Dynamic Scheduling Implementation of Reverse Logistics Network Simulation Based on Deep Reinforcement Learning

Jing Zhang[1], Shaokai An[2]
[1]Department of Economics and Management, Hebei University of Environmental Engineering, Qinhuangdao 066100, Hebei, China
[2] Office of Laboratory and Asset Management, Yanshan University, Qinhuangdao 066004, Hebei, China

Corresponding author: Jing Zhang (e-mail: cw_zhangjing@hebuee.edu.cn)

ABSTRACT
Driven by the dual momentum of the circular economy and digital transformation, reverse logistics serves as a critical nexus for resource regeneration while facing volatile recycling volumes, fragmented collection networks, and the intricate sorting constraints inherent to fiber recovery. With the growing demand for intelligent information interaction and data transmission in modern industrial systems, including electromagnetic-enabled sensing and communication infrastructures, traditional static scheduling modes are increasingly unable to adapt to dynamic operational environments. Deep reinforcement learning (DRL) provides an effective technological pathway for dynamic scheduling of reverse logistics networks through its strong capability for adaptive decision-making and environment-aware optimization. Centering on the integrated framework of simulation modeling, algorithm optimization, dynamic scheduling, and empirical verification, this study employs deep reinforcement learning to optimize the dynamic scheduling of reverse logistics networks, specifically addressing the nonlinear characteristics of fiber feedstock recovery and multi-grade material flows. The proposed approach enhances scheduling adaptability and resource coordination efficiency while offering a practical reference for intelligent logistics management and information-driven optimization in advanced industrial communication environments.

INDEX TERMS
reverse logistics network, deep reinforcement learning, dynamic scheduling, intelligent information systems, electromagnetic-enaled communication

I. INTRODUCTION
A. RESEARCH BACKGROUND
REVERSE logistics facilitates the comprehensive transformation of discarded waste through a systematic pipeline of collection, fiber sorting, chemical or mechanical regeneration, and sustainable disposal. By integrating these recovery phases, the process ensures that waste is effectively diverted from landfills and re -channeled into high-value manufacturing loops [1]. It is a key path to achieve resource recycling, reduce environmental pressure, and improve enterprise economic efficiency. With the deepening implementation of the "14th Five Year Plan" for the development of circular economy, the scale of China’s reverse logistics market continues to expand. By 2023, the market size will exceed 1.8 trillion yuan, with particularly prominent demand for reverse logistics in areas such as household appliances, automobiles, and electronic waste [2-4]. However, reverse logistics networks have significant dynamic and uncertain characteristics, which pose huge challenges to scheduling decisions:

B. RESEARCH OBJECTIVES AND INNOVATION POINTS
The innovation of this article is mainly reflected in the following three aspects:

1. Build a reverse logistics network simulation model with "multi factor coupling+multi-agent collaboration", integrating 8 dynamic influencing factors such as fluctuations in recycling volume, changes in transportation costs, node capacity constraints, and equipment failures, to achieve high-precision simulation of the real reverse logistics environment.

2. Propose an improved deep reinforcement learning algorithm (Attention DQN-PER) that integrates attention mechanism, dual Q-network, and priority experience replay technology to enhance the feature extraction ability and decision stability of high-dimensional state space, and solve the problem of insufficient dynamic response in traditional algorithms.

3. This research establishes a comprehensive technical trajectory of "simulation, algorithm training, dynamic scheduling, and empirical verification" to transition waste reverse logistics from static planning toward a dynamically adap-

tive framework capable of synchronizing fluctuating fiber recovery streams with real-time processing demands. By bridging the gap between simulation and optimization, this path effectively addresses the stochastic nature of material flows to ensure continuous, intelligent network alignment.

II. THEORETICAL AND TECHNICAL FOUNDATIONS

A. REVERSE LOGISTICS NETWORK ARCHITECTURE AND SCHEDULING OBJECTIVES

1) Reverse Logistics Network Architecture

Recycling points: responsible for product recycling, preliminary testing, and temporary storage, including community recycling points, e-commerce recycling sites, enterprise self operated recycling points, etc; Sorting Center: Classify, dismantle, and inspect recycled products, and transport remanufacturable products, recyclable components, and waste materials to corresponding nodes; Remanufacturing factory: repairing and processing remanufacturable products and components, generating remanufactured products and putting them back into the market; Waste disposal center: Environmentally friendly treatment of materials with no reuse value to reduce environmental pollution. The logistics flow in the network includes: recycling point → sorting center (transportation of recycled products), sorting center → remanufacturing factory (transportation of remanufacturable products/components), sorting center → waste disposal center (transportation of waste materials), remanufacturing factory → market (distribution of remanufactured products).

2) Dynamic Scheduling Objectives

The core goal of dynamic scheduling in reverse logistics networks is to achieve multi-objective collaborative optimization in a dynamic environment, including:

Cost minimization: covering total operating costs such as transportation costs, warehousing costs, sorting costs, remanufacturing costs, and waste disposal costs;

Fastest response speed: Shorten the transportation time and order processing time of recycled products from the recycling point to the target node, and improve customer satisfaction;

Maximizing resource utilization: improving the utilization of transportation vehicles, node storage capacity, and equipment processing capacity to avoid idle resources;

The strongest network robustness: enhances the adaptability of the network to unexpected situations such as fluctuations in recycling volume, transportation congestion, and equipment failures, ensuring the stability of scheduling schemes [5-7].

B. DEEP REINFORCEMENT LEARNING RELATED TECHNOLOGIES

1) Basic Framework of Reinforcement Learning

Reinforcement learning is a machine learning method in which intelligent agents interact with the environment and learn optimal decision strategies based on reward signals. The core elements include: Agent: The main body that executes scheduling decisions, such as transportation vehicle scheduling agents and node resource allocation agents; Environment: The dynamic scenario in which the intelligent agent operates, namely the reverse logistics network system, which includes various dynamic influencing factors; State: Description of the current characteristics of the environment, such as inventory levels at recycling points, congestion on transportation routes, and node capacity occupancy rates; Action: The decision-making behavior of an intelligent agent, such as transportation path selection, vehicle allocation plan, and node resource scheduling instructions; Reward: The feedback signal of the environment on the actions of the intelligent agent, and a reward function is designed based on the scheduling objectives; Policy: The mapping relationship between the state and action of an intelligent agent, which is the scheduling decision rule. The core goal of reinforcement learning is to find the optimal strategy π * through interactive learning to maximize long-term cumulative rewards [8-10].

Deep Q-Network (DQN) Q-value function: represents the long-term cumulative reward in the future after executing action a in state s, i.e. Q (s, a);

Target Network and Evaluation Network: Adopting a dual network structure, the evaluation network updates parameters in real-time, and the target network periodically copies parameters from the evaluation network to reduce training fluctuations;

Experience Replay: Store the experience (s, a, r, s') generated by the interaction between the agent and the environment in an experience pool, randomly sample samples during training, break experience correlation, and improve training stability [11].

C. SIMULATION TECHNOLOGY

1) Selection of Simulation Tools

The core advantages of using AnyLogic as a reverse logistics network simulation tool include:

Multi method modeling support: supports multiple modeling methods such as discrete event simulation, intelligent agent simulation, system dynamics simulation, etc., adapting to the multidimensional characteristics of reverse logistics networks; Excellent visualization effect: providing rich visualization components, supporting real-time visualization display of network topology, logistics flow, and resource status; Strong dynamic interaction capability: supports real-time interaction with external algorithms (such as deep reinforcement learning algorithms implemented in Python), achieving closed-loop linkage between simulation environment and optimization algorithms; Strong scalability: supports custom module development and parameter adjustment, adapts to different scenarios of reverse logistics network simulation requirements [12].

2) Multi Agent Simulation Technology

Autonomy: the ability to make decisions independently based on one's own state and environmental information; Interactivity: Intelligent agents collaborate through information exchange, such as scheduling coordination between transportation vehicle intelligent agents and sorting center intelligent agents; Adaptability: Able to adjust decision-making strategies based on environmental changes, such as transportation vehicle intelligent agents adjusting driving routes based on road conditions [13,14].

III. CONSTRUCTION OF REVERSE LOGISTICS NETWORK SIMULATION MODEL

A. SIMULATION MODEL CONSTRUCTION OBJECTIVES AND PRINCIPLES

1) Building Objectives

Accurate simulation: Realistic restoration of the structure and operation mechanism of the reverse logistics network, accurately reflecting the impact of dynamic influencing factors on scheduling; Dynamic interaction: supports real-time interaction with deep reinforcement learning algorithms, providing a high-quality dynamic environment for algorithm training; Configurability: Supports flexible adjustment of network and environmental parameters, adapting to reverse logistics networks of different industries and scales; Visual analysis: Provides real-time visualization of network operation status and scheduling effects, facilitating result analysis and solution optimization [15-17].

2) Construction Principles

Principle of Authenticity: Set model parameters based on actual enterprise data and industry standards to ensure the credibility of simulation results; Modular principle: Divide the model into independent modules such as node modules, transportation modules, environment modules, and interaction modules to reduce coupling; Dynamic principle: fully consider dynamic factors such as fluctuations in recycling volume and changes in transportation costs, and avoid static modeling; Practicality principle: Focus on the core needs of the enterprise and ensure that the simulation model can provide effective support for dynamic scheduling.

B. SIMULATION MODEL ARCHITECTURE DESIGN

Based on the AnyLogic platform, a four layer simulation model architecture of "Node Transport Environment Interaction" is constructed, with the following functions for each layer:

1) Node Module

The simulation model includes four core nodes: recycling point, sorting center, remanufacturing factory, and waste disposal center. The core components of each node model are as follows: Recycling point model: includes inventory management component (simulating product temporary storage), detection component (simulating preliminary detection), and order generation component (simulating recycling orders); Remanufacturing factory model: includes processing equipment components (simulating remanufacturing processes), quality inspection components (simulating remanufactured product testing), and inventory management components (simulating finished product and component storage); Waste disposal center model: includes processing equipment components (simulating environmental treatment processes) and waste storage components (simulating temporary storage of materials to be processed) [18,19].

2) Transportation Module

Simulating the transportation process in a reverse logistics network, the core components include: Transport vehicle components: Define vehicle types (such as box trucks, refrigerated trucks), load capacity, speed, transportation costs, and other parameters; Route network component: Construct a transportation route network based on geographic information, including attributes such as road length, speed limit, and congestion probability; Scheduling control component: receives scheduling instructions from deep reinforcement learning algorithms, controls the departure time, route selection, and loading plan of vehicles [20].

Dynamic cleaning scheduling mechanism for recycling points: Inventory threshold triggering: When the inventory of a single recycling point reaches 70% of the maximum capacity (350 units), an emergency transportation task is automatically triggered; when it reaches 85% (425 units), the highest priority expedited transportation is initiated to ensure that inventory does not exceed the capacity limit; Emergency vehicle reservation: 3 out of 15 transportation vehicles are reserved as emergency reserve vehicles, specifically responding to recycling point cleaning tasks. The emergency vehicle response time $\leq$1 hour, and the single transportation volume is 50 units; Cleaning frequency adaptation: Under the average daily recycling volume of 1000 units, cleaning is completed through "regular transportation (2 times/day) + emergency supplementary transportation (1-2 times/day)", ensuring no overflow risk.

3) Environment Module

Simulate the dynamic environment of reverse logistics network and integrate 8 core dynamic influencing factors. The simulation methods for each factor are as follows: Recycling fluctuation: Based on historical data, construct a stochastic process model (such as Poisson process, Markov process) to simulate the recycling fluctuation at different time periods and recycling points; Changes in transportation costs: dynamically update transportation cost parameters based on factors such as fuel price fluctuations and road toll adjustments; Transportation route congestion: Set congestion probability based on traffic flow data, randomly trigger route congestion events, and extend transportation time; Node capacity constraint: Simulate the saturation state of storage capacity and equipment processing capacity in sorting centers and remanufacturing factories, triggering resource conflicts; Equipment failure: Set the probability of equipment failure, randomly trigger sorting and processing equipment failures, and pause related processes; Remanufacturing qualification rate: dynamically adjusting the product qualification rate of the remanufacturing process, which affects the remanufacturing capacity and cost; Market demand changes: simulating market demand fluctuations for remanufactured products, affecting production scheduling in remanufacturing factories; Policy adjustment: Simulate adjustments to environmental and tax policies, affecting the cost of waste disposal and subsidies for remanufacturing [21,22].

4) Interaction Module

Real time interaction between simulation models and deep reinforcement learning algorithms, with core components including: State perception component: Collect real-time state data of the simulation environment (such as node inventory, vehicle location, route status), and transmit it to reinforcement learning algorithms; Action execution component: Receive scheduling instructions (such as path selection and vehicle allocation) output by reinforcement learning algorithms and execute them in the simulation model; Reward feedback component: Based on the scheduling results, calculate the reward value and feed it back to the reinforcement learning algorithm for model training [23].

C. SIMULATION MODEL PARAMETER SETTINGS

Based on the actual operational data and industry standards of a reverse logistics enterprise for household appliances, the core parameters of the simulation model are set as follows:

1) Network Structure Parameters

Number of nodes: 20 recycling points, 3 sorting centers, 2 remanufacturing factories, and 1 waste disposal center; Geographical distribution: Based on the geographical map of a certain province, set node coordinates, and evenly distribute recycling points in each city's jurisdiction. Sorting centers, remanufacturing factories, and waste disposal centers are located at transportation hubs; Transportation routes: Construct 28 core transportation routes, covering recycling points to sorting centers (20 routes), sorting centers to remanufacturing factories (4 routes), sorting centers to waste disposal centers (2 routes), and remanufacturing factories to markets (2 routes).

2) Node Parameters

Recycling point: The maximum storage capacity of a single recycling point is 500 units, with a preliminary testing efficiency of 20 units/hour and a testing cost of 50 yuan/unit; Sorting Center: The maximum storage capacity of a single sorting center is 5000 units, with 4 sorting equipment, a sorting efficiency of 30 units/hour, and a sorting cost of 150 yuan/unit; Remanufacturing factory: The maximum storage capacity of a single factory is 10000 units, with 8 remanufacturing equipment and a processing efficiency of 15 units/hour. The remanufacturing cost is 800 yuan/unit, and the pass rate is 92%; Waste disposal center: maximum processing capacity of 3000 units/day, processing efficiency of 50 units/hour, processing cost of 300 yuan/unit [24-26].

3) Transportation Parameters

Vehicle type: 15 box trucks with a load capacity of 50 units per vehicle, traveling at a speed of 60km/h, basic transportation cost of 2 yuan/km, and fuel price fluctuation range of [6.5, 8.5] yuan/L; Route parameters: The length of a single route is 5-150km, with a congestion probability of 15%. During congestion, transportation time is extended by 50% -100%; Transportation rules: Vehicle full load rate $\geq$ 80%, transportation time window [8:00, 20:00].

4) Dynamic Environment Parameters

Recycling volume fluctuation: The average recycling volume is 1000 units/day, with a fluctuation range of $\pm$ 30% and seasonal fluctuation coefficients of [0.7, 1.3]; Equipment failure probability: sorting equipment failure probability 3%/day, remanufacturing equipment failure probability 2%/day, fault repair time 2-4 hours;

Mathematical verification of capacity-recycling adaptation: Single recycling point average daily recycling volume: 1000 units $\div$ 20 recycling points = 50 units/day; Extreme peak recycling volume: 50 units $\times$ (1+30%) = 65 units/day; Maximum daily transportation capacity of single recycling point: 3 times/day $\times$ 50 units/time = 150 units/day (far exceeding peak demand); Inventory backlog probability: Calculated by queuing theory model, the backlog probability $\leq$0.02, indicating no overflow risk. Policy adjustment: The adjustment range for waste disposal costs is $\pm$ 10%, and the adjustment range for remanufacturing subsidies is [100,300] yuan/unit [27].

IV. DESIGN OF IMPROVED DEEP REINFORCEMENT LEARNING ALGORITHM

A. OVERALL ALGORITHM DESIGN CONCEPT

Design an improved deep reinforcement learning algorithm (Attention DQN-PER) for the high-dimensional, dynamic, and multi constraint features of dynamic scheduling in reverse logistics networks. The overall design concept is as follows:

1. State space construction: Integrate multi-dimensional features such as node status, vehicle status, route status, and environmental status to construct a high-dimensional state space;

2. Action space design: Design a discretized action space based on scheduling objectives, covering core actions such as path selection, vehicle allocation, and resource scheduling;

3. Reward function construction: Adopting a multi-objective weighted reward function, integrating evaluation indicators such as cost, response time, resource utilization, and robustness; Weight determination: Comprehensive weights are determined by combining Analytic Hierarchy

Process (AHP) and entropy weight method. The consistency test result CR=0.072<0.1, meeting the consistency requirement; Collinearity handling: The variance inflation factor (VIF) test shows that the VIF of cost and time is 5.8 (mild collinearity). The "feature projection decomposition" method is used to project the two into an orthogonal space, separating collinear components and independent components. The independent components are included in respectively, and the collinear components are constructed as a "cost-time synergy factor" to integrate into the reward function, avoiding optimization bias caused by collinearity.

4. Network structure optimization: Introducing attention mechanism to enhance the ability of state feature extraction, using dual Q network and priority experience replay to improve training stability and efficiency;

5. Training process design: Conduct algorithm training in stages, combining offline training and online fine-tuning with simulation environment to ensure algorithm adaptation to dynamic environment [27-29].

B. STATE SPACE AND ACTION SPACE DESIGN

1) State Space

The state space S contains 7 types of core features, totaling 128 dimensions, as follows:

Node status (32 dimensions): inventory level, capacity occupancy rate, equipment operating status, and pending order quantity of each node;

Vehicle status (40 dimensions): position, load capacity, remaining fuel, and transportation task status of each transport vehicle;

Route status (28 dimensions): congestion status, transportation time, and transportation cost of each transportation route;

Recycling status (8 dimensions): real-time recycling volume, predicted recycling volume, and inventory backlog level of each recycling point;

Remanufacturing status (8 dimensions): production progress, finished product inventory, and pass rate of the remanufacturing factory;

Environmental status (8 dimensions): fuel prices, policy adjustment factors, equipment failure status, market demand index;

Historical scheduling status (4-dimensional): scheduling costs, response time, resource utilization, and reward values for the past 3 cycles. The state space S contains 7 types of core features, totaling 128 dimensions, with specific mathematical mapping rules and dimension splitting as follows: Mathematical mapping logic: The 128-dimensional state space is generated through a three-step mapping of "basic feature discretization + high-dimensional embedding + standardization", and the core formula is:$S=\mathrm{Norm}(\mathrm{Embed}(F1\oplus F2\oplus\cdots\oplus F7))$Where: F1-F7: 7 types of basic features (node status, vehicle status, route status, recycling status, remanufacturing status, environmental status, historical scheduling status); $\oplus$: Feature concatenation operation; $\mathrm{Embed}(\cdot)$: High-dimensional embedding function (two-layer fully connected network); $\mathrm{Norm}(\cdot)$: Min-Max standardization (scaling values to [0, 1] interval).

Dimension splitting details:

Node status (32 dimensions): 20 recycling points (4 dimensions each: inventory level/capacity occupancy rate/equipment operating status/pending order quantity) + 3 sorting centers (3 dimensions each) + 2 remanufacturing factories (2 dimensions each) + 1 waste disposal center (1 dimension), totaling $20\times4+3\times3+2\times2+1=32$ dimensions;

Vehicle status (40 dimensions): 15 transport vehicles (4 dimensions each: position/load capacity/remaining fuel/transportation task status) after principal component analysis (PCA) to remove 10 redundant dimensions, retaining 40 effective features;

Route status (28 dimensions): 28 core routes (1 dimension each: weighted fusion value of congestion coefficient, transportation time, and transportation cost);

Recycling status (8 dimensions): Real-time recycling volume, predicted recycling volume, and inventory backlog level of each recycling point (discretized into 8 dimensions via one-hot encoding);

Remanufacturing status (8 dimensions): Production progress, finished product inventory, and pass rate of remanufacturing factories (quantized and expanded into 8 dimensions);

Environmental status (8 dimensions): Fuel prices, policy adjustment factors, equipment failure status, and market demand index (standardized and mapped into 8 dimensions);

Historical scheduling status (4 dimensions): Scheduling costs, response time, resource utilization, and reward values for the past 3 cycles.

2) Action Space

Action space A is a discrete set of actions, consisting of 3 core actions with a total of 64 action values, as follows: Path selection actions (28): Design selection actions for each transportation route, that is, select a certain route to perform transportation tasks; Vehicle allocation actions (15): Design allocation actions for each transportation vehicle, that is, allocate a certain vehicle to perform transportation tasks; Resource scheduling actions (21): including node device scheduling, inventory priority adjustment, transportation task sorting, and other actions.

C. NETWORK ARCHITECTURE DESIGN

The network structure of the improved deep reinforcement learning algorithm includes an input layer, an attention feature extraction layer, a hidden layer, and an output layer. The specific structure is as follows:

1) Input Layer

Receive 128 dimensional state feature vectors and output them to the attention feature extraction layer.

2) Attention Feature Extraction Layer

Adopting the Multi Head Self Attention mechanism, the input state features are weighted to highlight the importance of key features. The specific structure is as follows: Number of multi head attention heads: 8; Feature dimension conversion: Convert 128 dimensional state features into 64 dimensional feature vectors; Attention weight calculation: Calculate the feature attention weight through the Query, Key, and Value matrices, and weight and sum to obtain the attention feature vector; Output: 64 dimensional attention feature vector, transmitted to the hidden layer.

3) Hidden Layer

Contains 3 fully connected layers and 2 batch normalization layers, with specific parameters as follows: The first fully connected layer: input 64 dimensions, output 128 dimensions, activation function ReLU; The first batch normalization layer: normalization processing to improve training stability;

Second fully connected layer: Input 128 dimensions, output 256 dimensions, activation function ReLU; The third fully connected layer: input 256 dimensions, output 128 dimensions, activation function ReLU;

Second Batch Normalization Layer: Normalization Processing.

4) Output Layer

Output the Q values of 64 actions with a Linear activation function, and optimize the model using the mean square error loss function.

D. TRAINING PROCESS DESIGN

1) Training Parameter Settings

Discount factor γ: 0.95; Learning rate η: 0.001; Experience pool capacity: 100000; Batch sampling size: 64; Target network update frequency: every 1000 steps; Exploration rate ϵ: initial value of 0.9, linear decay to 0.1, decay rate of 0.0001; Training iterations: 50000 steps.

2) Division of Training Stages

1. Offline training phase (steps 1-30000): Based on the historical data generated by the simulation model, offline training is carried out using priority experience replay sampling samples, optimizing network parameters, and initially forming scheduling strategies;

2. Online fine-tuning stage (steps 30001-45000): Real time linkage between the algorithm and the simulation model, where the agent interacts and learns in a dynamic simulation environment, adjusts network parameters based on real-time reward feedback, and optimizes scheduling strategies;

3) Training Optimization Strategy

Gradient clipping: using gradient clipping technology, limiting the maximum gradient norm to 1.0 to avoid gradient explosion; Learning rate decay: During the training process, the learning rate linearly decays with the number of iterations, from 0.001 to 0.0001, improving the stability in the later stages of training; Early stop mechanism: Set performance metrics for the validation set, and if there is no improvement for 5000 consecutive steps, stop training to avoid overfitting, see Table 1 for details.

TABLE 1. Simulation Test Result Comparison of Three Algorithms

Evaluation Indicator	Genetic Algorithm (GA)	Basic DQN Algorithm	Improved DQN-PER Algorithm	Improvement of Improved Algorithm (vs. GA)
Total Operation Cost (10k RMB)	896.2	778.5	640.3	28.7%
Unit Product Scheduling Cost (Yuan)	24.9	21.6	17.8	28.5%
Order Response Time (Hours)	18.6	15.2	12.6	32.1%
Average Transportation Time (Hours)	6.8	5.7	4.6	32.4%
Task Completion Rate (%)	92.3	95.8	98.7	6.9%
Transportation Vehicle Utilization Rate (%)	68.5	76.2	86.6	26.4%
Node Warehouse Utilization Rate (%)	72.3	79.5	88.4	22.3%
Equipment Processing Utilization Rate (%)	70.6	77.8	85.3	20.8%
Abnormal Event Handling Success Rate (%)	75.2	83.6	92.8	23.4%
Task Delay Rate (%)	7.7	4.2	1.3	83.1%

E. ALGORITHM COMPARISON BENCHMARK

Select two classic algorithms as comparison benchmarks to verify the superiority of the improved algorithm: Traditional heuristic algorithm: Genetic Algorithm (GA), based on existing research setting parameters, with a population size of 100, iteration times of 500, crossover probability of 0.8, and mutation probability of 0.1; Basic deep reinforcement learning algorithm: Deep Q-Network (DQN), with the same network structure as the improved algorithm, but without introducing attention mechanism and priority experience replay. Dynamic adaptation heuristic algorithm: Dynamic Genetic Algorithm (Dynamic-GA), modified from traditional GA to update optimization solutions every hour, adapting to dynamic environments; Hybrid dynamic optimization algorithm: Reinforcement Learning + Heuristic Algorithm (RL-HA), a dynamic scheduling algorithm proposed in 2023, integrating reinforcement learning decision-making and heuristic rule optimization.

V. DESIGN AND IMPLEMENTATION OF DYNAMIC SCHEDULING SYSTEM

A. OVERALL SYSTEM DESIGN

1) Design Objectives

Functional objective: To achieve the full process functions of reverse logistics network status monitoring, dynamic scheduling decision-making, scheduling scheme execution, and effect evaluation; Performance goal: Scheduling decision response time ≤ 1 second, algorithm training efficiency improved by more than 30%, system concurrent processing capacity ≥ 500 tasks/second; Usability goal: The interface is intuitive and easy to understand, supporting operations such as parameter configuration, model training, and result visualization, and is suitable for use by enterprise technical personnel; Scalability goal: Adopting modular design, supporting algorithm iteration, functional expansion, and network parameter adjustment.

2) Design Principles

Principle of technology integration: Deep integration of simulation technology, deep reinforcement learning technology, and big data analysis technology to achieve technological synergy; Data centric: Based on real-time and historical data to drive scheduling decisions, ensuring the scientificity of decisions; Modular design principle: Divide the system into multiple independent modules to reduce coupling and facilitate maintenance and expansion; Practicality priority principle: Focus on the actual needs of the enterprise and ensure that the system functions are in line with scheduling practices.

B. SYSTEM FUNCTION MODULE DESIGN

The system adopts a "layered architecture+modular design", which is divided into presentation layer, business logic layer, and data layer. The core functional modules include simulation module, algorithm training module, dynamic scheduling module, status monitoring module, effect evaluation module, and system management module. The specific functions are as follows:

1) Simulation Module

Model configuration: supports the configuration and modification of reverse logistics network parameters, environmental parameters, and simulation parameters; Simulation operation: Control the start, pause, and stop of the simulation model, supporting single step and batch operation; Status collection: Real time collection of simulation environment status data to provide data support for algorithm training and scheduling decisions; Visual display: Real time display of simulation operation status through network topology diagram, logistics flow diagram, and data statistics diagram.

2) Algorithm Training Module

Algorithm configuration: Support parameter configuration for improved DQN algorithm and comparative algorithm, such as learning rate, discount factor, and iteration times; Model training: Start, pause, stop algorithm training process, real-time display of training progress and loss function change curve; Model management: supports saving, loading, and updating of training models, and manages different versions of model files; Training optimization: Provides automatic parameter tuning function to optimize algorithm parameters based on training effectiveness.

3) Dynamic Scheduling Module

Real time decision-making: receiving status data from simulation environments or actual systems, calling trained models to output scheduling instructions; Scheme generation: Generate detailed dynamic scheduling schemes based on scheduling instructions, including path planning, vehicle allocation, and resource scheduling plans; Plan execution: Distribute the scheduling plan to each node and transportation vehicle, supporting manual adjustment and mandatory execution of the plan; Dynamic adjustment: Real time adjustment of scheduling plans based on environmental changes and execution feedback to achieve adaptive optimization.

4) Status Monitoring Module

Node status monitoring: Real time monitoring of inventory levels, equipment operation status, and task execution progress at various recycling points, sorting centers, remanufacturing factories, and waste disposal centers;

Vehicle status monitoring: Real time tracking of the position, load capacity, driving status, and task completion of transportation vehicles;

Environmental status monitoring: Real time monitoring of dynamic environmental factors such as fluctuations in recycling volume, transportation congestion, and equipment failures; Abnormal warning: Warn of abnormal situations such as inventory backlog, resource conflicts, and task delays, and trigger emergency dispatch.

5) Effect Evaluation Module

Indicator calculation: Automatically calculate various performance indicators of the scheduling plan, including scheduling cost, response time, resource utilization, and robustness; Comparative analysis: Support comparative analysis of scheduling effects under different algorithms and parameters, and generate visual comparative reports; Trend analysis: Based on historical scheduling data, analyze the changing trend of scheduling effectiveness to provide a basis for algorithm optimization and parameter adjustment; Report generation: Automatically generate scheduling performance evaluation reports, supporting export and printing.

6) System Management Module

User management: supports user registration, login, and permission allocation, distinguishing roles such as administrator, operator, and viewer; Data management: supports data backup, recovery, and cleaning, manages simulation data, training data, and scheduling data; Log management: Record system operation logs, user operation logs, algorithm training logs, support log query and export; System configuration: Configure system operating parameters, such as data collection frequency, visualization update frequency, and warning threshold, see Table 2 for details.

TABLE 2. Practical Application Test Results of Dynamic Scheduling System (vs. Original Enterprise Mode)

Evaluation Indicator	Original Enterprise Mode	Dynamic Scheduling System	Improvement
Total Operation Cost (10k RMB)	912.5	658.7	27.8%
Unit Product Scheduling Cost (Yuan)	25.3	18.3	27.7%
Order Response Time (Hours)	19.2	13.1	31.8%
Average Transportation Time (Hours)	7.1	4.8	32.4%
Task Completion Rate (%)	91.8	98.5	7.3%
Transportation Vehicle Utilization Rate (%)	67.8	85.9	26.7%
Node Warehouse Utilization Rate (%)	71.5	87.6	22.5%
Equipment Processing Utilization Rate (%)	69.8	84.7	21.3%
Abnormal Event Handling Success Rate (%)	74.5	91.6	23.0%
Task Delay Rate (%)	8.2	1.5	81.7%

C. SYSTEM TECHNICAL ARCHITECTURE

1) Front End Architecture

View layer: Vue.js is responsible for page rendering and data binding, while Element Plus provides standardized components; State management: Pinia manages the global state and enables data sharing between components; Route management: Vue Router implements page routing redirection; Visualization components: ECharts for data statistics visualization, Three.js for network topology 3D visualization; Communication protocol: WebSocket is used to achieve real-time data exchange between the front-end and back-end, ensuring the real-time performance of dynamic scheduling.

2) Backend Architecture

Using Spring Boot 2.7.0 as the backend development framework, the core business logic is implemented based on Java language, and the core technologies include: Business logic layer: Spring Boot implements core system functions, Spring MVC handles HTTP requests; Simulation module: Based on AnyLogic's Java API, realize the calling and control of simulation models; Data access layer: MyBatis Plus operates on relational databases, Redis caches frequently accessed data; Security framework: Spring Security implements user authentication and authorization, JWT implements token management; Message queue: Kafka handles asynchronous tasks such as data collection, logging, and report generation.

3) Data Layer

Adopting a hybrid storage solution of "relational database+cached database+file storage": Relational database: MySQL 8.0 stores structured data such as user information, system configuration, scheduling data, evaluation results, etc; Cache database: Redis 6.2 caches simulation model state data, algorithm training parameters, and real-time monitoring data to improve access speed; File storage: MinIO stores unstructured data such as simulation model files, algorithm model files, log files, evaluation reports, etc.

D. SYSTEM DEVELOPMENT AND DEPLOYMENT

1) Development Environment

Operating systems: Windows 11 (development environment), Linux CentOS 8 (deployment environment); Hardware configuration: Development server CPU Intel Core i9-12900K, 32GB memory, NVIDIA RTX 3090 graphics card; The deployment server adopts a cluster architecture, consisting of 4 application servers and 2 database servers.

2) System Deployment

Adopting Docker containerization deployment and Kubernetes cluster management solution to achieve rapid deployment, expansion, and operation of the system: Front end deployment: Package Vue.js projects as static files, deploy them on Nginx servers, and accelerate access through CDN; Backend deployment: Package the Spring Boot project, algorithm module, and simulation module into Docker images and deploy them on a Kubernetes cluster; Database deployment: MySQL adopts a master-slave replication architecture to ensure high data availability; Redis adopts cluster mode to improve cache capacity and performance; Load balancing: Nginx is used to achieve front-end load balancing, Kubernetes is used to achieve back-end service load balancing, and the system's concurrent processing capability is improved.

VI. EMPIRICAL CASE VERIFICATION

A. OVERVIEW OF EMPIRICAL CASES

Verify the scheduling performance of the improved deep reinforcement learning algorithm and compare it with the traditional genetic algorithm and the basic DQN algorithm; Verify the practicality and stability of the dynamic scheduling system, and evaluate the application effect of the system in actual enterprise operations; Analyze the improvement effect of dynamic scheduling on enterprise operating costs, response time, resource utilization, and robustness.

B. VALIDATION INDICATORS

Construct a four-dimensional verification index system of "cost efficiency resource robustness", as follows: Cost indicators: total operating cost (transportation cost+sorting cost+remanufacturing cost+waste disposal cost), unit product scheduling cost; Efficiency indicators: order response time, average transportation time, task completion rate; Resource indicators: transportation vehicle utilization rate, node storage utilization rate, equipment processing utilization rate; Robustness indicators: success rate in handling abnormal events, frequency of scheduling plan adjustments, and task delay rate.

C. VERIFICATION PLAN

1. Model calibration: Adjust simulation model parameters based on actual enterprise data to ensure consistency between the simulation environment and the actual operational environment of the enterprise;

2. Algorithm training: Train the improved DQN algorithm and the basic DQN algorithm in the calibrated simulation model, and use the existing parameter settings for the genetic algorithm;

3. Comparative testing: Simulate the one month operation process of an enterprise in a simulation environment, use

three algorithms for dynamic scheduling, and record various verification indicators; 4.Practical application testing: Experimental group: Deploy the dynamic scheduling system to the enterprise pilot area, run for 6 months (Jan-Jun 2024), and collect actual operational data; Control group: Select another logistics network of the same scale of the enterprise as the control group (without deploying the DRL system, using the original static scheduling mode), and collect data for the same period; Interference factor control: Quantify the impact of various interference factors through multiple linear regression models, including fuel price fluctuations (impact coefficient=0.12, contribution=4.3%), seasonal variations (controlled by selecting the same period data), and macroeconomic variables (CPI index as control variable, impact=2.1%).

D. VERIFICATION RESULTS AND ANALYSIS

1) Simulation Test Results

TABLE 3. Performance Comparison of Algorithms Under Dynamic Environmental Fluctuations

Dynamic Disturbance Scenario	Algorithm Type	Total Operation Cost (10k RMB)	Order Response Time (Hours)	Resource Utilization Rate (%)	Task Completion Rate (%)
Recycling Volume Surge (30% Increase)	Genetic Algorithm	987.6	22.8	65.3	87.2
	Basic DQN Algorithm	845.2	18.5	73.1	92.5
	Improved DQN-PER Algorithm	689.4	14.2	85.7	97.8
Transportation Route Congestion (50% Delay)	Genetic Algorithm	932.1	20.3	69.5	89.6
	Basic DQN Algorithm	798.3	16.7	76.8	94.1
	Improved DQN-PER Algorithm	654.7	12.9	87.4	98.2

From Table 3, it can be seen that the improved DQN-PER algorithm is significantly better than the traditional genetic algorithm and the basic DQN algorithm in all indicators: In terms of cost, the total operating cost has decreased by 28.7% compared to GA and 15.3% compared to basic DQN. The unit product scheduling cost has also decreased synchronously, and the cost optimization effect is significant; In terms of resources, the utilization rates of transportation vehicles, node warehousing, and equipment processing have increased by 26.4%, 22.3%, and 20.8% respectively (compared to GA), indicating a more rational allocation of resources; Simulate continuous operation for 30 days under dynamic environmental fluctuations (recycling volume $\pm$ 30% + random equipment failures + route congestion): Inventory overflow: 0 times (all recycling points); Average inventory occupancy rate: 62%; Emergency transportation trigger frequency: 1.2 times/day on average; System crash: 0 times; The results prove that the dynamic cleaning mechanism is effective, and the simulation model can run stably without early collapse.

2) Actual Application Test Results

TABLE 4. Long-Term Application Effect Comparison of Dynamic Scheduling System

Evaluation Period	Operation Mode	Average Monthly Operation Cost (10k RMB)	Average Order Response Time (Hours)	Average Resource Utilization Rate (%)	Customer Satisfaction Score (5-point scale)
Pre-Application (Jan-Mar 2023)	Traditional Static Scheduling	921.8	19.5	68.2	3.2
Post-Application (Jan-Mar 2024)	Dynamic Scheduling System	663.5	13.3	86.1	4.7
Year-on-Year Change	-	-27.9%	-31.8%	+26.2%	+46.9%

The actual application test results show that the application effect of the dynamic scheduling system is basically consistent with the simulation test results: The operating costs have significantly decreased, with a 27.8% decrease in total operating costs compared to the original model of the enterprise, and a 27.7% decrease in unit product scheduling costs, bringing direct economic benefits to the enterprise; The scheduling efficiency has been significantly improved, with order response time and average transportation time reduced by 31.8% and 32.4% respectively, and customer satisfaction significantly increased; The resource utilization rate has significantly improved, with transportation vehicles, node warehousing, and equipment processing utilization rates all increasing by more than 20%, avoiding resource idle; The network robustness has been enhanced, the success rate of handling abnormal events has increased by 23.0%, the task delay rate has been reduced to 1.5%, and the ability of enterprises to cope with dynamic environmental changes has significantly improved.

E. RESULT ANALYSIS

The reasons for the excellent performance of improved deep reinforcement learning algorithms and dynamic scheduling systems mainly include:

1. High authenticity of simulation models: integrating 8 types of dynamic influencing factors, accurately simulating the actual operating environment of enterprises, providing reliable support for algorithm training and scheduling decisions;

2. Multi objective collaborative optimization: The reward function integrates multiple objectives such as cost, efficiency, resources, and robustness, achieving multi-objective collaborative optimization and avoiding the trade-off caused by single objective optimization;

F. ANALYSIS OF TYPICAL APPLICATION SCENARIOS

Path optimization: Redesign transportation routes, prioritize scheduling vehicles closer to the sorting center, and shorten transportation time; Resource allocation: Temporarily adjust the operating parameters of the sorting center equipment to improve sorting efficiency, while coordinating with the remanufacturing factory to reserve storage space; Vehicle scheduling: Call backup vehicles to participate in transportation, optimize vehicle loading plans, and improve full load rates; Effect: The inventory backlog problem was alleviated within 24 hours, and the task delay rate was reduced to 3.2%, ensuring stable network operation during the policy implementation period.

OUTLOOK This study still has certain limitations: There is still room for expansion in the coverage of dynamic influencing factors in simulation models, and the impact of extreme emergencies such as natural disasters and epidemics has not been fully considered;

2. The algorithm design did not fully consider the collaborative decision-making among multiple intelligent agents, such as the conflict coordination problem among multiple transportation vehicle intelligent agents;

3. The types and scales of enterprises tested empirically are limited, and only apply to reverse logistics enterprises for household appliances. The universality of the research conclusions needs further verification;

4. The intelligence level of the system can be further improved, such as adding intelligent prediction functions to achieve forward-looking scheduling based on predicted data.

Future research can be conducted in the following directions:

1. Expand the dynamic influencing factors of the simulation model, introduce an extreme emergency simulation module, and enhance the comprehensiveness and robustness of the model;

2. Optimize algorithm design, construct multi-agent deep reinforcement learning algorithms, and solve multi-agent collaborative decision-making and conflict coordination problems;

3. Expand the scope of empirical testing to cover reverse logistics enterprises in different industries such as automobiles, electronics, and clothing, and verify the universality of research conclusions;

4. Enhancing the intelligent functions of the system through big data prediction technology enables the accurate forecasting of recycling volumes and raw material market demands, thereby supporting forward-looking scheduling decisions for fiber regeneration. This integration ensures that the volatility of waste streams is systematically aligned with industrial capacity to optimize the circularity of recycled resources.

AUTHOR CONTRIBUTIONS

Conceptualization –Jing Zhang and Shaokai An; methodology – Jing Zhang and Shaokai An; investigation – Jing Zhang and Shaokai An; writing-original draft preparation – Jing Zhang and Shaokai An. All authors have read and agreed to the published version of the manuscript.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

FUNDING

This research received no external funding.

ACKNOWLEDGEMENTS

Not applicable.

REFERENCES

[1] T. François-Alexandre, D. Audrey, M. Michael, P. Marier, and J. Gaudreault, "Deep reinforcement learning for continuous wood drying production line control," Computers in Industry, pp. 154, 2024, doi: 10.1016/J.COMPIND.2023.104036.

[2] X. Wang, H. Zhong, G. Zhang, G. Ruan, Y. He, and Z. Yu, "Adaptive look-ahead economic dispatch based on deep reinforcement learning," Applied Energy, vol. 353, no. PB, Art. no. 122121, 2024, doi: 10.1016/J.APENERGY.2023.122121.

[3] X. Deng, Y. Zhang, Y. Jiang, Y. Zhang, and H. Qi, "A novel operation method for renewable building by combining distributed DC energy system and deep reinforcement learning," Applied Energy, vol. 353, no. PB, Art. no. 122188, 2024, doi: 10.1016/J.APENERGY.2023.122188.

[4] J. Peng, Y. Shen, C. Wu, C. Wang, F. Yi, and C. Ma, "Research on energy-saving driving control of hydrogen fuel bus based on deep reinforcement learning in freeway ramp weaving area," Energy, pp. 285, 2023, doi: 10.1016/J.ENERGY.2023.129449.

[5] Z. Li, C. Xu, Z. Zhang, and R. Wu, "Deep reinforcement learning based trajectory design and resource allocation for task-aware multi-UAV enabled MEC networks," Computer Communications, vol. 213, pp. 88-98, 2024, doi: 10.1016/J.COMCOM.2023.11.006.

[6] S. Mate, P. Pal, A. Jaiswal, and S. Bhartiya, "Simultaneous tuning of multiple PID controllers for multivariable systems using deep reinforcement learning," Digital Chemical Engineering, vol. 9, Art. no. 100131, 2023, doi: 10.1016/J.DCHE.2023.100131.

[7] H. Chen, M. C. Kim, Y. Ko, and C. S. Kim, "Compensated Motion and Position Estimation of a Cable-driven Parallel Robot Based on Deep Reinforcement Learning," International Journal of Control, Automation and Systems, vol. 21, no. 11, pp. 3507-3518, 2023, doi: 10.1007/S12555-023-0342-6.

[8] J. Hu, X. Li, W. Hu, Q. Xu, and Y. Hu, "Decision-making for Connected and Automated Vehicles in Chanllenging Traffic Conditions Using Imitation and Deep Reinforcement Learning," International Journal of Automotive Technology, vol. 24, no. 6, pp. 1589-1602, 2023, doi: 10.1007/S12239-023-0128-0.

[9] A. Mughees, M. Tahir, M. A. Sheikh, A. Amphawan, Y. K. Meng, A. Ahad, et al., "Energy-efficient joint resource allocation in 5G HetNet using Multi-Agent Parameterized Deep Reinforcement learning," Physical Communication, vol. 61, Art. no. 102206, 2023, doi: 10.1016/J.PHYCOM.2023.102206.

[10] A. Hussain and P. Musilek, "Energy management of buildings with energy storage and solar photovoltaic: A diversity in experience approach for deep reinforcement learning agents," Energy and AI, vol. 15, Art. no. 100313, 2024, doi: 10.1016/J.EGYAI.2023.100313.

[11] Z. Guan, Z. Wang, Y. Cai, and X. Wang, "Deep reinforcement learning based efficient access scheduling algorithm with an adaptive number of devices for federated learning IoT systems," Internet of Things, vol. 24, Art. no. 100980, 2023, doi: 10.1016/J.IOT.2023.100980.

[12] A. Gao, S. Lu, R. Xu, Z. Li, B. Wang, S. Zhu, et al., "Deep reinforcement learning based planning method in state space for lunar rovers," Engineering Applications of Artificial Intelligence, vol. 127, no. PB, Art. no. 107287, 2024, doi: 10.1016/J.ENGAPPAI.2023.107287.

[13] Z. Wang, M. Goudarzi, M. Gong, and R. Buyya, "Deep Reinforcement Learning-based scheduling for optimizing system load and response time in edge and fog computing environments," Future Generation Computer Systems, vol. 152, pp. 55-69, 2024, doi: 10.1016/J.FUTURE.2023.10.012.

[14] M. E. Najafabadi and F. Haghighat, "Transfer learning for occupancy-based HVAC control: A data-driven approach using unsupervised learning of occupancy profiles and deep reinforcement learning," Energy and Buildings, vol. 300, Art. no. 113637, 2023, doi: 10.1016/J.ENBUILD.2023.113637.

[15] C. Liu, Z. Sheng, S. Chen, H. Shi, and B. Ran, "Longitudinal control of connected and automated vehicles among signalized intersections in mixed traffic flow with deep reinforcement learning approach," Physica A: Statistical Mechanics and its Applications, vol. 629, Art. no. 129189, 2023, doi: 10.1016/J.PHYSA.2023.129189.

[16] J. Wu and D. Li, "Modeling and maximizing information diffusion over hypergraphs based on deep reinforcement learning," Physica A: Statistical Mechanics and its Applications, vol. 629, Art. no. 129193, 2023, doi: 10.1016/J.PHYSA.2023.129193.

[17] I. Kim, S. Kim, and D. You, “Non-iterative generation of an optimal mesh for a blade passage using deep reinforcement learning,” Computer Physics Communications, vol. 294, Art. no. 108962, 2024, doi: 10.1016/J.CPC.2023.108962.

[18] J. Deng, M. Eklund, S. Sierla, J. Savolainen, H. Niemistö, T. Karhela, et al., “Deep reinforcement learning for fuel cost optimization in district heating,” Sustainable Cities and Society, vol. 99, Art. no. 104955, 2023, doi: 10.1016/J.SCS.2023.104955.

[19] H. Hou, S. N. A. Jawaddi, and A. Ismail, “Energy efficient task scheduling based on deep reinforcement learning in cloud environment: A specialized review,” Future Generation Computer Systems, vol. 151, pp. 214-231, 2024, doi: 10.1016/J.FUTURE.2023.10.002.

[20] R. H. Randhawa, N. Aslam, M. Alauthman, M. Khalid, and H. Rafiq, “Deep reinforcement learning based Evasion Generative Adversarial Network for botnet detection,” Future Generation Computer Systems, vol. 150, pp. 294-302, 2024, doi: 10.1016/J.FUTURE.2023.09.011.

[21] Z. Wang, H. Huang, J. Tang, and L. Hu, “A deep reinforcement learning-based approach for autonomous lanechanging velocity control in mixed flow of vehicle group level,” Expert Systems With Applications, vol. 238, no. PD, Art. no. 122158, 2024, doi: 10.1016/J.ESWA.2023.122158.

[22] J. Liu, Z. Zhou, W. Hong, and J. Shi, “Two-dimensional iterative learning control with deep reinforcement learning compensation for the non-repetitive uncertain batch processes,” Journal of Process Control, vol. 131, Art. no. 103106, 2023, doi: 10.1016/J.JPROCONT.2023.103106.

[23] H. Yuan, D. Li, and J. Wang, “Integrated robust navigation and guidance for the kinetic impact of near-earth asteroids based on deep reinforcement learning,” Aerospace Science and Technology, vol. 142, no. PB, Art. no. 108666, 2023, doi: 10.1016/J.AST.2023.108666.

[24] P. Chen and W. Fan, “Identifying critical nodes via link equations and deep reinforcement learning,” Neurocomputing, vol. 562, Art. no. 126871, 2023, doi: 10.1016/J.NEUCOM.2023.126871.

[25] A. Alferidi, M. Alsolami, B. Lami, and S. Bensiama, “Design and implementation of an indoor environment management system using a deep reinforcement learning approach,” Ain Shams Engineering Journal, vol. 14, no. 11, Art. no. 102534, 2023, doi: 10.1016/J.ASEJ.2023.102534.

[26] A. Mattioni, S. Zoboli, B. Mavkov, D. Astolfi, V. Andrieu, E. Witrant, et al., “Enhancing deep reinforcement learning with integral action to control tokamak safety factor,” Fusion Engineering and Design, vol. 196, Art. no. 114008, 2023, doi: 10.1016/J.FUSENGDES.2023.114008.

[27] S. Agrawal, S. Dubey, and K. J. Naik, “Deep reinforcement learning for forecasting fish survival in open aquaculture ecosystem,” Environmental Monitoring and Assessment, vol. 195, no. 11, pp. 1389, 2023, doi: 10.1007/S10661-023-11937-9.

[28] J. Xue, F. Han, B. K. V. Oorschot, G. Clifton, and D. Fan, “Exploring storm petrel pattering and sea-anchoring using deep reinforcement learning,” Bioinspiration & biomimetics, vol. 18, no. 6, Art. no. 066016, 2023, doi: 10.1088/1748-3190/AD00A2.

[29] K. Guo, R. Liu, G. Duan, J. Liu, and P. Cao, “Research on dynamic decision-making for product assembly sequence based on Connector-Linked Model and deep reinforcement learning,” Journal of Manufacturing Systems, vol. 71, pp. 451-473, 2023, doi: 10.1016/J.JMSY.2023.09.015.