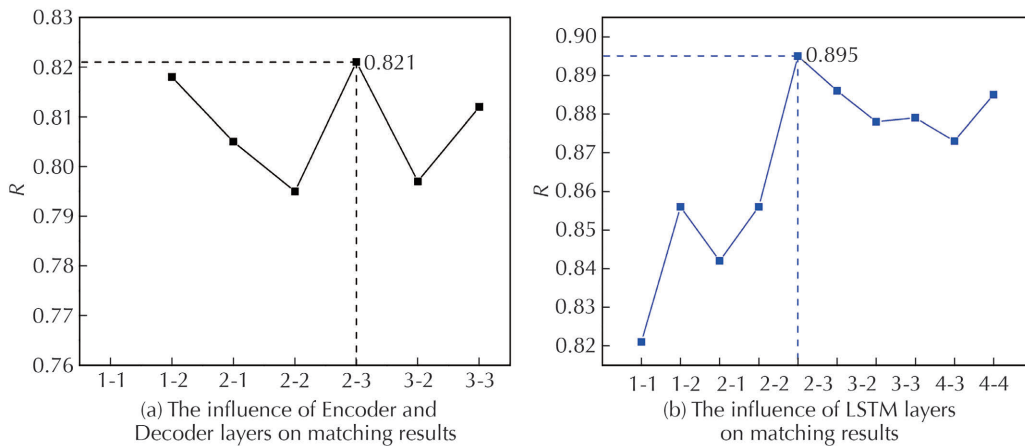


Figure 9. Effect of Encoder-Decoder Layers and LSTM Layers on DLSAM Performance.

The performance trends of DLSAM under various combinations of encoder-decoder and LSTM layer configurations. The results indicate that when both the encoder and decoder have 2 layers, the performance of DLSAM improves as the number of LSTM layers increases, reaching a peak before subsequently declining. Specifically, when the encoder has 2 layers, the decoder has 3 layers, and the number of LSTM layers in the encoder and decoder are set to 2 and 3, respectively, DLSAM achieves its highest R -value and optimal performance. This suggests that moderately increasing the number of LSTM layers within a certain range can enhance model performance. However, excessive layers may increase the complexity of DLSAM, negatively impacting its overall effectiveness.

The impact of batch size, learning rate, discount factor (γ), and memory capacity on the model's performance is illustrated in Figure 10. R , a comprehensive metric, measures the overall performance of DLSAM as a weighted combination of four key indicators: type matching degree (TM), loading rate (LR), empty running rate (ER), and time matching degree (TiM). Experimental results demonstrate that the model achieves optimal performance with a batch size of 64, yielding an R value of 0.874. Increasing or decreasing the batch size leads to a decline in model performance, indicating that an appropriate batch size contributes to the model's stability and generalization capability. Regarding the learning rate, the model performs best with a learning rate of 0.05, achieving an R value of 0.865. Both higher and lower learning rates weaken the model's performance, suggesting that a balanced learning rate optimizes the tradeoff between convergence speed and learning stability. For the discount factor (γ), the model achieves the highest R value of 0.856 when $\gamma = 0.75$. A very high or low discount factor reduces model performance, indicating that a moderate discount factor strikes the best balance between short-term and long-term rewards. Lastly, experimental results regarding memory capacity show that the model performs optimally with a memory size of 10,000, yielding an R value of 0.856. Smaller or larger memory sizes may hinder the model's effective use of historical information, thus affecting overall performance. These findings highlight the critical role of parameter tuning in optimizing model performance.

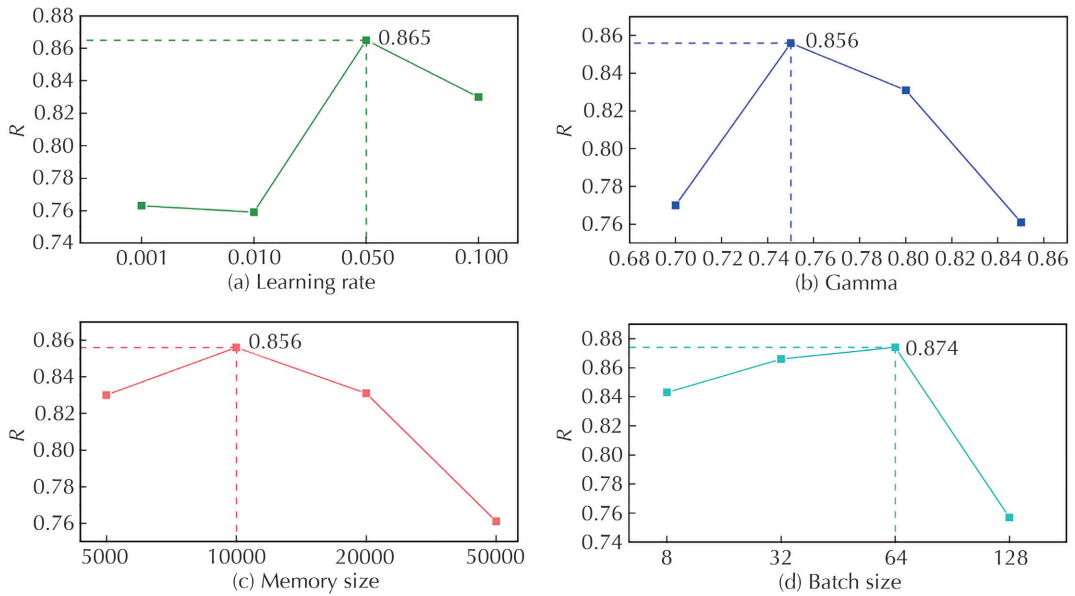


Figure 10. Experimental Results of Other Parameters.

6. PRACTICE MANAGEMENT

In modern logistics, resource optimization is critical for improving transportation efficiency and reducing costs. The proposed vehicle-cargo matching algorithm, DLSAM, which integrates Deep Q-Learning, Long Short-Term Memory (LSTM) networks, and a multi-head self-attention mechanism, has

demonstrated exceptional resource optimization capabilities in real-world logistics scenarios. Experimental results show that on the CV500 dataset, the Loading Rate (LR) achieved by DLSAM reached 87.4%, significantly outperforming traditional methods such as DQN (85.3%) and DBN (58.3%). Furthermore, the Empty Running Rate (ER) was reduced to just 29.6%, effectively minimizing resource waste in large-scale logistics networks. This optimization has substantial economic implications for logistics enterprises. For example, in nationwide parcel delivery networks, DLSAM can reduce transportation costs by approximately 15% by decreasing empty driving distances and improving loading efficiency, while simultaneously enhancing overall delivery efficiency. This performance is particularly well-suited to complex scenarios such as multi-point distribution and reverse logistics, achieving the dual objectives of cost reduction and efficiency improvement while maximizing the utilization of logistics resources.

Time Matching Rate (TiM), a core metric for evaluating the timeliness of logistics services, demonstrated exceptional performance in the experiments. On the CV50 dataset, DLSAM achieved a TiM of 0.970, significantly outperforming traditional methods such as DBN (0.29) and DQN (0.935). In time-sensitive tasks, such as cold chain transportation, DLSAM showed a clear advantage by effectively ensuring the on-time delivery of goods. For instance, in cold chain logistics scenarios, DLSAM can dynamically optimize delivery routes and scheduling to ensure that perishable goods are delivered within the specified time window. It is estimated that, when applied in practice, DLSAM significantly improve customer satisfaction scores. This ability to optimize delivery timeliness provides logistics enterprises with a competitive edge in attracting and retaining customers in highly competitive market environments.

The logistics distribution process is often influenced by external uncertainties such as traffic congestion, weather changes, and policy restrictions. By integrating Long Short-Term Memory (LSTM) networks and a Multi-Head Self-Attention Mechanism, DLSAM effectively captures dynamic environmental features and adjusts its matching strategies in real time. On the CV5K dataset, which consists of 5,000 vehicle-cargo matching tasks, DLSAM achieved an Empty Running Rate (ER) of 35.8% and a Loading Rate (LR) of 84.2%, maintaining high efficiency even in large-scale and complex scenarios. For example, during the “Double 11” promotional event on e-commerce platforms, where order volumes surge and delivery regions are highly dispersed, DLSAM efficiently processes real-time data to dynamically prioritize order assignments. This capability significantly reduces the proportion of delayed orders, thereby enhancing the overall robustness of logistics operations in complex and high-demand scenarios.

7. CONCLUSION

This study addresses a core challenge in logistics distribution vehicle-cargo matching by proposing an innovative solution based on deep reinforcement learning: the DLSAM (Deep QLearning with LSTM and Self-Attention Mechanism) model. This problem involves efficiently assigning cargo to appropriate transportation vehicles in dynamic, multi-objective environments. To tackle these challenges, we designed a deep reinforcement learning approach that integrates LSTM and multi-head self-attention mechanisms, enabling the model to capture long-term dependencies in time series data and extract key features for optimizing complex matching scenarios.

Extensive experimental validation demonstrates that DLSAM outperforms traditional heuristic methods and other reinforcement learning approaches, such as DQN and PPO, across multiple datasets. The results highlight DLSAM's superior performance on key metrics, including type matching rate, loading rate, empty running rate, and time matching rate. The model not only efficiently computes near-optimal solutions, but also exhibits strong generalization capabilities across varying data distributions. Notably, improvements in the Q-network enable better capture and optimization of the complex dependencies between vehicles and cargo, significantly enhancing the model's decision-making ability and adaptability. Furthermore, DLSAM demonstrates considerable flexibility and robustness, dynamically adjusting its matching strategies in realtime to ensure efficient transportation and resource utilization in diverse scenarios. Compared to existing reinforcement learning algorithms, DLSAM achieves superior overall performance, particularly in handling large-scale datasets, where it offers notable advantages in both matching speed and accuracy.

In future research, our objective is to further optimize and expand the DLSAM model to enhance its practical value in logistics distribution. Specifically, our efforts will focus on the following areas: (1) As logistics demands continue to evolve, vehicle-cargo matching tasks in logistics distribution scenarios are becoming increasingly complex and dynamic. Future studies will focus on improving DLSAM's adaptability to dynamic environments, enabling it to respond quickly to real-time changes in logistics demands, such as surges in orders, fluctuations in vehicle availability, and sudden traffic disruptions. To achieve this, we plan to explore the integration of dynamic programming with deep reinforcement learning, equipping the model with enhanced adaptive capabilities to make efficient matching decisions in the face of diverse logistics environments and demand variations. (2) In multi-objective optimization, there are often trade-offs between different objectives, such as improving the loading rate potentially increasing vehicle travel time. Future research will focus on optimizing strategies to balance multiple objectives and exploring how to define weight relationships more precisely among various objectives within the model. We will investigate approaches based on game theory and Pareto front optimization to establish a more robust multi-objective optimization framework for DLSAM, ensuring that the model achieves a comprehensive optimal solution by balancing key performance metrics effectively.

REFERENCES

- [1] A. Li, S. Zhuang, T. Yang, W. Lu, and J. Xu, "Optimization of logistics cargo tracking and transportation efficiency based on data science deep learning models," *Applied and Computational Engineering*, vol. 69, pp. 71–77, 2024.
- [2] L. Kong and M. Huang, "Stable truckload matching in the vehicle-cargo matching platform," in *2024 36th Chinese Control and Decision Conference (CCDC)*, 2024, pp. 5817–5821.
- [3] R. Tian, C. Wang, Z. Ma, Y. Liu, and S. Gao, "Research on vehicle-cargo matching algorithm based on improved dynamic bayesian network," *Computers & Industrial Engineering*, vol. 168, p. 108039, 2022.
- [4] J. Deng, H. Zhang, and S. Wei, "Prediction of vehicle-cargo matching probability based on dynamic bayesian network," *International Journal of Production Research*, vol. 59, no. 17, pp. 5164–5178, 2021.
- [5] M. Feng and Y. Cheng, "Solving truck-cargo matching for drop-and-pull transport with genetic algorithm based on demand-capacity fitness," *Alexandria Engineering Journal*, vol. 60, no. 1, pp. 61–72, 2021.

- [6] O. Kesemen and E. Özkul, "Solving cross-matching puzzles using intelligent genetic algorithms," *Artificial Intelligence Review*, vol. 49, pp. 211–225, 2018.
- [7] X. He, Y. Zhang, Z. Le, and Y. Du, "Research on "one-to-many" vehicle and cargo matching optimization problem based on improved genetic algorithm," in *Proceedings of the 2022 7th International Conference on Intelligent Information Technology*. Association for Computing Machinery, 2022, pp. 7–15.
- [8] Z.-H. Hu and C. Wei, "Synchronizing vehicles for multi-vehicle and one-cargo transportation," *Computers & Industrial Engineering*, vol. 119, pp. 36–49, 2018.
- [9] Z. Jiang, Y. Chen, X. Li, and B. Li, "A heuristic optimization approach for multi-vehicle and one-cargo green transportation scheduling in shipbuilding," *Advanced Engineering Informatics*, vol. 49, p. 101306, 2021.
- [10] D. L. Enthoven, B. Jargalsaikhan, K. J. Roodbergen, M. A. uit het Broek, and A. H. Schrotenboer, "The two-echelon vehicle routing problem with covering options: City logistics with cargo bikes and parcel lockers," *Computers & Operations Research*, vol. 118, p. 104919, 2020.
- [11] H. Wang, M. Huang, and H. Kuang, "One-to-one stable vehicle cargo matching in the fourth party logistics," in *2021 33rd Chinese Control and Decision Conference (CCDC)*, 2021, pp. 4221–4226.
- [12] P. Jiang, S. Guo, B. Du, and J. Guo, "Two-sided matching decision-making model for complex product system based on life-cycle sustainability assessment," *Expert Systems with Applications*, vol. 208, p. 118184, 2022.
- [13] X. Mu, Y. Chen, and H. Zhao, "Freight information matching based on quantum evolutionary algorithm," in *Proceedings of the 2016 International Conference on Engineering Management (Iconf-EM 2016)*. Atlantis Press, 2017/01.
- [14] T. Shanthi, M. Ramprasath, A. Kavitha, and T. Muruganantham, "Deep learning based autonomous transport system for secure vehicle and cargo matching," *Intelligent Automation & Soft Computing*, vol. 35, no. 1, 2023.
- [15] L. Almuqren, F. S. Alrayes, M. A. Arasi, F. F. Alruwaili, A. Mohamed, and M. Assiri, "Blockchain-assisted vehicle and cargo matching using optimal fuzzy restricted boltzmann machine in autonomous transport system," *IEEE Access*, vol. 11, pp. 105698–105705, 2023.
- [16] Z. Zhou, M. Wang, J. Huang, S. Lin, and Z. Lv, "Blockchain in big data security for intelligent transportation with 6g," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 9736–9746, 2022.
- [17] J. Li, Y. Zheng, B. Dai, and J. Yu, "Implications of matching and pricing strategies for multiple-delivery-points service in a freight o2o platform," *Transportation Research Part E: Logistics and Transportation Review*, vol. 136, p. 101871, 2020.
- [18] P. Fontaine, "The vehicle routing problem with load-dependent travel times for cargo bicycles," *European Journal of Operational Research*, vol. 300, no. 3, pp. 1005–1016, 2022.
- [19] S. Narayanan and C. Antoniou, "Electric cargo cycles-a comprehensive review," *Transport Policy*, vol. 116, pp. 278–303, 2022.
- [20] H. Golpira, S. A. R. Khan, and S. Safaeipour, "A review of logistics internet-of-things: Current trends and scope for future research," *Journal of Industrial Information Integration*, vol. 22, p. 100194, 2021.
- [21] D. Pu and G. Yuan, "Two-sided matching model considering multi-information fusion of stakeholders," *Expert Systems with Applications*, vol. 212, p. 118784, 2023.
- [22] Z. Wang, Y. Li, F. Gu, J. Guo, and X. Wu, "Two-sided matching and strategic selection on freight resource sharing platforms," *Physica A: Statistical Mechanics and its Applications*, vol. 559, p. 125014, 2020.
- [23] N. Batarliene, "Essential safety factors for the transport of dangerous goods by road: A case study of lithuania," *Sustainability*, vol. 12, no. 12, 2020.

- [24] S. Hosseini, D. Ivanov, and A. Dolgui, "Ripple effect modelling of supplier disruption: integrated markov chain and dynamic bayesian network approach," *International Journal of Production Research*, vol. 58, no. 11, pp. 3284–3303, 2020.
- [25] Z. Wen, H. Liao, R. Ren, C. Bai, E. K. Zavadskas, J. Antucheviciene, and A. Al-Barakati, "Cold chain logistics management of medicine with an integrated multi-criteria decision-making method," *International Journal of Environmental Research and Public Health*, vol. 16, no. 23, 2019.
- [26] M. Shurrah, S. Singh, H. Otrouk, R. Mizouni, V. Khadkikar, and H. Zeineldin, "A stable matching game for v2v energy sharing-a user satisfaction framework," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7601–7613, 2022.
- [27] B. Yang, K. Han, W. Tu, and Q. Ge, "Fairness in online vehicle-cargo matching: An intuitionistic fuzzy set theory and tripartite evolutionary game approach," *Applied Soft Computing*, vol. 167, p. 112418, 2024.
- [28] L. Ding, H. Huang, and H. Yuan, "Impacts of distributional fairness concerns on quality disclosure in a dyadic supply chain," *Computers & Industrial Engineering*, vol. 182, p. 109416, 2023.
- [29] S. Shen, C. Cai, Z. Li, Y. Shen, G. Wu, and S. Yu, "Deep q-network-based heuristic intrusion detection against edge-based sdn zero-day attacks," *Applied Soft Computing*, vol. 150, p. 111080, 2024.
- [30] E. H. Haro, O. Avalos, J. Gálvez, and O. Camarena, "An integrated process planning and scheduling problem solved from an adaptive multi-objective perspective," *Journal of Manufacturing Systems*, vol. 75, pp. 1–23, 2024.
- [31] B. Kamana-Williams, D. Bishop, G. Hooper, and J. Chase, "Driving change: Electric vehicle charging behavior and peak loading," *Renewable and Sustainable Energy Reviews*, vol. 189, p. 113953, 2024.
- [32] E. Vakili, A. Amirkhani, and B. Mashadi, "Dqn-based ethical decision-making for self-driving cars in unavoidable crashes: An applied ethical knob," *Expert Systems with Applications*, vol. 255, p. 124569, 2024.
- [33] S. M. Al-Selwi, M. F. Hassan, S. J. Abdulkadir, A. Muneer, E. H. Sumiea, A. Alqushaibi, and M. G. Ragab, "Rnnlstm: From applications to modeling techniques and beyond—systematic review," *Journal of King Saud University-Computer and Information Sciences*, vol. 36, no. 5, p. 102068, 2024.
- [34] R. Reka, R. Karthick, R. Saravana Ram, and G. Singh, "Multi head self-attention gated graph convolutional network based multi-attack intrusion detection in manet," *Computers & Security*, vol. 136, p. 103526, 2024.

AUTHOR BIOGRAPHY

Ran Tian obtained his Ph.D. in Computer Application Technology from Southwest Jiaotong University in 2015. Presently, he is an associate professor at the College of Software, Northwest Normal University (NWNNU), China. His ongoing research concentrates on resolving optimization issues that arise within supply chains.

(E-mail: tianran@nwnu.edu.cn)

Jian Yang is a graduate student at the College of Software, Northwest Normal University, Lanzhou 730000, China.

(E-mail: 2023222160@nwnu.edu.cn)

Qingxia Tang is a graduate student at the College of Software, Northwest Normal University, Lanzhou 730000, China.

(E-mail: 18909445744@163.com)

Jiangtao Guo is a graduate student at the College of Software, Northwest Normal University, Lanzhou 730000, China.

(E-mail: guojt1224@163.com)

Ruihan Yuan is a graduate student at the College of Software, Northwest Normal University, Lanzhou 730000, China.

(E-mail: 18103481438@163.com)

Jitong Lian is a graduate student at the College of Software, Northwest Normal University, Lanzhou 730000, China.

(E-mail: lianjitong@nwnu.edu.cn)