Mech. Sci., 17, 671–684, 2026
https://doi.org/10.5194/ms-17-671-2026

Mechanical Sciences
Open Access

Curriculum-learning-driven hierarchical multi-agent deep reinforcement learning for collaborative scheduling in complex supply chain networks

Jingya Dong[1], Han Zhao[1], Suyi Zhao[1], Yijie Wang[1], Mengfan Guo[1], Chunhe Song[2], and Mingliang Xu[1]

[1]School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou 450001, China
[2]Institute of AI for Industries, Nanjing 211100, China

Correspondence: Chunhe Song (chhsong@iaii.ac.cn)

Received: 26 April 2026 – Revised: 19 May 2026 – Accepted: 29 May 2026 – Published: 25 June 2026

Abstract. With the growing scale, heterogeneity, and dynamic uncertainty of modern supply chain networks, collaborative scheduling across order assignment, manufacturer selection, and logistics operations has become increasingly critical and challenging because of strong inter-stage coupling, high decision complexity, and dynamic operational constraints. To address these challenges, this paper investigates the joint optimization problem of order assignment, heterogeneous manufacturer selection, and logistics vehicle scheduling in dynamic supply chain collaborative networks and proposes a curriculum-learning-driven hierarchical multi-agent deep reinforcement learning framework (CH-MADRL) for coordinated scheduling in complex environments. First, the joint optimization problem is formulated as a hierarchical multi-agent Markov decision process to capture the hierarchical dependencies and dynamic interactions among order assignment, heterogeneous manufacturer selection, and logistics vehicle scheduling, which establishes a unified modeling foundation for multi-stage collaborative scheduling. Second, based on this formulation, a hierarchical multi-agent deep reinforcement learning architecture is developed to decompose the tightly coupled high-dimensional joint scheduling problem into three correlated sub-problems, enabling coordinated optimization across different stages of the supply chain. Third, a constraint-progressive adaptive curriculum-learning mechanism is developed to facilitate policy learning under dynamic constraints, where a stage-conditioned dynamic masking mechanism regulates feasible action spaces, and a dual-gated promotion strategy stabilizes transitions across curriculum stages. Simulation experiments demonstrate that the proposed method surpasses baseline approaches in scheduling performance, training efficiency, and cross-scale generalization capability.

1 Introduction

The global supply chain system is undergoing a significant evolution characterized by heightened uncertainty. The superposition of multiple factors, including trade frictions, geopolitical risks, public health emergencies, and demand fluctuations, imposes continuous shocks on the stable operation of cross-regional supply networks (de Lima et al., 2022). Under macro environments where volatility, uncertainty, complexity, and ambiguity coexist, supply chain management is transitioning from traditional models dominated by informatization and standardized processes to digital-intelligent systems featuring data-driven approaches, intelligent perception, and autonomous decision making (Hao and Demir, 2025; Yu et al., 2026). Meanwhile, enterprises impose higher requirements on the response speed, collaborative efficiency, and resilience of supply chain systems. Multi-stage collaborative optimization and real-time scheduling thus emerge as critical challenges in scenarios with dynamically arriving orders, heterogeneous resource capabilities, and strong coupling between manufacturing and logistics, particularly for intelligent decision making in complex supply chains.

Researchers worldwide conduct systematic studies on collaborative scheduling in complex supply chains, and existing methods generally fall into three categories: mathemati-

cal programming and decomposition optimization methods, heuristic and metaheuristic methods, and data-driven methods based on reinforcement learning (Wu et al., 2024). The first two formulate order production and distribution as optimization models subject to process flows, capacity limits, delivery deadlines, and transportation constraints, solving them via exact algorithms or approximate search techniques. Vié et al. (2025) propose a matheuristic framework that combines local search with exact methods to coordinate production and distribution, optimizing total costs of order fulfillment and shipment. Subsequent work has extended production and logistics scheduling from static single-stage settings to complex environments with equipment failures, dynamic order arrivals, and multi-objective trade-offs (Atasagun and Karaoğlan, 2024). These methods remain effective for small- to medium-scale problems or relatively stable environments. However, continuous order arrivals, intensified cross-stage coupling, and frequent disturbances expose critical limitations: model reconstruction is costly, solution times escalate rapidly, and schedule stability deteriorates.

Reinforcement learning (RL) handles dynamic task arrivals, resource state changes, and frequent disturbances by interacting with the environment, learning from trial and error, and optimizing long-term returns. Unlike traditional methods that require explicit modeling and static solution processes, RL updates decision policies through continuous feedback. Zhang et al. (2024a) formulate dynamic flexible scheduling problems with transportation time constraints as multi-agent collaborative decision processes, coordinating machine agents with job agents. Shi et al. (2025) propose a nested hierarchical deep reinforcement learning method for production and logistics collaborative scheduling in dynamic flexible job shops. Li et al. (2025a) develop a real-time scheduling method using multi-agent deep reinforcement learning (MARL) for production and logistics coordination under large-scale dynamic order arrivals. For logistics operations, Li et al. (2024) propose a hierarchical framework for dynamic conflict-free automated guided vehicle (AGV) scheduling in automated container terminals. The above studies validate RL, particularly multi-agent and hierarchical variants, as effective for complex scheduling and logistics coordination (Shi et al., 2025; Li et al., 2025a, 2024). However, existing methods primarily address single workshops, single logistics systems, or local resource coordination. Extending these methods to full-chain supply chain scenarios is difficult because strong coupling among order assignment, heterogeneous manufacturer selection, and vehicle scheduling leads to exponential growth of the joint state and action space. Furthermore, dynamic task arrivals, resource competition, and spatiotemporal dependencies compound challenges, including sparse rewards, inefficient exploration, and training non-stationarity. Recent studies note that, although RL and MARL have advanced rapidly in dynamic scheduling, practical deployment remains limited by scalability, training stability, and cross-scenario generalization (Zhang et al., 2024b; Hady et al., 2025).

For RL in complex supply chain collaborative scheduling, curriculum learning (CL) provides a feasible way to organize training under conditions of cold-start difficulties, sparse rewards, and training instability. Training tasks are arranged progressively so that agents can first learn basic decision rules and then adapt to larger networks, stronger stage couplings, and tighter operational constraints. Results from related scheduling studies support the usefulness of this training paradigm. Iklassov et al. (2023) demonstrate that curriculum-based training performs better than direct training for large-scale instances in job shop scheduling, while Narvekar et al. (2020) show that task ordering based on difficulty and transferability improves reinforcement learning performance. A similar situation arises in supply chain collaborative scheduling, where increasing network scale is usually accompanied by stronger interactions among decision stages and more complex constraints. However, existing CL methods rely on predefined difficulty orders or rough task selection heuristics, and only limited attention has been paid to the gradual regulation of feasible action spaces during training. This issue is particularly important in large discrete decision spaces, where invalid action masking can reduce infeasible exploration and improve training stability (Huang and Ontañón, 2022). Recent work has started to examine automatic curriculum design in sparse-reward cooperative multi-agent reinforcement learning (Chen et al., 2021), but research that combines curriculum learning, hierarchical multi-agent coordination, and dynamic feasible-action control for complex supply chain scheduling is still limited.

To address these limitations, we investigate the joint optimization of order assignment, heterogeneous manufacturer selection, and logistics vehicle scheduling in dynamic supply chain collaborative networks. First, we formulate this problem as a hierarchical multi-agent Markov decision process that captures the coupling among order, manufacturing, and logistics decisions. Then, we propose a curriculum-learning-driven hierarchical multi-agent deep reinforcement learning framework (CH-MADRL), which coordinates scheduling across stages through hierarchical decomposition, curriculum progression, and dynamic constraint handling. Furthermore, we design a stage-conditioned dynamic masking mechanism and a dual-gated promotion strategy to gradually expand feasible action spaces and stabilize curriculum progression. Finally, experiments on dynamic environments validate the proposed method. The main contributions of this paper are as follows:

1. A CH-MADRL framework is proposed, which decouples order assignment, heterogeneous manufacturer selection, and logistics vehicle scheduling into three hierarchically correlated sub-problems to alleviate the combinatorial complexity caused by high-dimensional joint decision making.

2. A constraint-progressive adaptive curriculum-learning mechanism is designed, which achieves smooth transition across scale tasks and improves policy training stability through complexity evolution paths from simple to complex and dual-gated promotion strategies.

3. A stage-conditioned dynamic masking mechanism is constructed, which embeds task dependencies, resource availability, and scale boundaries into the action selection process, achieving progressive unlocking of feasible action spaces and reducing invalid exploration.

4. Comparative experiments show that the proposed method outperforms baselines in scheduling performance, training efficiency, and cross-scale generalization.

The remainder of this paper is organized as follows. Section 2 introduces the related literature and techniques. Section 3 provides a formal description and mathematical modeling of the research problem. Section 4 elaborates on the proposed CH-MADRL framework in detail. Section 5 presents experimental results. Section 6 concludes and outlines future research directions.

2 Related work

2.1 Collaborative logistics scheduling methods in supply chains

Coordination between production and logistics stages constitutes the basis of supply chain collaborative scheduling. Expansion from single factories and distribution chains to multi-facility, multi-resource, and multi-node networks has shifted research from separate optimization of orders, production, and distribution toward integrated production and distribution and joint manufacturing and transportation optimization. Liang et al. (2025) examine collaborative production and material distribution, showing that independent optimization of either stage lowers system-wide efficiency under strong spatiotemporal coupling. On this basis, Homayouni and Fontes (2021) investigate joint production and transportation scheduling problems in flexible job shops, Yao et al. (2024) incorporate limited AGV resources into flexible job shop models to formalize manufacturing and transportation collaboration, and Sidki et al. (2025) develop batch-centric mixed-integer programming for integrated production and pipeline distribution. These studies demonstrate that production and logistics collaborative scheduling has evolved from traditional single-stage optimization to integrated optimization oriented toward multi-resource coupling.

Regarding solution methodologies, existing research includes exact algorithms, heuristic algorithms, and metaheuristic algorithms. Exact methods guarantee optimality for small-scale instances, yet computational complexity escalates rapidly with order scale, node quantity, and transportation resources, precluding real-time decision making in dynamic scenarios (Yao et al., 2024; Sidki et al., 2025). Consequently, researchers generally adopt learning-augmented heuristic and metaheuristic methods. Uzunoglu et al. (2023) combine learning-augmented mechanisms with local search for parallel serial batch processing machine scheduling; Amirteimoori et al. (2023) propose parallel heuristic methods for hybrid job shop scheduling with conflict-free AGV path planning; Pérez et al. (2023) fuse GRASP, genetic algorithms, and learning mechanisms for supply chain scheduling; Karimi and Alinia (2025) introduce energy consumption factors into multi-factory supply chain scheduling, expanding modeling dimensions. However, most of these studies are still centered on single factories or limited collaborative settings, and research on integrated scheduling across order assignment, manufacturer selection, and vehicle dispatching in dynamic supply chain networks remains relatively scarce.

2.2 Dynamic-scheduling methods based on deep reinforcement learning

In recent years, deep reinforcement learning (DRL) has emerged as an important direction in dynamic-scheduling research because it learns sequential decision policies end to end in high-dimensional state spaces. Liu et al. (2022) formulate dynamic flexible job shop scheduling as Markov decision processes, demonstrating DRL's real-time response advantages under random order arrivals. Subsequent work extends from single-agent methods to graph neural networks, multi-agent reinforcement learning, and hierarchical reinforcement learning. Ngwu et al. (2026) and Hamou et al. (2025) review DRL applications for dynamic job shop scheduling and supply chain production scheduling, respectively; Liu and Huang (2023) combine graph neural networks with DRL for dynamic job shop scheduling; Wang et al. (2025a) propose attention-enhanced reinforcement learning methods for flexible job shop scheduling with transportation constraints; Li et al. (2025b) propose an end-to-end decentralized scheduling framework for dynamic distributed heterogeneous flow shops; Wang et al. (2025b) extend hierarchical multi-agent deep reinforcement learning to flexible job shops with transportation constraints. Meanwhile, multi-agent deep reinforcement learning (MADRL) extends from workshop scheduling to broader collaborative decision problems such as online scheduling in assembly systems, distributed hybrid flow shops, and multi-echelon inventory management (Kaven et al., 2024; Di et al., 2024; Xu et al., 2025; Liu et al., 2025). These studies demonstrate that DRL gradually evolves from single-workshop internal scheduling toward complex scenarios involving cross-resource, cross-node, and multi-agent collaboration.

However, existing DRL research exhibits two limitations in full-chain collaborative scenarios within complex supply chains. First, most methods address single workshops, single logistics systems, or local resource allocation, lacking

integrated modeling of order decomposition, heterogeneous manufacturer collaboration, and logistics vehicle scheduling. Second, as supply chain scale and constraint complexity increase, the joint state and action space expands rapidly, causing cold-start difficulties, sparse rewards, inefficient exploration, and multi-agent training non-stationarity. To alleviate these difficulties, CL is introduced into RL training processes. Its fundamental idea improves sample utilization efficiency and policy transfer capability through task organization from easy to difficult (Lin et al., 2025). Applications span flexible job shop scheduling (Lu et al., 2025) and hierarchical reinforcement learning for dynamic AGV scheduling, automated terminal task allocation, and port equipment coordination (Hu et al., 2025a; Chang et al., 2025; Hu et al., 2025b; Yang et al., 2026). However, existing CL methods remain limited to single-dimensional difficulty progression or empirical stage switching without systematic coupling of the synchronous expansion of order scale, manufacturing nodes, logistics capacity, and hierarchical multi-agent decision making with dynamic action space constraints. Based on this, this paper proposes a hierarchical MADRL framework incorporating constraint-progressive curriculum evolution to address cross-level joint decision problems in complex supply chain collaborative scheduling.

3 Problem formulation

This paper investigates a multi-stage decision optimization problem in collaborative supply chain logistics, covering three interrelated decision dimensions: order task decomposition, manufacturing resource allocation, and transportation resource scheduling. Unlike the classical flexible job shop scheduling problem (FJSP), limited to single-facility operations, our collaborative setting introduces strict cross-node logistics constraints and spatiotemporal coupling. This integration exponentially expands the joint action space, making the proposed adaptive curriculum mechanism computationally essential to overcome the resulting high-dimensional exploration challenges. For formal modeling, the problem is represented as a standardized supply chain instance with scale $n \times m \times l$, where n, m, and l denote the number of orders, heterogeneous manufacturers, and logistics vehicles, respectively. Let $\mathcal{O} = \{O_1, \ldots, O_n\}$ represent the order set, $\mathcal{M} = \{M_1, \ldots, M_m\}$ represent the manufacturer set, and $\mathcal{L} = \{L_1, \ldots, L_l\}$ represent the vehicle set. Each order O_i contains H_i operations with process precedence, denoted as $K_i = \{o_{i,1}, \ldots, o_{i,H_i}\}$, where operation $o_{i,k}$ must be completed before $o_{i,k+1}$. Each operation $o_{i,k}$ can only be assigned to a manufacturer $j \in \mathcal{M}$ with the required processing capability, and its processing time is denoted as $p_{i,k,j}$. When adjacent operations are executed by different manufacturers, vehicle $v \in \mathcal{L}$ is required to complete inter-node transportation. The transportation time consists of empty vehicle dispatch time from the current position to the pickup point and loaded transportation time from the pickup point to the target manufacturing node. An operation can only start processing when its predecessor operation is completed, materials arrive via transportation, and the corresponding manufacturer resource becomes available. This paper takes minimizing the system makespan as the optimization objective:

$$\min C_{\max} = \min(\max_{i \in \mathcal{O}} C_{i,H_i}), \tag{1}$$

where C_{i,H_i} is determined by the manufacturing and transportation processes at each stage of the order. For order i, the completion time of the final operation is as follows:

$$C_{i,H_i} = \max\left[T_j^{\text{avail}}\left(\max\left(C_{i,H_i-1}, T_{\text{v}}^{\text{free}} + T_{\text{v},i*}^{\text{empty}}\right) + T_{i,j}^{\text{trans}}\right)\right] + p_{i,H_i,j}. \tag{2}$$

For each operation, the processing start time is constrained by the completion of its predecessor, the arrival of transported materials, and the availability of the assigned manufacturer. The completion time is determined by the start time and the corresponding processing duration. Accordingly, the final order completion time is jointly affected by manufacturing capacity constraints, process precedence constraints, and transportation time constraints. For modeling purposes, the following assumptions are adopted:

1. Manufacturer resources are exclusive and non-preemptive, meaning that each manufacturer can process only one operation at a time and processing cannot be interrupted.

2. Tasks follow a strict serial flow of processing, transportation, and reprocessing, and all operations must be executed in the prescribed order.

3. When two adjacent operations are assigned to different manufacturers, inter-node transportation must be completed by a logistics vehicle, and the corresponding transportation time is nonzero.

4. Logistics vehicles are subject to exclusivity constraints, meaning that each vehicle can execute only one transportation task at a time.

5. Orders arrive randomly over time, and fluctuations in actual operational efficiency are determined by manufacturer fulfillment reputation q_i and logistics vehicle service quality q_{v}.

4 CH-MADRL framework

To address high-dimensional decision complexity in large-scale supply chain collaborative scheduling, together with

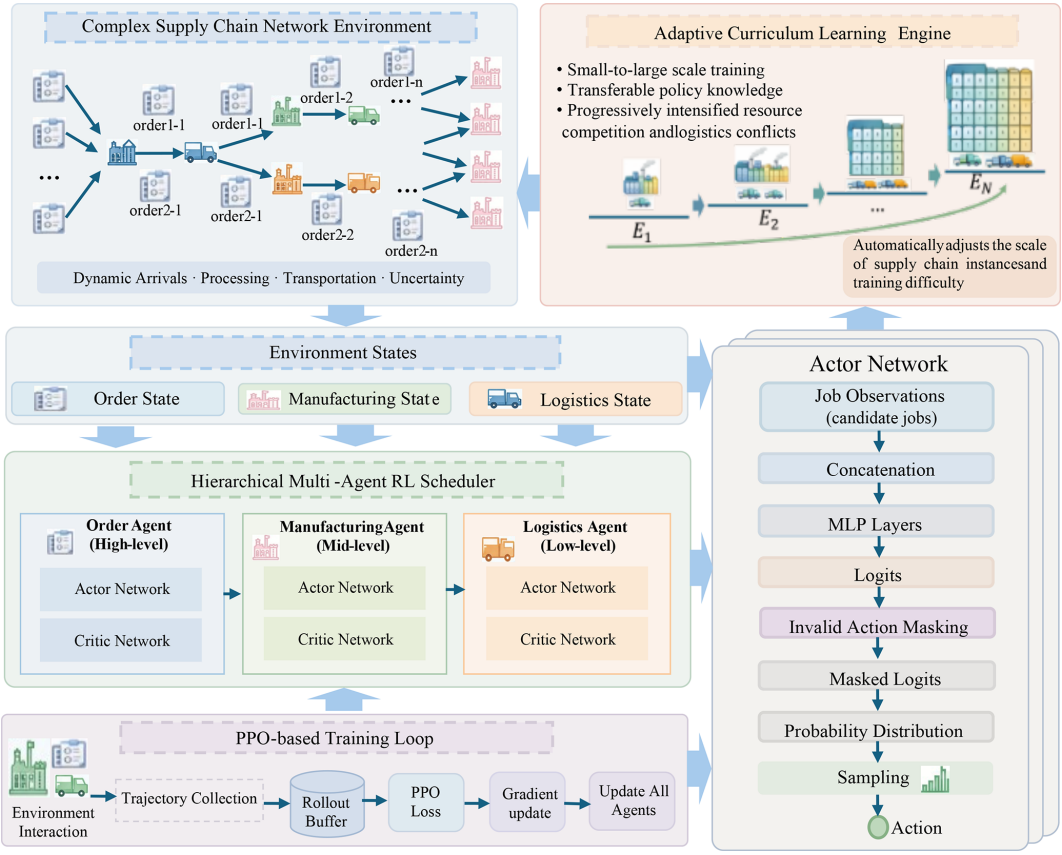


Figure 1. Architecture of the proposed CH-MADRL framework.

cold-start difficulty and inefficient exploration under complex spatiotemporal constraints, this paper proposes the CH-MADRL framework. As shown in Fig. 1, the framework combines hierarchical task decomposition, dynamic masking, and adaptive curriculum learning to reduce decision complexity, incorporate multi-stage dependencies and spatiotemporal constraints, and improve convergence efficiency and generalization performance.

4.1 Hierarchical multi-agent MDP model construction

Considering the dynamic, partially observable, and multi-resource-coupled nature of supply chain networks, the scheduling problem is formulated as a hierarchical decentralized partially observable Markov decision process. The model is defined by the tuple $\langle \mathcal{F}, S, A, P, R, \Omega, \gamma \rangle$, where $\mathcal{F} = \{F_{\text{ord}}, F_{\text{mfg}}, F_{\text{log}}\}$ denotes the heterogeneous agent set for order assignment, manufacturer selection, and logistics scheduling; S denotes the global state space describing real-time information on resource nodes and order flows; Ω denotes the joint observation space composed of local observations $\{o_{\text{job}}, o_{\text{mfg}}, o_{\text{log}}\}$ at different decision layers; A denotes the joint action space composed of discrete action subspaces at three hierarchies: order assignment, manufacturer selection, and logistics scheduling; P denotes the state transition function governed by operation precedence, transportation delays, and other physical process constraints; R denotes the reward function designed to promote global collaborative optimization; and $\gamma \in [0, 1)$ denotes the discount factor.

4.1.1 State space

Following the partially observable modeling framework, the state space is decomposed into three hierarchical local subspaces: the order agent state S_t^{O}, the manufacturer agent state S_t^{M}, and the logistics agent state S_t^{L}. The state features of the three agent layers are listed in Table 1, where all temporal variables are quantified in minutes to ensure dimensional consistency.

4.1.2 Action space

The joint action space is decomposed as $\mathcal{A} = \mathcal{A}^{\text{O}} \times \mathcal{A}^{\text{M}} \times \mathcal{A}^{\text{L}}$. A dynamic mask M_t is introduced to project action probability distributions onto feasible domains and enforce physical feasibility constraints.

Order assignment subspace ($\mathcal{A}^{\text{O}}$). The action $a_t^{\text{O}} \in \{1, \ldots, n\}$ selects a high-priority order for processing. The mask $M_t^{\text{O}} \in \{0, 1\}^n$ is activated only when an order has not

Table 1. Definitions of state space features.

State	Symbol	Description
Order agent state	i	Order index
	k	Index of the operation currently pending processing
	$C_{i,k-1}$	Actual completion time of the preceding operation
	Loc_i	Current geographical location of the order
	Prog_i	Order scheduling progress, computed as $(k-1)/H_i \times 100\,\%$, where H_i is the total number of operations for order i
	T_i^{cum}	Cumulative actual processing and logistics time incurred by the order
	$\overline{P}_{i,k}$	Average expected processing time of the current pending operation k
	$\overline{T}_i^{\mathrm{rem}}$	Estimated average cumulative processing duration of the order's remaining operations
Manufacturer Agent State	j	Manufacturer index
	q_j	Production quality grade of the manufacturer
	N_j^{ops}	Cumulative total number of operations processed by the manufacturer
	T_j^{avail}	Earliest available time of the manufacturer upon completion of preceding tasks
	$p_{i^*,k,j}$	Interaction feature: standard processing time of the current operation k of the selected order i^* at manufacturer j
	$T_{i^*,j}^{\mathrm{trans}}$	Interaction feature: estimated in-transit transportation time for the selected order i^* from its current location Loc_{i^*} to manufacturer j
Logistics Agent State	v	Logistics vehicle index
	q_{v}	Service quality grade of the logistics vehicle
	$N_{\mathrm{v}}^{\mathrm{tasks}}$	Cumulative number of transportation tasks executed by the vehicle
	$T_{\mathrm{v}}^{\mathrm{free}}$	Release time of the logistics vehicle upon completion of its previous task
	$\mathrm{Loc}_{\mathrm{v}}$	Current geographical location of the vehicle
	$T_{\mathrm{v},i^*}^{\mathrm{empty}}$	Interaction feature: estimated dispatching time for vehicle v to travel empty from its current location $\mathrm{Loc}_{\mathrm{v}}$ to the location of the selected order i^*

been delivered and satisfies the prerequisite readiness conditions:

$$M_{\mathrm{t}}^{\mathrm{O}}[i] = \mathbb{I}(\delta_i^{\mathrm{done}} = 0) \cdot \mathbb{I}(\mathrm{Ready}(i, \mathrm{t})), \tag{3}$$

where $\mathbb{I}(\cdot)$ is the indicator function, and $\mathrm{Ready}(i, \mathrm{t})$ indicates material and preceding-operation readiness for order i.

Manufacturer selection subspace ($\mathcal{A}^{\mathrm{M}}$). The action $a_{\mathrm{t}}^{\mathrm{M}} \in \{1, \ldots, m\}$ selects a manufacturer for the current operation. The mask $M_{\mathrm{t}}^{\mathrm{M}} \in \{0,1\}^m$ enforces process qualification constraints based on the capability matrix $\mathbf{T}_{\mathrm{cap}}$, retaining only manufacturers with the required capability for operation k:

$$M_{\mathrm{t}}^{\mathrm{M}}[j] = \mathbb{I}(\mathbf{T}_{\mathrm{cap}}[k, j] \neq \varnothing). \tag{4}$$

Logistics scheduling subspace ($\mathcal{A}^{\mathrm{L}}$). The action $a_{\mathrm{t}}^{\mathrm{L}} \in \{1, \ldots, l\}$ dispatches a logistics vehicle. The mask $M_{\mathrm{t}}^{\mathrm{L}} \in \{0,1\}^l$ reflects resource availability and excludes vehicles under failure or maintenance:

$$M_{\mathrm{t}}^{\mathrm{L}}[v] = \mathbb{I}(\mathrm{Status}_{\mathrm{v}}(t) \in \{\mathrm{Idle}, \mathrm{Active}\}). \tag{5}$$

4.1.3 Reward function

For makespan minimization, a terminal sparse reward provides only limited learning signals, which often leads to temporal credit assignment difficulties and slow convergence in long-horizon scheduling tasks. To improve training efficiency, a potential-based dense reward is adopted, where the potential function is defined as $\Phi(S_{\mathrm{t}}) = -C_{\max}(S_{\mathrm{t}})$. The immediate reward at step t is therefore defined as follows:

$$r_{\mathrm{t}} = \Phi(S_{\mathrm{t}}) - \Phi(S_{t-1}) = C_{\max}(S_{t-1}) - C_{\max}(S_{\mathrm{t}}). \tag{6}$$

Since the makespan is monotonically non-decreasing during scheduling and satisfies $C_{\max}(S_{\mathrm{t}}) \geq C_{\max}(S_{t-1})$, the immediate reward always satisfies $r_{\mathrm{t}} \leq 0$. Actions that increase the critical path receive an immediate penalty, while actions that leave it unchanged receive zero reward. For an episode of length T, the cumulative return can be written in telescoping form:

$$\begin{aligned} R &= \sum_{t=1}^{T} r_{\mathrm{t}} = \sum_{t=1}^{T} [C_{\max}(S_{t-1}) - C_{\max}(S_{\mathrm{t}})] \\ &= C_{\max}(S_0) - C_{\max}(S_T) = -C_{\max}^{\mathrm{final}}. \end{aligned} \tag{7}$$

4.2 Constraint-progressive adaptive curriculum learning mechanism

To improve learning in large-scale supply chain scheduling with dynamic task arrivals, sparse rewards, and high-dimensional constraints, a constraint-progressive adaptive

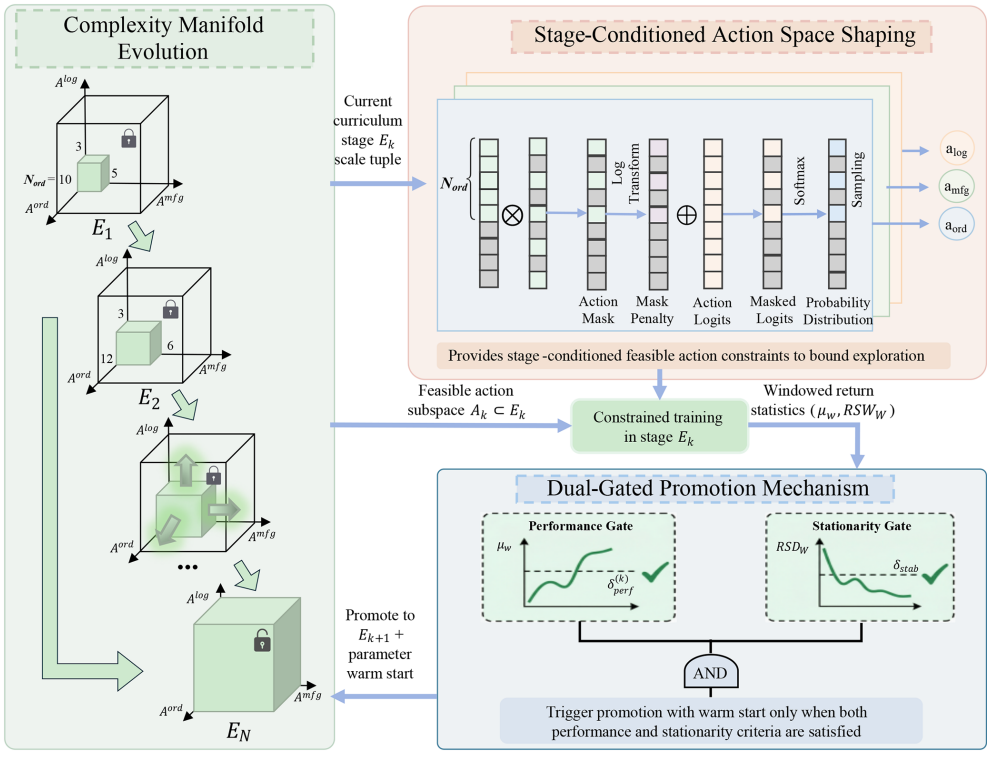


Figure 2. Constraint-progressive adaptive curriculum learning mechanism.

curriculum is introduced. As shown in Fig. 2, training starts with simple supply chain instances and gradually progresses to more complex ones by expanding network topology and task scale. Dynamic masking then releases feasible actions stage by stage, reducing invalid exploration in large decision spaces. Promotion to the next stage is allowed only after policy stabilization, which helps maintain stable training and reliable convergence.

4.2.1 Supply chain complexity manifold evolution

To reduce exploration difficulty in high-dimensional environments, the curriculum-learning process is defined as a discrete sequence of environment sets, $\mathcal{C} = \{\mathcal{E}_1, \mathcal{E}_2, \ldots, \mathcal{E}_k\}$, where $k = 6$. At each stage $\mathcal{E}_k$, the supply chain network is specified by the tuple $\Omega_k = \langle N_k^{\text{ord}}, N_k^{\text{mfg}}, N_k^{\text{log}} \rangle$, which denotes the numbers of orders, manufacturers, and logistics vehicles, respectively. The complexity progression is organized into three stages.

Foundation logic construction stage ($\mathcal{E}_1$). The initial scale is set to $\Omega_1 = \langle 10, 5, 3 \rangle$. This stage is used to support rapid policy initialization in a low-dimensional setting and to learn basic coordination rules, including order assignment, resource availability, and task dependencies.

Scale continuous increment stage ($\mathcal{E}_2 \sim \mathcal{E}_5$). The network scale is increased step by step without changing the underlying scheduling mechanism so as to strengthen policy robustness and generalization under growing resource competition and spatiotemporal conflicts.

Target scale collaboration stage ($\mathcal{E}_6$). The environment reaches the target scale $\Omega_6 = \langle 20, 10, 5 \rangle$. At this stage, the policy is trained under a larger state space and stronger spatiotemporal coupling to form a stable collaborative scheduling strategy.

4.2.2 Dynamic masking-driven progressive unlocking of action space

In multi-agent systems, directly facing target-scale configurations with preset global maximum action space $\mathcal{A}_{\max}$ (whose dimensions are determined by $N_{\max}^{\text{ord}}$, $N_{\max}^{\text{mfg}}$, and $N_{\max}^{\text{log}}$) causes an extremely severe curse of dimensionality. To this end, this paper decouples the curriculum evolution mechanism from underlying control logic, formalizing it as a dimensionality release process of decision feasible domains: at the early curriculum stage $\mathcal{E}_k$, the system utilizes the dynamic masking mechanism M_t to construct hard constraint boundaries, forcibly masking all high-dimensional action indices exceeding the current network scale, that is, strictly ensuring $M_t[a] \equiv 0, \forall a > N_k$. As the curriculum advances toward higher-stage manifolds, masking constraints are gradually released, and effective decision boundaries of agents are progressively unlocked.

This mechanism reduces the global optimization problem into nested subspace searches, mitigating exploration disorientation and improving training stability.

4.2.3 Adaptive promotion mechanism based on dual-gating

This paper designs an adaptive promotion mechanism based on dual-gating, where promotion is triggered only when both indicators simultaneously satisfy the required conditions within sliding window W. The sliding-window size was set to $W = 20$ episodes, determined through preliminary sensitivity experiments balancing promotion responsiveness and policy convergence stability.

1. Performance lower-bound gating: The average return μ_W within the current window must exceed the preset baseline threshold $\delta_{\text{perf}}^{(k)}$:

$$\mu_W = \mathbb{E}[G_{t-W:t}] > \delta_{\text{perf}}^{(k)}. \tag{8}$$

   The threshold $\delta_{\text{perf}}^{(k)}$ is determined from the optimal solutions of heuristic algorithms at the corresponding scale. This criterion requires the DRL policy to reach a scheduling level beyond that of traditional rules before entering the next stage.

2. Policy stationarity detection: Due to the high variance of DRL exploration, mean-based indicators can be biased by outliers or occasional high returns. Therefore, a relative standard deviation criterion is introduced:

$$\text{RSD}_W = \frac{\sigma(G_{t-W:t})}{|\mu_W|} < \delta_{\text{stab}}, \tag{9}$$

   where $\sigma(\cdot)$ is the standard deviation function, and δ_{stab} is the preset stationarity tolerance factor, which is set to 0.05 through preliminary tuning experiments to balance sensitivity to policy fluctuations and training efficiency. The dual-gating mechanism effectively avoids misjudgment risks from single indicators through joint constraints of performance and stationarity, ensuring robustness of curriculum stage transitions. The complete adaptive curriculum evolution process is detailed in Algorithm

4.3 Hierarchical proximal policy optimization algorithm

This paper adopts a hierarchical proximal policy optimization (H-PPO) algorithm based on the actor–critic architecture, with feasible-domain hard-constraint projection and a global collaborative loss design. At the network level, agents at each hierarchy use structurally symmetric actor–critic networks. The actor network employs a multilayer perceptron (MLP) to extract state features and introduces a hard-constraint projection layer to ensure physical feasibility. Using the dynamic mask M_t defined in Sect. 4.1.2, log-domain

Algorithm 1 Adaptive curriculum evolution algorithm based on stationarity detection.

Input: Initial policy network parameters θ, ϕ; curriculum environment set $\mathcal{C} = \{E_1, E_2, \ldots, E_K\}$
Output: Converged optimal parameters θ_k^* for each stage; final global optimal policy parameters θ_K^*
Initialization: K – total number of curriculum stages; W – sliding-window size; $\delta_{\text{perf}}(k)$ – performance promotion threshold for stage k; δ_{stab} – stationarity (RSD) threshold; G – sliding-window return sequence.
for curriculum stage $k = 1$ **to** K **do**
  Initialize environment configuration $\Omega_k \leftarrow \text{Config}(E_k)$; **If** $k > 1$ **then** $\theta \leftarrow \theta_{k-1}^*$
  while True **do**
    Perform one PPO training iteration; obtain current episode total return G_t
    Update sliding-window return sequence G; compute current window mean return $\mu_W = \mathbb{E}[G_{t-W:t}]$
    Compute current policy relative standard deviation $\text{RSD}_W = \sigma(G_{t-W:t})/|\mu_W|$
    if $\mu_W > \delta_{\text{perf}}(k)$ **And** $\text{RSD}_W < \delta_{\text{stab}}$ **then**
      Save current converged optimal policy parameters $\theta_k^* \leftarrow \theta$
      Break (promotion condition satisfied; proceed to next stage E_{k+1})
    end if
  end while
end for

renormalization is applied to the output logits z_t, with $z_t' = z_t + \log(M_t + \epsilon)$. This operation assigns zero probability to invalid actions and blocks gradient propagation along illegal paths, thereby improving search efficiency in constrained spaces.

The training process follows a decentralized execution and centralized evaluation paradigm. All agents share the unified differential reward r_t in Sect. 4.1.3, and local policies are jointly optimized for makespan minimization. Parameter updates use the following composite loss function:

$$\begin{aligned} L_{\text{Total}} = & -\mathbb{E}_t[\min(\rho_t A_t, \text{clip}(\rho_t, 1-\epsilon, 1+\epsilon)A_t)] \\ & + c_1 L_{\text{VF}}(\phi) - c_2 S[\pi_\theta], \end{aligned} \tag{10}$$

where ρ_t denotes the ratio between new and old policies, A_t denotes the advantage function, and $S[\pi_\theta]$ denotes policy entropy.

5 Experimental results and analysis

The proposed method is implemented with PyTorch and runs on a workstation equipped with an AMD Ryzen 7 8845H CPU, 16 GB RAM, and an NVIDIA GeForce RTX 4060 Laptop GPU with 8 GB VRAM. All random seeds are fixed to ensure reproducibility, and each reported result is averaged over 100 independently sampled test instances per scale configuration.

5.1 Training efficiency and curriculum efficacy validation

To evaluate the effect of the curriculum mechanism, two training settings are considered: progressive training with CH-MADRL and direct training with standard DRL. CH-MADRL is trained from $\mathcal{E}_1$ ($10 \times 5 \times 3$) and promoted stage by stage under the dual-gating criteria defined by $\delta_{\text{perf}}(k)$ and δ_{stab}, with parameter warm start used at each stage transition. Standard DRL is trained end to end on the target scale $20 \times 10 \times 5$.

Supply chain networks involve coupled order, resource, and logistics flows, making curriculum design an important factor in training efficiency. Two curriculum expansion strategies are compared: single-dimension expansion, which increases only the number of orders while keeping the manufacturer scale and logistics capacity fixed, and mixed-dimension expansion, which jointly increases order quantity, manufacturer scale, and logistics capacity.

As illustrated in Fig. 3a, standard DRL shows pronounced oscillation in the early training stage and converges to a suboptimal makespan of about 1500, reflecting cold-start difficulty. CH-MADRL exhibits brief performance drops near stage transition points but recovers quickly after each transition and converges to a lower makespan. Figure 3b shows that mixed-dimension expansion leads to smoother stage transitions and faster convergence, while single-dimension expansion produces noticeable performance drops when later stages introduce abrupt parameter changes.

5.2 Overall scheduling performance comparison

To evaluate the scalability of the framework, experiments are conducted across six gradient scales from $10 \times 10 \times 5$ to $20 \times 10 \times 5$, with 100 independent random test instances generated for each configuration. To simulate dynamic disturbances and resource heterogeneity in supply chains, critical environment parameters are independently sampled from uniform distributions:

1. Dynamic order arrivals: Release times $A_i \sim \mathcal{U}[30, 300]$.
2. Resource capability fluctuations: Manufacturer processing quality q_j and logistics transportation quality $q_{\text{v}} \sim \mathcal{U}[0.1, 1.0]$.
3. Logistics time-lag distribution: Inter-node transportation times $t_{\text{trans}} \sim \mathcal{U}[1, 10]$.

The average makespan comparison of various algorithms across different scales is presented in Fig. 4. Results demonstrate that CH-MADRL maintains performance advantages across all test scales. Particularly in the highest-complexity $20 \times 10 \times 5$ large-scale scenario, CH-MADRL converges to an average makespan of 1764.82, representing a 2.3 % reduction compared with standard DRL (1806.67), validating the scalability of the proposed method in high-dimensional state spaces.

The reliability of supply chain scheduling solutions is typically measured through statistical dispersion of the results. The boxplot distribution in Fig. 5 further reveals that CH-MADRL solutions exhibit more compact distribution patterns with fewer outliers. Specifically, at the $20 \times 10 \times 5$ scale, CH-MADRL achieves a makespan standard deviation of 257.36, lower than that of the baseline (272.58), representing a 5.6 % reduction.

5.3 Horizontal comparison with baseline algorithms

To comprehensively evaluate the performance advantages of CH-MADRL, this subsection conducts comparative experiments with three representative algorithm categories:

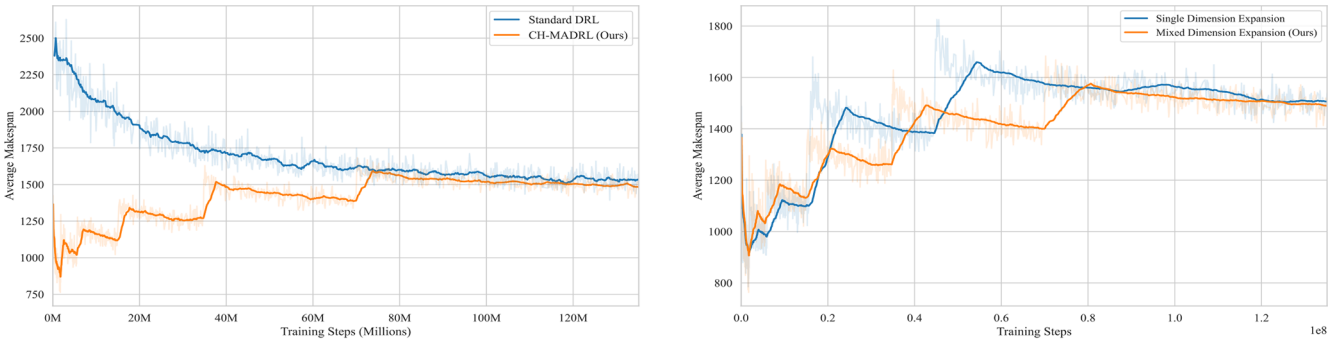
1. Composite heuristic rules (rule 1–4): Four composite greedy strategies are constructed for multi-stage collaborative characteristics of supply chains. Four classical order priority rules (EST, MOPNR, SPT, MTWR) are combined with, respectively, manufacturer and logistics assignment rules based on shortest processing time (SPT), serving as local-optimal baselines.
2. Metaheuristic algorithm (GA): Standard GA is introduced as a classical global search baseline to evaluate whether deep reinforcement learning methods can approach or surpass traditional evolutionary computation methods in solution quality.
3. Deep reinforcement learning baselines (A2C, DRL + IL): The on-policy algorithm A2C is introduced to validate the advantages of the PPO architecture adopted in this paper regarding update stability. DRL + IL incorporating expert demonstration is introduced to validate whether the proposed curriculum-learning mechanism better facilitates agents in breaking through expert experience limitations and exploring globally superior strategies compared to passive imitation learning.

The three-dimensional bar chart in Fig. 6 intuitively demonstrates the scheduling results of all algorithms for the first 20 instances at the $20 \times 10 \times 5$ scale. CH-MADRL achieves the lowest makespan in the vast majority of instances, outperforming rule algorithms based on local greedy strategies.

Figure 7 demonstrates solution quality distributions of various algorithms across different test sets through boxplots. Results indicate that CH-MADRL exhibits comprehensive advantages in the vast majority of test scenarios: lowest makespan mean, most compact box range, and minimal outliers.

5.4 Zero-shot generalization capability testing

To evaluate the model's zero-shot cross-domain generalization, the converged models from Sect. 5.1, trained at the



(a) (b)

Figure 3. Training efficiency and curriculum efficacy validation: **(a)** convergence curve comparison between CH-MADRL and standard DRL; **(b)** training convergence comparison between mixed-dimension expansion and single-dimension expansion strategies.

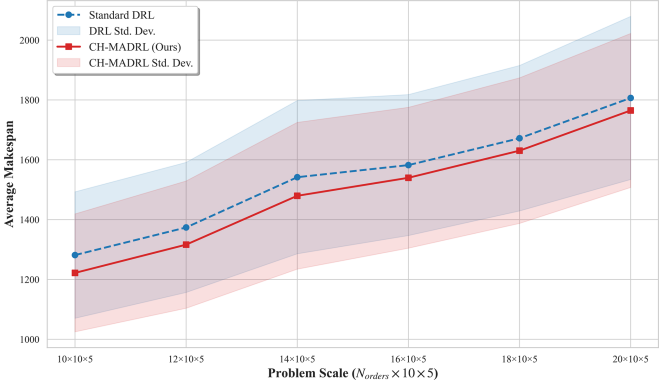


Figure 4. Performance trend analysis of CH-MADRL versus standard DRL in multi-scale scalability tests.

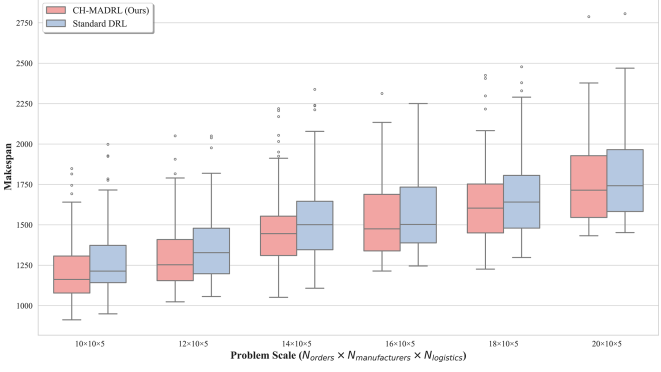


Figure 5. Boxplot comparison of makespan distributions across different scales for CH-MADRL versus standard DRL.

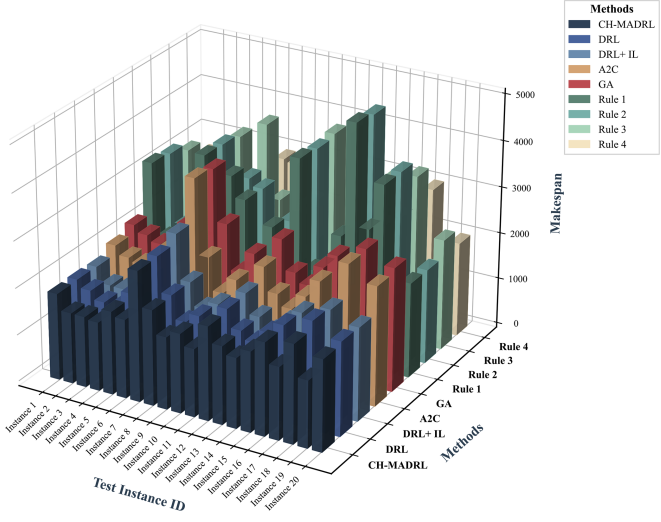


Figure 6. Makespan comparison of all methods for 20 test instances at $20 \times 10 \times 5$ scale.

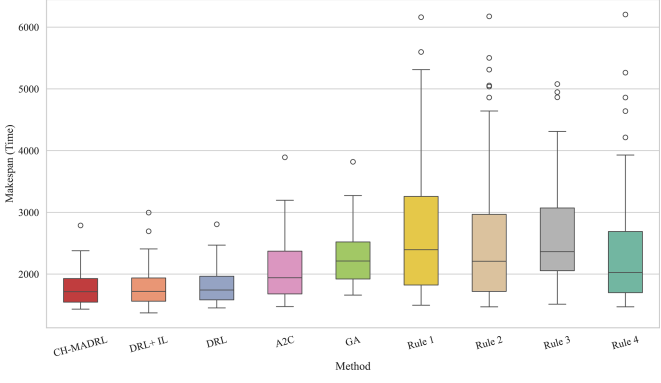


Figure 7. Statistical boxplot analysis of different algorithms at $20 \times 10 \times 5$ scale

$20 \times 10 \times 5$ scale, are directly transferred to four unseen instances ranging from $13 \times 6 \times 4$ to $30 \times 15 \times 7$ without fine-tuning. The tests include two scenarios: a fixed-order scenario, where the order quantity remains constant to assess scalability under expanded dimensions, and a random-order scenario, where the order quantity fluctuates to assess robustness under dynamic loads.

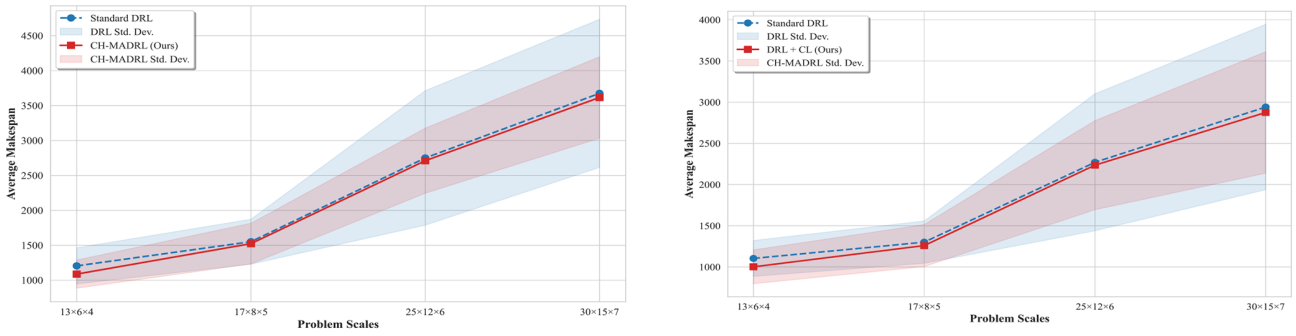


Figure 8. Zero-shot generalization performance evaluation at different unseen problem scales: **(a)** fixed-order-quantity scenario; **(b)** random-order-quantity scenario.

Table 2. Zero-shot generalization performance comparison at different unseen problem scales.

Problem scale	Scenario	Standard DRL	CH-MADRL (ours)	Improvement
$13 \times 6 \times 4$	Fixed	1205.00 ± 258.48	1087.38 ± 202.11	$\uparrow 9.76\,\%$
	Random	1103.47 ± 217.78	1001.62 ± 205.49	$\uparrow 9.23\,\%$
$17 \times 8 \times 5$	Fixed	1551.18 ± 321.50	1523.32 ± 291.68	$\uparrow 1.80\,\%$
	Random	1300.97 ± 256.79	1259.36 ± 255.15	$\uparrow 3.20\,\%$
$25 \times 12 \times 6$	Fixed	2749.87 ± 962.86	2710.55 ± 466.19	$\uparrow 1.43\,\%$
	Random	2270.23 ± 831.72	2235.53 ± 541.18	$\uparrow 1.53\,\%$
$30 \times 15 \times 7$	Fixed	3674.75 ± 1060.23	3615.21 ± 582.97	$\uparrow 1.62\,\%$
	Random	2941.36 ± 1004.26	2875.22 ± 737.65	$\uparrow 2.25\,\%$

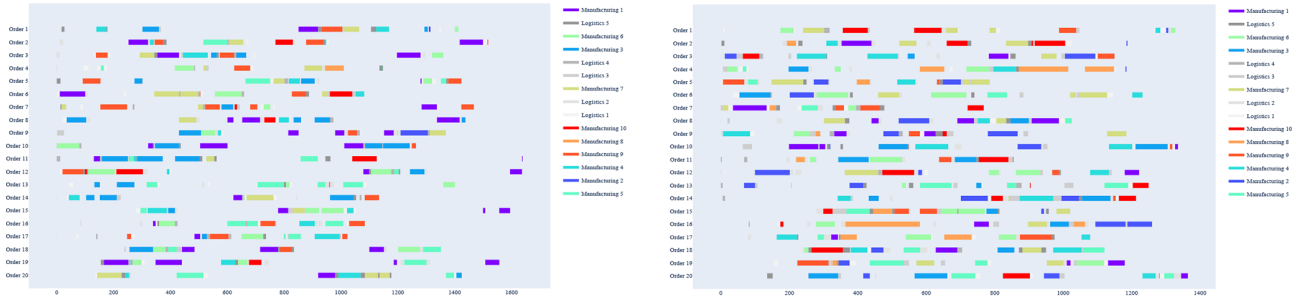
As illustrated in Fig. 8 and Table 2, CH-MADRL consistently outperforms standard DRL in terms of makespan and variance as problem dimensionality expands stepwise. Specifically, in in-distribution interpolation generalization tests (exemplified by $13 \times 6 \times 4$ and other scales), the model achieves performance improvements of 9.76 % and 9.23 % under fixed- and random-order conditions, respectively, indicating that curriculum learning effectively drives agents to extract general representations of underlying collaborative logic rather than overfitting to specific training distributions. Furthermore, in out-of-distribution extrapolation generalization tests (facing the $30 \times 15 \times 7$ scenario exceeding the training scale), CH-MADRL maintains performance advantages of 1.62 % and 2.25 %, with standard deviations consistently below the control group. These results comprehensively demonstrate that the proposed framework possesses high policy stability and structural generalization capability when facing cross-domain and out-of-bound scales.

5.5 Interpretability and microscopic behavior analysis

To examine the decision behavior of the agents at a finer scale, Fig. 9 compares the Gantt-chart schedules produced by the two methods for a $20 \times 10 \times 5$ instance. The traditional rule-based method, shown in Fig. 9a, is limited by locally greedy decisions and does not account for the cumulative effects of cross-node logistics delays. As a result, manufacturing and logistics flows become temporally misaligned, leading to task fragmentation and resource idleness. By contrast, CH-MADRL, shown in Fig. 9b, produces a more compact spatiotemporal schedule. Through time window coordination and dynamic load balancing, the agents improve the alignment between processing, transportation, and subsequent processing stages, avoid node congestion, and reduce the overall makespan.

6 Conclusions

This paper proposes CH-MADRL for collaborative scheduling in complex supply chain networks. A hierarchical multi-agent Markov decision process is constructed for the joint optimization of order assignment, heterogeneous manufacturer selection, and logistics vehicle scheduling. To improve training under dynamic constraints, a constraint-progressive adaptive curriculum is introduced, together with a stage-conditioned dynamic masking mechanism and a dual-gated promotion strategy. Experimental results show that CH-MADRL achieves better convergence, lower makespan, and



(a) (b)

Figure 9. Microscopic scheduling behavior visualization comparison between CH-MADRL and rule methods at $20 \times 10 \times 5$ scale: **(a)** rule-based method; **(b)** CH-MADRL.

stronger zero-shot generalization across different problem scales.

The current work still has several limitations. Experimental evaluation is conducted in simulated environments with predefined scale progression. Although the framework incorporates dynamic order arrivals and resource quality fluctuations, it does not fully capture all disruption patterns encountered in real supply chain systems, such as supplier defaults and logistics network interruptions. Extending the framework to handle such real-world disruptions remains an important direction for future work. In addition, the current framework considers only makespan minimization and does not address other objectives, such as operational cost, carbon emissions, and delivery reliability. Future research will extend CH-MADRL to multi-objective scheduling and explore graph neural-network-based representations to improve scalability in larger and more complex supply chain scenarios.

Data availability. Data will be made available on reasonable request to the corresponding author.

Author contributions. Jingya Dong conceptualized this study, developed the methodology, and wrote the original draft. Han Zhao implemented the software and contributed to validation. Suyi Zhao curated the data and contributed to visualization. Yijie Wang conducted the investigation and validation. Mengfan Guo conducted the investigation and revised the paper. Chunhe Song acquired the funding and provided supervision. Mingliang Xu provided project administration.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. The authors bear the ultimate responsibility for providing appropriate place names. Views expressed in the text are those of the authors and do not necessarily reflect the views of the publisher.

Acknowledgements. During the initial preparation of this paper, the authors used ChatGPT and Kimi for language polishing and readability improvement. The authors reviewed and revised all AI-assisted outputs and take full responsibility for the final content.

Financial support. This work was supported in part by the National Key RD Program of China under grant no. 2024YFB3311600, in part by the Key Science and Technology Research Project of Henan Province (grant no. 252102211074), and in part by the Key Scientific Research Projects of Henan Higher Education Institutions under grant no. 25A520012.

Review statement. This paper was edited by Pengyuan Zhao and reviewed by two anonymous referees.

References

Amirteimoori, A., Tirkolaee, E. B., Simic, V., and Weber, G.-W.: A parallel heuristic for hybrid job shop scheduling problem considering conflict-free AGV routing, Swarm Evol. Comput., 79, 101312, https://doi.org/10.1016/j.swevo.2023.101312, 2023.

Atasagun, G. C. and Karaoğlan, İ.: Integrated production and outbound distribution scheduling problem with multiple facilities/vehicles and perishable items, Appl. Soft Comput., 166, 112144, https://doi.org/10.1016/j.asoc.2024.112144, 2024.

Chang, X., Jia, X., and Hu, H.: Energy-efficient and self-adaptive AGV scheduling approach based on hierarchical reinforcement learning for flexible shop floor, Comput. Ind. Eng., 205, 111140, https://doi.org/10.1016/j.cie.2025.111140, 2025.

Chen, J., Zhang, Y., Xu, Y., Ma, H., Yang, H., Song, J., Wang, Y., and Wu, Y.: Variational automatic curriculum learning for sparse-reward cooperative multi-agent problems, in: Advances in Neural Information Processing Systems (NeurIPS), 34, 2021.

Di, Y., Deng, L., and Zhang, L.: A collaborative-learning multi-agent reinforcement learning method for distributed hybrid flow

shop scheduling problem, Swarm Evol. Comput., 91, 101764, https://doi.org/10.1016/j.swevo.2024.101764, 2024.

de Lima, F. A., Seuring, S., and Sauer, P. C.: A systematic literature review exploring uncertainty management and sustainability outcomes in circular supply chains, Int. J. Prod. Res., 60, 6013–6046, 2022.

Hady, M. A., Hu, S., Pratama, M., Cao, Z., and Kowalczyk, R.: Multi-agent reinforcement learning for resources allocation optimization: a survey, Artif. Intell. Rev., 58, 354, https://doi.org/10.1007/s10462-025-11340-5, 2025.

Hamou, K. A. B., Jarir, Z., and Elfirdoussi, S.: Using machine learning for production scheduling problems in the supply chain: A review, Comput. Ind. Eng., 206, 111243, https://doi.org/10.1016/j.cie.2025.111243, 2025.

Hao, X. and Demir, E.: Artificial intelligence in supply chain management: enablers and constraints in pre-development, deployment, and post-development stages, Prod. Plan. Control, 36, 748–770, 2025.

Homayouni, S. M. and Fontes, D. B. M. M.: Production and transport scheduling in flexible job shop manufacturing systems, J. Global Optim., 79, 463–502, 2021.

Hu, H., Liu, L., and Yang, X.: A deep reinforcement learning framework for real-time joint task assignment and storage allocation problems considering random tasks in automated container terminals, Comput. Ind. Eng., 111544, https://doi.org/10.1016/j.cie.2025.111544, 2025.

Hu, Y., Wang, M., Min, R., Liu, J., Lukinykh, V. F., Tang, S., and Zhao, D.: Coordinated scheduling optimization of quay cranes and AGVs in automated container terminals, Comput. Oper. Res., 182, 107147, https://doi.org/10.1016/j.cor.2025.107147, 2025.

Huang, S. and Ontañón, S.: A closer look at invalid action masking in policy gradient algorithms, in: International FLAIRS Conference Proceedings, 35, https://doi.org/10.32473/flairs.v35i.130584, 2022.

Iklassov, Z., Medvedev, D., Solozabal Ochoa de Retana, R., and Takac, M.: On the study of curriculum learning for inferring dispatching policies on the job shop scheduling, in: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 5350–5358, https://doi.org/10.24963/ijcai.2023/594, 2023.

Karimi, N. and Alinia, S.: Towards a sustainable future: Integrating energy efficiency in multi-factory supply chain scheduling, Process Integration and Optimization for Sustainability, 9, 1425–1443, 2025.

Kaven, L., Huke, P., Göppert, A., and Schmitt, R. H.: Multi agent reinforcement learning for online layout planning and scheduling in flexible assembly systems, J. Intell. Manuf., 35, 3917–3936, https://doi.org/10.1007/s10845-023-02309-8, 2024.

Li, H., Gao, L., Fan, Q., Li, X., and Han, B.: An end-to-end decentralised scheduling framework based on deep reinforcement learning for dynamic distributed heterogeneous flowshop scheduling, Int. J. Prod. Res., 63, 4368–4388, https://doi.org/10.1080/00207543.2024.2449240, 2025.

Li, S., Fan, L., and Jia, S.: A hierarchical solution framework for dynamic and conflict-free AGV scheduling in an automated container terminal, Transport. Res. C-Emer., 165, 104724, https://doi.org/10.1016/j.trc.2024.104724, 2024.

Li, Y., Li, X., and Gao, L.: Real-time scheduling for production-logistics collaborative environment using multi-agent deep reinforcement learning, Adv. Eng. Inform., 65, 103216, https://doi.org/10.1016/j.aei.2025.103216, 2025.

Liang, T., Zhou, L., and Jiang, Z.: Integrated scheduling of production and material delivery for the intelligent manufacturing system, Int. J. Prod. Res., 63, 882–903, 2025.

Lin, S., Mi, Q., and Gao, T.: A survey of curriculum learning in deep reinforcement learning, in: Proceedings of the 2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC), IEEE, 1141–1147, https://doi.org/10.1109/CCWC62904.2025.10903795, 2025.

Liu, C. L. and Huang, T. H.: Dynamic job-shop scheduling problems using graph neural network and deep reinforcement learning, IEEE T. Syst. Man. Cy.-S., 53, 6836–6848, 2023.

Liu, R., Piplani, R., and Toro, C.: Deep reinforcement learning for dynamic scheduling of a flexible job shop, Int. J. Prod. Res., 60, 4049–4069, 2022.

Liu, X., Hu, M., Peng, Y., and Yang, Y.: Multi-agent deep reinforcement learning for multi-echelon inventory management, Prod. Oper. Manag., 34, 1836–1856, https://doi.org/10.1177/10591478241305863, 2025.

Lu, C., Xiao, Y., Zhang, B., and Gao, L.: Curriculum reinforcement learning algorithm for flexible job shop scheduling problems, Journal of National University of Defense Technology, 47, 49–59, https://doi.org/10.11887/j.cn.202502004, 2025.

Narvekar, S., Peng, B., Leonetti, M., Sinapov, J., Taylor, M. E., and Stone, P.: Curriculum learning for reinforcement learning domains: A framework and survey, J. Mach. Learn. Res., 21, 1–50, 2020.

Ngwu, C., Liu, Y., and Wu, R.: Reinforcement learning in dynamic job shop scheduling: a comprehensive review of AI-driven approaches in modern manufacturing, J. Intell. Manuf., 37, 1093–1108, 2026.

Pérez, C., Climent, L., Nicoló, G., Arbelaez, A., and Salido, M. A.: A hybrid metaheuristic with learning for a real supply chain scheduling problem, Eng. Appl. Artif. Intell., 126, 107188, https://doi.org/10.1016/j.engappai.2023.107188, 2023.

Uzunoglu, A., Gahm, C., Wahl, S., and Tuma, A.: Learning-augmented heuristics for scheduling parallel serial-batch processing machines, Comput. Oper. Res., 151, 106122, https://doi.org/10.1016/j.cor.2022.106122, 2023.

Shi, J., Qiao, F., Liu, J., Ma, Y., Wang, D., and Ding, C.: Production-logistics collaborative scheduling in dynamic flexible job shops using nested-hierarchical deep reinforcement learning, Adv. Eng. Inform., 65, 103195, https://doi.org/10.1016/j.aei.2025.103195, 2025.

Sidki, M., Tchernev, N., Féniès, P., and Ren, L.: A monolithic batch-centric MILP approach for a real-world integrated production and pipeline distribution scheduling problem, Comput. Ind. Eng., 203, 111028, https://doi.org/10.1016/j.cie.2025.111028, 2025.

Vié, M. S., Zufferey, N., and Coelho, L. C.: A production and distribution scheduling matheuristic for reducing supply chain variations, Transport. Res. E-Log., 194, 103905, https://doi.org/10.1016/j.tre.2024.103905, 2025.

Wang, W., Zhang, Y., Wang, Y., Pan, G., and Feng, Y.: Hierarchical multi-agent deep reinforcement learning for dynamic flexible job-shop scheduling with transportation, Int. J. Prod. Res., 1–28, https://doi.org/10.1080/00207543.2025.2511239, 2025.

Wang, Y., Wang, R., Sun, J., Deng, F., Wang, G., and Chen, J.: Attention enhanced reinforcement learning for flexible job shop scheduling with transportation constraints, Expert Syst. Appl., 282, 127671, https://doi.org/10.1016/j.eswa.2025.127671, 2025.

Wu, C. C., Zhang, R. M., Zhao, P. Y., Li, L., and Zhang, D. G.: Curing simulation and data-driven curing curve prediction of thermoset composites, Sci. Rep., 14, 31860, https://doi.org/10.1038/s41598-024-83379-3, 2024.

Xu, W., Gu, J., Zhang, W., Gen, M., and Ohwada, H.: Multi-agent reinforcement learning for flexible shop scheduling problem: a survey, Front. Ind. Eng., 3, 1611512, https://doi.org/10.3389/fieng.2025.1611512, 2025.

Yang, L., Yang, Z., Bi, L., and Jiao, X.: Dynamic flexible job shop co-scheduling optimization based on graph neural network and deep reinforcement learning, Operations Research Perspectives, 16, 100379, https://doi.org/10.1016/j.orp.2026.100379, 2026.

Yao, Y., Liu, Q., Fu, L., Li, X., Yu, Y., Gao, L., and Zhou, W.: A novel mathematical model for the flexible job-shop scheduling problem with limited automated guided vehicles, IEEE T. Autom. Sci. Eng., 22, 7449–7462, https://doi.org/10.1109/TASE.2024.3356255, 2024.

Yu, H., Lv, M., Hu, B., Zhang, Y., and Zhao, P.: Review article: A review of control technologies for soft robots: from structural design to intelligent control, Mech. Sci., 17, 313–332, https://doi.org/10.5194/ms-17-313-2026, 2026.

Zhang, C., Juraschek, M., and Herrmann, C.: Deep reinforcement learning-based dynamic scheduling for resilient and sustainable manufacturing: A systematic review, J. Manuf. Syst., 77, 962–989, 2024.

Zhang, L., Yan, Y., and Hu, Y.: Dynamic flexible scheduling with transportation constraints by multi-agent reinforcement learning, Eng. Appl. Artif. Intell., 134, 108699, https://doi.org/10.1016/j.engappai.2024.108699, 2024.