LLM-Enhanced Dueling DQN for Multi-Factory Remanufacturing Optimization with Hybrid Disassembly Lines and Drone Delivery

Shujin Qin, Shaokang Dai, and Bin Hu

***Abstract*—This paper addresses a profit-maximization problem in multi-factory remanufacturing that integrates hybrid disassembly lines (straight and U-shaped) with cross-factory drone delivery. The problem is formulated as a mixed-integer program that employs AND/OR graphs to represent disassembly sequencing and incorporates workstation opening and assignment, precedence and conflict constraints, and drone routing costs. To solve the resulting high-dimensional combinatorial problem, we propose the LLM-enhanced Dueling Deep Q-Network (LLM-DUEL), which extends the standard Dueling DQN by incorporating a large language model fine-tuned with low-rank adaptation. The fine-tuned LLM generates feasible disassembly sequences, compressing the reinforcement learning action space, while hierarchical action design and a profit-increment reward mechanism further accelerate policy learning. Experiments on multiple synthetic case sets demonstrate that LLM-DUEL achieves faster convergence, improved stability, and higher objective values compared with DQN, DUEL, and PPO, while closely approaching CPLEX optima on tractable instances. These results suggest that domain-adapted LLMs can substantially enhance reinforcement learning by improving feasibility and efficiency in complex remanufacturing scheduling problems.**

***Key Words*—Large language model, dueling DQN, remanufacturing scheduling, hybrid disassembly line, drone delivery, LoRA.**

I. Introduction

With the accelerated global transition toward a circular economy, remanufacturing has become a cornerstone of sustainable development, functioning as a vital pathway for efficient resource recycling and environmental protection [1]. Unlike conventional recycling, remanufacturing preserves much of the product's embedded value by disassembling, refurbishing, and reassembling components, thereby reducing both material consumption and energy demand. This not only mitigates resource waste but also promotes the integration of ecological and economic benefits [2]. To maximize these advantages, optimizing the disassembly and recovery process has emerged as a prominent research direction in both academic and industrial communities [3, 4]. Traditional manual disassembly methods, while long utilized, are increasingly unable to meet the escalating demands of large-scale electronic waste processing due to their low efficiency, high labor costs, and susceptibility to human error [5]. In response, disassembly line balancing problems (DLBP) have been extensively studied, improving operational efficiency to some extent. However, current disassembly optimization research often abstracts away critical supply chain complexities, overlooking factors such as product allocation strategies, inter-factory coordination, and transportation logistics. These multi-level elements of the reverse supply chain significantly influence overall system efficiency, cost-effectiveness, and scalability [6, 7].

In the domain of multi-factory remanufacturing, the challenge of coordinating disassembly scheduling and resource allocation across geographically distributed plants becomes even more critical [8]. A well-designed coordination mechanism enables higher overall efficiency, better resource utilization, and improved responsiveness to demand fluctuations. Several representative studies have explored this problem space. For example, Jing et al. [9] investigated collaborative production planning in multi-factory environments with consideration of remanufacturing and stockout management, while Marandi et al. [10] developed an integrated supply chain model linking multiple factories in a network, focusing on the cross-regional flow of intermediate and finished products. Beyond inter-factory planning, other research has investigated intra-factory disassembly line balancing and scheduling. Qi et al. [11] introduced multi-factory disassembly line balancing frameworks to improve efficiency and reduce costs, whereas Guo [12] incorporated human factors, optimizing both the disassembly process and ergonomic aspects such as worker posture and line configuration. Similarly, Zhou [13] addressed worker fatigue and its impact on health and productivity, thereby extending the optimization scope of multi-factory remanufacturing systems to human-centric considerations. Together, these studies illustrate the breadth of approaches to multi-factory optimization, but they also underscore the lack of holistic solutions that unify operational scheduling, logistics, and human-in-the-loop factors.

One particularly underdeveloped dimension in remanufacturing optimization is transportation logistics. As highlighted by Lohmer and Lasch [14], transportation is often a decisive factor in remanufacturing supply chains, especially when long-distance movements are necessitated by capacity limitations or cost constraints. Yet, despite its importance, more than half of existing studies neglect transportation entirely. When

Manuscript received November 16, 2025; revised November 23 and November 30, 2025; accepted November 30, 2025. This article was recommended for publication by Associate Editor Shujin Qin upon evaluation of the reviewers' comments.

This work was supported in part by NSFC under Grant Nos. 61573089, 62073069 and 51405075, and in part by the Natural Science Foundation of Shandong Prov-ince under Grant ZR2019BF004.

S. Qin is with the School of Information and Technology, Shangqiu Normal University, Shangqiu 476000, China (e-mail: sjchin@vip.126.com).

S. Dai is with the Artificial Intelligence and Software College, Liaoning Petrochemical University, Fushun 113001, China (e-mail: 15751530086@163.com).

B. Hu is with the Department of Computer Science and Technology at Kean University, Union, NJ 07083, USA. (email: bhu@kean.edu).

Corresponding author: Shujin Qin

addressed, transportation has predominantly been modeled for finished product delivery (e.g., [10, 15]), with far fewer studies considering intra-factory logistics or inter-factory flows (e.g., [16, 17]). This narrow scope reinforces the widespread assumption that transportation introduces additional costs and burdens without offering potential performance improvements. However, more recent work, such as Gu et al.'s study of third-party logistics and multi-factory vehicle routing, has begun to demonstrate the value of rethinking transportation as an active enabler of system-wide optimization. This opens the door to innovative approaches for remanufacturing logistics that extend beyond static assumptions and embrace dynamic, flexible delivery strategies.

In this context, emerging drone delivery technologies provide a compelling new dimension to remanufacturing logistics. Drones offer agility, flexibility, and responsiveness in the delivery of disassembled parts across factories, enabling faster coordination and reducing bottlenecks. Their capacity to bypass ground congestion and operate on-demand positions them as an attractive complement to traditional logistics systems. However, their deployment also introduces novel challenges. These include spatiotemporal constraints, such as flight range, charging cycles, and payload limitations, as well as the need for dynamic route optimization in response to fluctuating demands and environmental uncertainties [18, 19]. Therefore, effectively integrating drones into multi-factory remanufacturing systems requires advanced optimization methods that can address both structural complexities and real-time adaptability.

Reinforcement learning (RL) has recently gained recognition as a powerful paradigm for dynamic, data-driven optimization in complex industrial environments [20]. Unlike heuristic or rule-based methods, RL achieves strategy optimization by enabling agents to learn through continuous interaction with their environments. This allows for adaptive decision-making that is especially valuable under uncertainty and high-dimensional state spaces. In manufacturing and logistics, RL has already been applied to various combinatorial optimization problems. Qin [21] tackled integrated production and distribution scheduling with heuristic-enhanced RL, minimizing tardiness and delivery costs. Cai [22] introduced a hybrid frog-leaping algorithm incorporating Q-learning, dynamically selecting search strategies to reduce production cycles. Liu [23] proposed a hierarchical distributed architecture for dynamic job shop scheduling, training Deep Q-Network (DQN) agents to capture relationships between production states and scheduling goals. Wang [24] advanced RL applications further by integrating cooperative memory agents with specialized encoding and decoding methods to handle multi-objective conflicts. These works collectively highlight the transformative potential of RL for real-time, flexible optimization in production and logistics systems.

Parallel to RL's rise, Large Language Models (LLMs) have demonstrated remarkable capabilities in semantic understanding, reasoning, and task decomposition, which are directly relevant to complex optimization challenges. LLMs' ability to parse structural knowledge and constraints positions them as valuable complements to RL systems, particularly in areas such as state space representation, action space reduction, and reward shaping [25, 26]. Recent studies suggest that integrating LLMs with RL can address exploration inefficiencies in high-dimensional spaces, accelerate learning, and inject domain knowledge into decision-making. However, a persistent limitation has been the insufficient fine-tuning of LLMs for domain-specific optimization tasks. Without adaptation, LLM-driven strategies may suffer from instability or weak generalization [27]. Fortunately, parameter-efficient fine-tuning methods, such as Low-Rank Adaptation (LoRA), have emerged as effective solutions. LoRA significantly reduces computational overhead while maintaining high task-specific performance [28]. This makes it particularly attractive for embedding LLM capabilities within industrial optimization frameworks.

This work distinguishes itself from prior LLM-RL integrations through its targeted application and methodology. While existing approaches often leverage LLMs for general purposes like reward shaping [26] or state representation [25], our method, LLM-DUEL, uniquely employs a fine-tuned LLM as a feasibility filter and action space compressor for combinatorial optimization. This focus is characterized by two key innovations: (1) We utilize LoRA to specialize a general-purpose LLM (Qwen2.5-7B) into a domain expert capable of generating feasible disassembly sequences, moving beyond the instability of prompt-based off-the-shelf models [27]. (2) Our core contribution lies in using the LLM to actively prune the vast action space of disassembly sequencing—a primary source of complexity in scheduling problems. This targeted strategy provides a more direct and efficient solution for our domain compared to broader LLM-RL hybrids.

Building on these insights, this paper proposes a novel LLM-enhanced Dueling Deep Q-Network (Dueling DQN) framework to solve the Multi-Factory Remanufacturing Optimization Problem with Hybrid Disassembly Line and Drone Delivery (MROP-HDD). Our framework integrates semantic reasoning from fine-tuned LLMs with RL's adaptive learning capacity, enabling the dynamic modeling of disassembly sequences, the compression of high-dimensional action spaces, and the acceleration of policy learning through knowledge-informed reward mechanisms. Specifically, the proposed approach leverages LoRA-based fine-tuning to efficiently incorporate domain-specific knowledge into the LLM, thereby ensuring both robustness and computational feasibility.

This work makes the following new contributions:

1) We establish a hybrid disassembly-line-based multi-product, multi-plant remanufacturing optimization problem incorporating drone delivery. A mixed-integer programming model is developed with the objective of maximizing overall system profit.
2) We exploit the semantic comprehension capabilities of LLMs to generate feasible disassembly sequences and reduce the dimensionality of the RL action space, directly addressing the inefficiency of exploration in traditional RL methods.
3) We design a novel RL exploration mechanism grounded in domain-informed reward functions. By fine-tuning the LLM with LoRA, we embed prior knowledge into

the learning process, thereby accelerating policy convergence and enhancing solution robustness.

The organization of the rest of this work is as follows. Section II provides a detailed description of the MROP-HDD, including its assumptions, AND/OR graph representation, and mathematical formulation. In Section III, the proposed LLM-enhanced Dueling DQN algorithm is presented, covering the LoRA-based fine-tuning strategy, action and state space design, and reward mechanism. Section IV illustrates the experimental setup, case generation, and comparative results with baseline algorithms. Finally, Section V concludes the paper and suggests future research directions.

II. Problem Description

A. Problem Statement

In modern manufacturing and remanufacturing processes, the design and optimization of disassembly lines are crucial for improving resource recovery efficiency and reducing operational costs. Particularly in multi-factory remanufacturing systems, efficiently organizing disassembly processes, maximizing resource recovery and profit, while controlling carbon emissions, is a challenging optimization problem.

This work focuses on the MROP-HDD in a drone delivery environment, exploring efficient optimization strategies by combining the practical application of straight-line and U-shaped hybrid disassembly lines. The MROP-HDD can be divided into three main stages: product allocation, disassembly decision-making, and delivery optimization, as shown in Fig. 1.

These three stages are interrelated; product allocation influences disassembly decisions, and disassembly decisions further determine the subsequent delivery optimization strategies. By designing optimal strategies for each stage, resource utilization efficiency can be improved, operational costs can be reduced, and profit maximization can be further achieved.

1) Product Allocation Stage

The recycling center receives various types of waste products. It allocates them reasonably based on product characteristics, market demand, disassembly factory processing capacity, resource allocation, and geographical location to optimize overall logistics costs. Once the product is allocated to a specific factory, the factory must further develop disassembly line allocation strategies to match different disassembly processes and resource conditions.

2) Disassembly Decision-Making Stage

The disassembly factory selects appropriate disassembly tasks based on product disassembly rules and production schedules, and allocates them to different disassembly lines. The layout of the disassembly line can be either straight-line or U-shaped, designed based on production efficiency and flexibility. In actual production, each disassembly line dynamically adjusts disassembly sequence and resource allocation based on task complexity, component quantity, and workstation load. Especially when recycling prices are uncertain, balancing workstation loads and optimizing task sequencing becomes key to improving disassembly efficiency and reducing costs.

3) Delivery Optimization Stage

After disassembly operations are completed, subassemblies must be transported via the drone system deployed at each factory to the corresponding manufacturing plants for the subsequent remanufacturing process. The optimization goal of this stage is to minimize transportation costs and maximize overall profit while meeting timeliness constraints. To achieve this, the following key factors must be considered: first, the drone delivery route planning, which must optimize flight trajectories to reduce total flight distance and energy consumption; second, the allocation and scheduling of transportation resources to enable efficient cross-factory coordination; third, the profit potential of the destination manufacturing factories, with priority given to those factories that can generate higher remanufacturing value, thereby enhancing the overall system benefits.

Through the collaborative optimization of these three stages, MROP-HDD can achieve efficient integration of recovery, disassembly, and delivery, improving the operation efficiency of the remanufacturing system and ultimately maximizing profit.

The integration of drone delivery is pivotal to this collaborative optimization, as it introduces a dynamic and flexible logistics capability that traditional ground vehicles lack. Drones enhance system responsiveness by bypassing ground congestion, enabling rapid, on-demand part delivery between factories, which is crucial for tightly coupled remanufacturing schedules. To establish a foundational and computationally tractable model for this novel integration, we begin with the simplifying assumption of a single drone per factory. This setup effectively captures the core challenges of coordinated route optimization and spatiotemporal constraints within the profit maximization objective. The proposed framework is, however, structurally extensible and lays the groundwork for future research incorporating heterogeneous drone fleets.

To establish the optimization model of the proposed MROP-HDD, we make the following assumptions:

- The matrices **D**, **A**, and **B** are known.
- Profit maximization is pursued, and products may not be fully disassembled.
- Each disassembly task can only be performed once.
- Each disassembly factory has one drone.
- The drone departs from the disassembly factory, passes through all the manufacturing plants requiring delivery, and then returns.

B. AND/OR Graph

Before product allocation to factories, it is crucial to understand the dismantling relationships among product subassemblies. The AND/OR graph, priority graph, and Petri net [29, 30] are common methods for modelling these relationships. The AND/OR graph employs a top-down modelling approach, effectively illustrating the dismantling relationships between tasks and subassemblies. For instance, the silicon oil fan clutch [31] serves as a case study modelled using an AND/OR graph. This fan clutch consists of the following components: (a) temperature sensor, (b) front cover, (c) valve axis, (d) valve, (e) driven plate, (f) seal ring, (g) active plate,

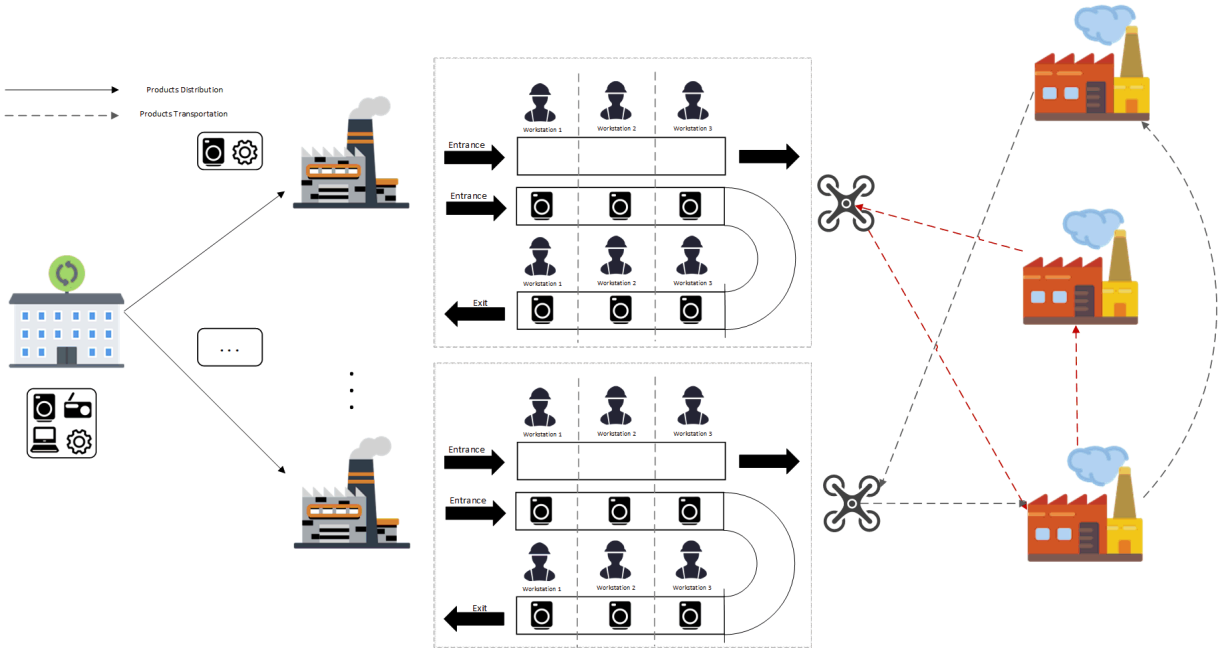


Fig. 1. Example diagram of MROP-HDD.

(h) bearing, (i) gasket, (j) back cover, and (k) active axis. Fig. 2 presents the schematic diagram of the silicon oil fan clutch, while Fig. 3 illustrates the corresponding AND/OR graph, comprising 11 components, 17 subassemblies, and 7 tasks.

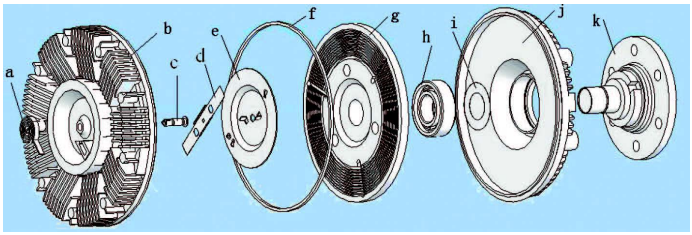


Fig. 2. A schematic of the silicon oil fan clutch.

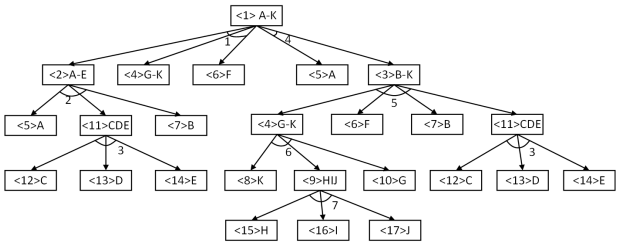


Fig. 3. An AND/OR diagram of the silicone oil fan clutch.

Fig. 3 illustrates that Subassembly 1 can be dismantled into Subassembly 2, Component 4, and Component 6 through Task 1, or it can be dismantled into Component and Subassembly 3 via Task 4. Moreover, Tasks 1 and 4 cannot be executed simultaneously, highlighting their conflict. Additionally, Task 1 must be completed before Task 2. Three matrices are employed to represent these relationships.

1) Incidence matrix

The correlation matrix $D = [d_{pij}]$ describes the disassembly relationships between tasks and subassemblies, where i represents the subassemblies, j represents the disassembly task, and p represents the product number.

$$d_{pij} = \begin{cases} 1, & \text{If performing task } j \text{ results in obtaining subassembly } i. \\ -1, & \text{If subassembly } i \text{ can be disassembled by task } j. \\ 0, & \text{Otherwise.} \end{cases}$$

2) Conflict matrix

The conflict matrix $R = [r_{pjq}]$ describes the conflicting relationship between two tasks, where j and q represent the disassembly task, p represent the product number.

$$r_{pjq} = \begin{cases} 1, & \text{if task } j \text{ of product } p \text{ has a conflicting relationship with } q; \\ 0, & \text{otherwise.} \end{cases}$$

3) Precedence matrix

The precedence matrix $S = [s_{pjq}]$, this work uses an immediately after the relationship.

$$s_{pjq} = \begin{cases} 1, & \text{if task } j \text{ of product } p \text{ has a precedence relationship with } q; \\ 0, & \text{otherwise.} \end{cases}$$

C. Mathematical Model

This section shows a linear model of the MROP, where the required notation and decision variables are defined as follows.

1) Notations:

Sets:

$\mathbb{F}$ Set of all disassembly factories,$\mathbb{F}$={1,2,...,F}.
$\mathbb{M}$ Set of all manufacture factories,$\mathbb{M}$={1,2,...,M}.
$\mathbb{P}$ Set of all End-of-life products,$\mathbb{P}$={1,2,...,P}.
$\mathbb{I}_p$ Set of all subassemblies/parts in product p, $\mathbb{I}_p = \{1, 2, \ldots, I_p\}$.
$\mathbb{J}_p$ Set of all tasks in production p, $\mathbb{J}_p = \{1, 2, \ldots, J_p\}$.
$\mathbb{W}_f^S$ Set of linear disassembly line workstations of the f-th disassembly factory, $\mathbb{W}_f^S = \{1, 2, \ldots, W_f^S\}$.
$\mathbb{W}_f^U$ Set of U-shaped disassembly line workstations of the f-th disassembly factory, $\mathbb{W}_f^U = \{1, 2, \ldots, W_f^U\}$.
$\mathbb{S}$ Set of edges of the U-shaped disassembly line workstation, $\mathbb{S} = \{1, 2, \}$.

Parameters:

v_{mpi} The price of the p-th product of the i-th subassemblies be purchased by the m-th factory.
c_{fmpi}^T The transportation cost of transferring subassembly or component i of product p from disassembly factory f to manufacturing factory m.
t_{pj} Time to execute the j-th task of the p-th product by a worker.
c_{fpj}^D The unite of time disassembly cost of task j for product p in factory f.
c_f^O The unite of time cost of operating disassembly factory f.
c_{fw}^S The fixed cost of operating straight-line workstation w in disassembly factory f.
c_{fw}^U The fixed cost of operating U-shaped workstation w in disassembly factory f.
N Number of path points. The starting point is factory f, and the subsequent path points are manufacturing factories m. $N = M + 1$.
d_{fij} Cost of drone delivery between various points.

2) Decision variables

$$z_{pf} = \begin{cases} 1, & \text{product } p \text{ is performed at the linear disassembly line of factory } f \\ 0, & \text{otherwise} \end{cases}$$

$$x_{pjfw}^S = \begin{cases} 1, & \text{if disassembly task } j \text{ in product } p \text{ is performed at the workstation } w \text{ in the straight-line disassembly line of factory } f \\ 0, & \text{otherwise} \end{cases}$$

$$x_{pjfws}^U = \begin{cases} 1, & \text{if disassembly task } j \text{ in product } p \text{ is performed at } s \text{ side of the workstation } w \text{ in the U-shaped disassembly line of factory } f \\ 0, & \text{otherwise} \end{cases}$$

$$y_f^S = \begin{cases} 1, & \text{Open the straight-line disassembly line of the disassembly factory } f \\ 0, & \text{otherwise} \end{cases}$$

$$y_f^U = \begin{cases} 1, & \text{Open the U-shaped disassembly line of the disassembly factory } f \\ 0, & \text{otherwise} \end{cases}$$

$$u_{fw}^S = \begin{cases} 1, & \text{open the workstation } w \text{ in the straight-line disassembly line of factory } f \\ 0, & \text{otherwise} \end{cases}$$

$$u_{fw}^U = \begin{cases} 1, & \text{open the workstation } w \text{ in the U-shaped disassembly line of factory } f \\ 0, & \text{otherwise} \end{cases}$$

$$\alpha_{fmpi} = \begin{cases} 1, & \text{if disassembly task } j \text{ in product } p \text{ is performed at the workstation } w \text{ in the linear disassembly line of factory } f \text{ by a worker} \\ 0, & \text{otherwise} \end{cases}$$

$$\beta_{fn} = \begin{cases} 1, & \text{the drones of } f\text{-th factory need to pass through the path } n\text{-th point} \\ 0, & \text{otherwise} \end{cases}$$

$$\gamma_{fij} = \begin{cases} 1, & \text{the drone of the } f\text{-th factory has a path from point } i \text{ to point } j \\ 0, & \text{otherwise} \end{cases}$$

T_f the linear disassembly line cycle time

u_{fi} indicating the position of the point i in the path

3) Objective of optimization

$$\begin{aligned} \max f = & \sum_{k\in\mathbb{K}} \sum_{m\in\mathbb{M}} \sum_{p\in\mathbb{P}} \sum_{i\in\mathbb{I}_p} v_{mpi}\alpha_{kmpi} - \sum_{k\in\mathbb{K}} \sum_{p\in\mathbb{P}} \sum_{j\in\mathbb{I}_p} \sum_{w\in\mathbb{W}_k^S} c_{pj}^D t_{pj} x_{pjkw}^S \\ & - \sum_{k\in\mathbb{K}} \sum_{p\in\mathbb{P}} \sum_{j\in\mathbb{J}_p} \sum_{w\in\mathbb{W}_k^U} \sum_{s\in\mathbb{S}} c_{pj}^D t_{pj} x_{pjkws}^U - \sum_{k\in\mathbb{K}} c_k^O T_k \\ & - \sum_{k\in\mathbb{K}} \sum_{w\in\mathbb{W}_k^S} c_{kw}^S u_{kw}^S - \sum_{k\in\mathbb{K}} \sum_{w\in\mathbb{W}_k^U} c_{kw}^U u_{kw}^U \\ & - \sum_{k\in\mathbb{K}} \sum_{i\in\mathbb{N}} \sum_{j\in\mathbb{N}, j\neq i} \gamma_{kij} d_{kij} \end{aligned} \tag{1}$$

The objective function 1 represents the profit in the remanufacturing process. The first term is the revenue from selling disassembled subcomponents or parts. The second term is the disassembly cost generated by the straight disassembly line. The third term is the disassembly cost generated by the U-shaped disassembly line. The fourth term is the operating cycle cost of opening the factory. The fifth and sixth terms are the workstation opening costs of the straight and U-shaped disassembly lines. The last term is the cost generated by drone delivery.

4) Constraints

$$\sum_{m\in\mathbb{M}} \alpha_{fmpi} \leq \sum_{w\in\mathbb{W}_f^S} \sum_{j\in\mathbb{J}_p} d_{pij} x_{pjfw}^S + \sum_{w\in\mathbb{W}_f^U} \sum_{j\in\mathbb{J}_p} \sum_{s\in\mathbb{S}} d_{pij} x_{pjfws}^U, \quad \forall f \in \mathbb{F}, \forall p \in \mathbb{P}, \forall i \in \mathbb{I}_p \setminus \{1\} \tag{2}$$

$$\sum_{f\in\mathbb{F}} z_{pf} = 1, \forall p \in \mathbb{P} \tag{3}$$

$$y_f^S + y_f^U \le 1, \forall f \in \mathbb{F} \tag{4}$$

$$z_{pf} \le y_f^S + y_f^U, \forall p \in \mathbb{P}, \forall f \in \mathbb{F} \tag{5}$$

$$u_{fw}^S \le y_f^S, \forall w \in \mathbb{W}_f^S, \forall f \in \mathbb{F} \tag{6}$$

$$u_{fw}^U \le y_f^U, \forall w \in \mathbb{W}_f^U, \forall f \in \mathbb{F} \tag{7}$$

$$\sum_{w \in \mathbb{W}_f^S} x_{pjfw}^S + \sum_{w \in \mathbb{W}_f^U} \sum_{s \in \mathbb{S}} x_{pjfws}^U \le z_{pf}, \quad \forall p \in \mathbb{P}, \forall \in \mathbb{F}, \forall j \in \mathbb{J}_p \tag{8}$$

$$x_{pjfw}^S \le u_{fw}^S, \forall p \in \mathbb{P}, \forall j \in \mathbb{J}_p, \forall f \in \mathbb{F}, \forall w \in \mathbb{W}_f^S \tag{9}$$

$$x_{pjfws}^U \le u_{fw}^U, \forall p \in \mathbb{P}, \forall j \in \mathbb{J}_p, \forall f \in \mathbb{F}, \forall w \in \mathbb{W}_f^U, \forall s \in \mathbb{S} \tag{10}$$

$$\sum_{f \in \mathbb{F}} \left(\sum_{w \in \mathbb{W}_f^S} x_{pjfw}^S + \sum_{w \in \mathbb{W}_f^U} \sum_{s \in \mathbb{S}} x_{pjfws}^U \right) \le 1, \quad \forall p \in \mathbb{P}, \forall j \in \mathbb{J}_p \tag{11}$$

$$\sum_{p \in \mathbb{P}} \sum_{j \in \mathbb{J}_p} t_{pj} x_{pjfw}^S \le T_f, \forall f \in \mathbb{F}, \forall w \in \mathbb{W}_f^S \tag{12}$$

$$\sum_{p \in \mathbb{P}} \sum_{j \in \mathbb{J}_p} \sum_{s \in \mathbb{S}} t_{pj} x_{pjfws}^U \le T_f, \forall f \in \mathbb{F}, \forall w \in \mathbb{W}_f^U \tag{13}$$

$$\sum_{w \in \mathbb{W}_f^S} w \left(x_{pj_1fw}^S - x_{pj_2fw}^S \right) + W_f^S \left(\sum_{w \in \mathbb{W}_f^S} x_{pj_2fw}^S - 1 \right) \le 0, \quad \forall f \in \mathbb{F}, \forall p \in \mathbb{P}, \forall j_1, j_2 \in \mathbb{J}^p, s_{pj_1j_2} = 1 \tag{14}$$

$$\sum_{w \in \mathbb{W}_f^S} x_{pj_2fw}^S \le \sum_{j_1 \in \mathbb{J}_p} \sum_{w \in \mathbb{W}_f^S} x_{pj_1fw}^S s_{pj_1j_2}, \quad \forall f \in \mathbb{F}, \forall p \in \mathbb{P}, \forall j_2 \in \mathbb{J}_p, d_{p1q} = 0 \tag{15}$$

$$\sum_{w \in W_f^U} (w(x_{pj_1fw1}^U - x_{pj_2fw1}^U) + (2W_f^U - w) (x_{pj_1fw2}^U - x_{pj_2fw2}^U)) + 2W_f^U \left(\sum_{w \in \mathbb{W}_f^U} \sum_{s \in \mathbb{S}} x_{pj_2fws}^U - 1 \right) \le 0, \quad \forall f \in \mathbb{F}, \forall p \in \mathbb{P}^p, \forall j_1, j_2 \in \mathbb{J}_p, s_{pj_1j_2} = 1 \tag{16}$$

$$\sum_{w \in \mathbb{W}_f^U} \sum_{s \in \mathbb{S}} x_{pj_2fws}^U \le \sum_{j_1 \in \mathbb{J}_p} \sum_{w \in \mathbb{W}_f^U} \sum_{s \in \mathbb{S}} (x_{pj_1fws}^U s_{pj_1j_2}), \quad \forall f \in \mathbb{F}, \forall p \in \mathbb{P}, \forall j_2 \in \mathbb{J}_p, i = 1, d_{pij_2} = 0 \tag{17}$$

$$\sum_{w \in \mathbb{W}_f^S} \left(x_{pj_1fw}^S + x_{pj_2fw}^S \right) \le 1, \quad \forall f \in \mathbb{F}, \forall p \in \mathbb{P}, \forall j_1, j_2 \in \mathbb{J}_p, r_{pj_1j_2} = 1 \tag{18}$$

$$\sum_{w \in \mathbb{W}_f^U} \sum_{s \in \mathbb{S}} \left(x_{pj_1fws}^U + x_{pj_2fws}^U \right) \le 1, \quad \forall f \in \mathbb{F}, \forall p \in \mathbb{P}, \forall j_1, j_2 \in \mathbb{J}_p, r_{pj_1j_2} = 1 \tag{19}$$

$$\beta_{fn} \ge \alpha_{fmpi}, \quad \forall f \in \mathbb{F}, \forall m \in M, \forall p \in \mathbb{P}, \forall i \in \mathbb{I}_p \setminus \{1\}, n \in \mathbb{N} \setminus \{1\} \tag{20}$$

$$\sum_{j \in \mathbb{N}, i \ne j} \gamma_{fij} = \beta_{fi}, \forall f \in \mathbb{F}, \forall i \in \mathbb{N} \tag{21}$$

$$\sum_{i \in \mathbb{N}, i \ne j} \gamma_{fij} = \beta_{fi}, \forall f \in \mathbb{F}, \forall j \in \mathbb{N} \tag{22}$$

$$u_{fi} - u_{fj} + (N-1)\gamma_{fij} \le N - 2, \forall f \in \mathbb{F}, \forall i, j \in \mathbb{N} \setminus \{1\}, i \ne j \tag{23}$$

Constraints 2–23 collectively define assignment, sequencing, resource opening and drone routing decisions. Briefly, decision variable α_{fmpi} records which subcomponents (or subassemblies) are disassembled and the plant to which they are assigned, while Constraint 3 guarantees that each product is assigned to exactly one plant. Constraints 4–5 enforce that each plant opens at most one line type, and Constraints 6–10 ensure tasks are executed only at workstations compatible with the opened line type. Constraint 11 requires each task of each product to be executed exactly once. Constraints 12–13 define the plant operating time via the maximum workstation load T_f. Precedence is enforced by Constraints 14 and 16, with Constraints 15 and 17 handling tasks that have no immediate predecessors. Task-conflict relations are addressed in Constraints 18–19.

Constraint 20 introduces binary variable β_{fn} to indicate whether the drone route visits delivery node n associated with plant f. Intuitively, β_{fn} is activated whenever the disassembly assignment decisions (recorded in the α variables) require a physical transfer or delivery to that plant — i.e., if one or more subassemblies are produced at some origin and must be delivered to plant f, Constraint 20 forces the corresponding β_{fn} to take value 1 so that the delivery node is included in the route.

Constraints 21–22 then enforce flow conservation for the drone tour: each visited delivery node has exactly one incoming and one outgoing arc, which guarantees connectivity of the selected route. Constraint 23 applies the Miller–Tucker–Zemlin (MTZ) sub-tour elimination technique to prevent the route from decomposing into disconnected cycles.

Taken together, these constraints tightly couple disassembly assignment and cross-factory logistics: the model cannot assign a subassembly to a remote plant without also accounting for a drone visit to that plant, and the route selection (subject to travel cost and capacity) will therefore be jointly optimized with disassembly and workstation assignment. This enables

explicit trade-offs between local disassembly decisions and global delivery costs, moving beyond models that treat logistics as a fixed post-processing cost.

III. Proposed Algorithm

A. LoRA Fine-tuning LLM

LoRA is a parameter-efficient fine-tuning method that reduces computational and storage costs while preserving model performance. It achieves this by applying low-rank decomposition to transformation matrices and updating only a small number of trainable parameters. The key idea is to freeze the original weights of the pre-trained model and insert trainable low-rank matrices into specific layers to adapt the model behavior.

In the Transformer architecture, consider a weight matrix $W \in \mathbb{R}^{d\times k}$ in a linear layer. Full fine-tuning requires updating all $d\times k$ parameters. By contrast, LoRA represents the update ΔW as the product of two low-rank matrices:

$$\Delta W = AB,$$

where $A \in \mathbb{R}^{d\times r}$, $B \in \mathbb{R}^{r\times k}$, and the rank $r \ll \min(d,k)$. This decomposition reduces the number of trainable parameters from $d\times k$ to $d\times r + r\times k$. For example, when W is of size 1024×1024 and $r = 8$, LoRA trains only $1024\times 8 + 8\times 1024 = 16{,}384$ parameters, which is approximately 98.44% fewer than the 1,048,576 parameters required by full fine-tuning.

In summary, LoRA provides an efficient and lightweight fine-tuning strategy that significantly reduces resource demands while maintaining strong transferability. This makes it particularly suitable for resource-constrained environments and multi-task adaptation.

In this study, we fine-tune the open-source large model Qwen2.5-7B using the LoRA method implemented in the open-source tool Unsloth to enhance its ability to understand product disassembly task sequences. The hyperparameters for the LoRA fine-tuning were selected based on common practices in the literature [28] and initial validation on a small held-out dataset. We used a LoRA rank $r = 8$, applied to all linear layers in the model. The training was conducted with a batch size of 2, using the AdamW optimizer with a learning rate of 2×10^{-4} and a linear learning rate scheduler. The model was fine-tuned for 3 epochs. This configuration was chosen to balance fine-tuning efficiency and model performance, successfully yielding the high accuracy required for subsequent RL tasks. To support this fine-tuning, we construct a dataset in Alpha format containing the following key components:

1) **Instruction:** A clear description of the specific task for the model, directing it to generate corresponding disassembly tasks.
2) **Input:** Additional background information or contextual data that helps the model better understand task requirements.
3) **Output:** The expected output of the model, namely the disassembly task sequence it should generate under given instruction and input conditions.

The dataset is designed to improve the model's understanding of disassembly tasks, enabling it to generate efficient and reasonable disassembly sequences. These outputs can then be used in subsequent optimization-based decision-making. Table I illustrates an example of the fine-tuning dataset.

B. Action Space Design

The action space is structured hierarchically to reflect the temporal sequence of decision-making in the disassembly process:

1) Assign a product to a disassembly factory **K** and a specific line type **L**.
2) Select a disassembly task sequence from the options generated by the fine-tuned LLM.
3) Assign each task in the chosen sequence to a specific workstation **W**.

In the initial decision step an action index is mapped to $(\mathbf{K}, \mathbf{L})$ (implemented via a modulo operation on the discrete action index). After the factory and line type are selected, the agent chooses one of the feasible disassembly sequences proposed by the LLM; each LLM sequence has a special token `-1` appended to signal a product switch, and the selected sequence is stored in the environment state for subsequent task-level actions. During the task-assignment phase an action corresponds to selecting a workstation **W** for the current task in the stored sequence; when `-1` is encountered the agent switches to the next product or terminates the episode.

The fine-tuned LLM acts as a sequence generator that proposes a small set of candidate sequences (top-K) for each product. At episode start (or when a new product type is first encountered) the LLM generates these top-K candidates; a lightweight constraint checker then validates each candidate against the AND/OR precedence relations and task-conflict constraints, and only validated sequences are retained and indexed in the environment (denote the validated set size by $K_p \le K$). The agent's action selection is hierarchical: (1) choose $(\mathbf{K}, \mathbf{L})$, (2) choose a sequence index $k \in \{1, \ldots, K_p\}$ selecting one validated LLM proposal, and (3) perform per-task workstation assignment following the chosen sequence. To preserve exploration we apply an ε-greedy policy over sequence indices (with occasional uniform sampling of sequence indices) and periodically allow sampling of random feasible sequences outside the LLM proposals. Sequence generation is invoked sparsely (cached after first generation and updated only periodically or when new product families appear) to reduce runtime overhead. This integration reduces invalid/low-quality sequencing choices and compresses the effective action space, while retaining task-level exploration necessary for fine-grained workstation assignment.

C. State Space Design

The state of the environment is represented by a 10-dimensional vector, designed to provide comprehensive contextual information for the agent's policy and value networks. The components of the state vector are defined in Table II.

This state representation encapsulates all essential resource allocation information required for immediate decision-making while also integrating the economic factors necessary

TABLE I An Example of Data Set.

	Content
Instruction	The clutch consists of the following complete disassembly sequence: [[0, 1, 5, 6, 2], [0, 1, 5, 2, 6], [0, 1, 2, 5, 6], [0, 5, 1, 6, 2], [0, 5, 1, 2, 6], [0, 5, 6, 1, 2], [3, 4, 5, 6, 2], [3, 4, 5, 2, 6], [3, 4, 2, 5, 6]].
Input	Select a disassembly sequence for the clutch.
Output	[0, 1, 5, 6, 2]

TABLE II State Vector Representation.

Index	Description
0	Product (P), identifying the product currently being processed.
1	Factory (K), indicating the selected disassembly factory.
2	Workstation (W), denoting the workstation assigned to the task.
3	Disassembly Line (L), specifying the type of disassembly line.
4	Task (J), representing the current disassembly task.
5–9: A set of accumulated economic indicators: • Component revenue (profit from sold parts), • Disassembly cost (based on task time and cost rate), • Factory cycle cost (operational cost proportional to workload), • Workstation startup cost (fixed and variable costs), • Transportation cost (computed via dynamic programming).	

for reward calculation. The holistic design supports informed value estimation and promotes stable and efficient learning convergence.

D. Reward Design

The reward function is designed to reflect the environmental dynamics and is segmented according to different operational phases. Specifically, during the product allocation phase, disassembly sequence selection, product switching, and at the end of an episode, none of these actions directly influence the profit. Therefore, the reward value in these cases is set to 0.

During the workstation assignment phase, however, the reward function is defined as the incremental profit generated by the action. The reward is calculated as follows:

$$R = \begin{cases} \text{profit} - \text{last}_{\text{profit}}, & \text{during the disassembly phase} \\ 0, & \text{otherwise} \end{cases} \tag{24}$$

This design ensures that the agent receives immediate feedback only when its actions lead to tangible economic gains, thereby promoting efficient and profit-oriented behavior during the disassembly operation.

IV. Experimental Study

To comprehensively evaluate the performance of the proposed LLM-based Dueling DQN (LLM-DUEL) algorithm, we conduct a comparative study against three widely used reinforcement learning baselines: the Dueling Deep Q-Network (DUEL), the standard Deep Q-Network (DQN), and Proximal Policy Optimization (PPO). All algorithms are tested on identical experimental cases to ensure fairness in comparative validation.

The computational experiments are executed on a Windows 11 platform equipped with an AMD(R) Core(TM) 4800H CPU (2.90 GHz, 16.00 GB RAM) and an NVIDIA GTX 1650 GPU for RL training and baseline comparisons. For LLM fine-tuning, the Qwen2.5-7B model is trained on Windows 11 with WSL2 and Ubuntu 22.04.5 LTS, running on an AMD Ryzen 5 7500f CPU (3.70 GHz, 32.00 GB RAM) and an NVIDIA GeForce RTX 4070 Super GPU. This dual-environment setup ensures both the RL agents and the LLM fine-tuning process have sufficient computational resources to handle the complexity of the multi-factory optimization problem.

A. Case Generation

To test the robustness of the algorithms across varying complexities, we select four distinct product types for disassembly: washing machine, computer, clutch, and radio. The washing machine and computer represent relatively small-scale disassembly cases, whereas the clutch and radio serve as complex cases due to their structural intricacy. Specifically, the clutch involves more sophisticated disassembly relations between components, while the radio contains a large number of subcomponents, thereby increasing both the task dimensionality and the difficulty of sequencing.

For experimental consistency, the number of disassembly plants is fixed at two and the number of manufacturing plants at three. Disassembly costs, subcomponent values, and transportation costs are generated stochastically, sampled from normal distributions centered around fixed baseline values. This approach simulates realistic market variations while preserving controlled experimental conditions.

Table III summarizes the disassembly characteristics of the selected products, while Table IV details the six test cases constructed by combining different product types in varying quantities. These cases span a wide spectrum of task complexities, from a single washing machine (Case 1) to a highly complex multi-product mix (Case 6).

TABLE III Products for disassembly test

Product	Number of tasks	Number of subcomponents
Washing machine	13	15
Computer	13	13
Clutch	7	17
Radio	30	29

B. Model Validation

Table V reports the maximum objective values found by each algorithm and the CPLEX optimum (where available).

TABLE IV Test cases of multiple products

Case ID	Number of products				Total number of disassembly tasks
	Washing machine	Computer	Clutch	Radio	
1	1	0	0	0	13
2	1	1	0	0	26
3	1	1	1	0	33
4	1	1	1	1	63
5	2	1	1	1	76
6	2	2	1	1	89

The performance gap between LLM-DUEL and the CPLEX optimum is quantified by the relative GAP, calculated as follows:

$$GAP = \frac{CPLEX - LLM}{CPLEX} \times 100\% \tag{25}$$

TABLE V Objective values obtained by each algorithm and LLM-DUEL gap to CPLEX.

Case ID	LLM-DUEL	DUEL	DQN	PPO	CPLEX	GAP (%)
1	664	664	664	664	664	0.00
2	1082	1077	1055	1056	1082	0.00
3	2400	2366	2357	2349	2417	0.70
4	3180	3105	2988	3038	3221	1.27
5	4063	3884	3772	3814	4086	0.56
6	4229	4075	4074	3884	–	–

LLM-DUEL attains the closest objective values to CPLEX across Cases 1–5, matching the optimum in Cases 1 and 2 and exhibiting a maximum gap of 1.27% (Case 4). The mean gap over Cases 1–5 is approximately 0.51%, indicating strong overall agreement with the exact solver on tractable instances. For these cases, CPLEX found proven optima within a time limit of three hours (10,800 seconds). In the complex Case 6, however, CPLEX failed to converge within this time limit; nevertheless, LLM-DUEL produced the best-known solution (4229), outperforming all other RL baselines. These results suggest that LLM-DUEL reliably finds near-optimal solutions while maintaining robust performance on high-complexity instances that challenge exact solvers.

C. Experimental Results and Analysis

To comprehensively evaluate the performance of the proposed LLM-DUEL algorithm, we select three representative reinforcement learning baselines for comparison: the standard Deep Q-Network (DQN) as a foundational value-based method, its enhancement Dueling DQN (DUEL) which improves value estimation by separately modeling state value and action advantages, and Proximal Policy Optimization (PPO) as a state-of-the-art policy-based algorithm known for its training stability. This selection allows us to benchmark against fundamental, advanced, and alternative RL paradigms. Comparing LLM-DUEL directly with DUEL is particularly insightful, as it isolates the performance gain attributable solely to the integration of the fine-tuned LLM. The anticipated underperformance of these baselines in our high-dimensional combinatorial setting highlights the critical contribution of the LLM, which provides explicit domain knowledge and compresses the action space.

As shown in the training curves for six representative cases (Fig. 4), LLM-DUEL consistently delivers the best performance across all instances: it attains the fastest convergence, the highest final objective values, and the lowest run-to-run variability. DUEL is the second-best method, providing a reasonable trade-off between convergence speed and final objective, but still trailing LLM-DUEL. DQN achieves acceptable results on some instances but generally exhibits slower convergence and larger variance. PPO shows the weakest performance here, suffering from unstable convergence and limited improvement in objective value.

To further compare the training cost, we examine the sample efficiency in terms of iteration steps required for convergence, as observed in Fig. 4. LLM-DUEL achieved convergence at approximately 10,420 steps, which was faster than DUEL (12,830 steps) and DQN (11,330 steps), demonstrating that the LLM's guidance leads to more efficient policy search. It is noteworthy that while PPO converged at a comparable number of steps (10,170), it settled at a significantly lower performance level, indicating ineffective exploration despite its sample efficiency. This analysis underscores that LLM-DUEL achieves a superior balance—attaining higher solution quality with fewer iterations than its value-based counterparts.

The superior behavior of LLM-DUEL stems primarily from the LLM-generated disassembly sequences, which (i) eliminate many invalid or low-quality actions, and (ii) substantially compress the agent's effective action space. This reduction in spurious actions improves exploration efficiency and accelerates policy learning, enabling LLM-DUEL to approach CPLEX performance on tractable instances while remaining far more efficient computationally for complex cases. Overall, the results demonstrate that coupling structured, domain-aware LLM guidance with RL yields a robust and efficient solution approach for complex remanufacturing scheduling problems.

D. Effectiveness of LLM Fine-Tuning

To validate the effectiveness of the proposed LoRA fine-tuning strategy, we conducted a comprehensive evaluation comparing the sequence generation accuracy of the base Qwen2.5-7B model against its fine-tuned counterpart. The evaluation was performed on four distinct product types with varying complexity levels: clutch, washing machine, radio, and computer.

Each model was tasked with generating 20 disassembly sequences per product to produce valid sequences that conform to the predefined AND/OR graph constraints. The evaluation metric was defined as the proportion of valid sequences generated out of the 20 attempts, averaged across all test cases.

As presented in Table VI, the fine-tuned Qwen2.5-7B model achieved perfect accuracy (100%) across all product types, demonstrating its capability to generate valid disassembly sequences consistently. In contrast, the base model exhibited an overall accuracy of 73.75%, with performance varying significantly based on product complexity.

The base model's performance degradation was particularly notable for the radio, which has the most complex disassembly structure with 25 valid sequences. This product recorded the

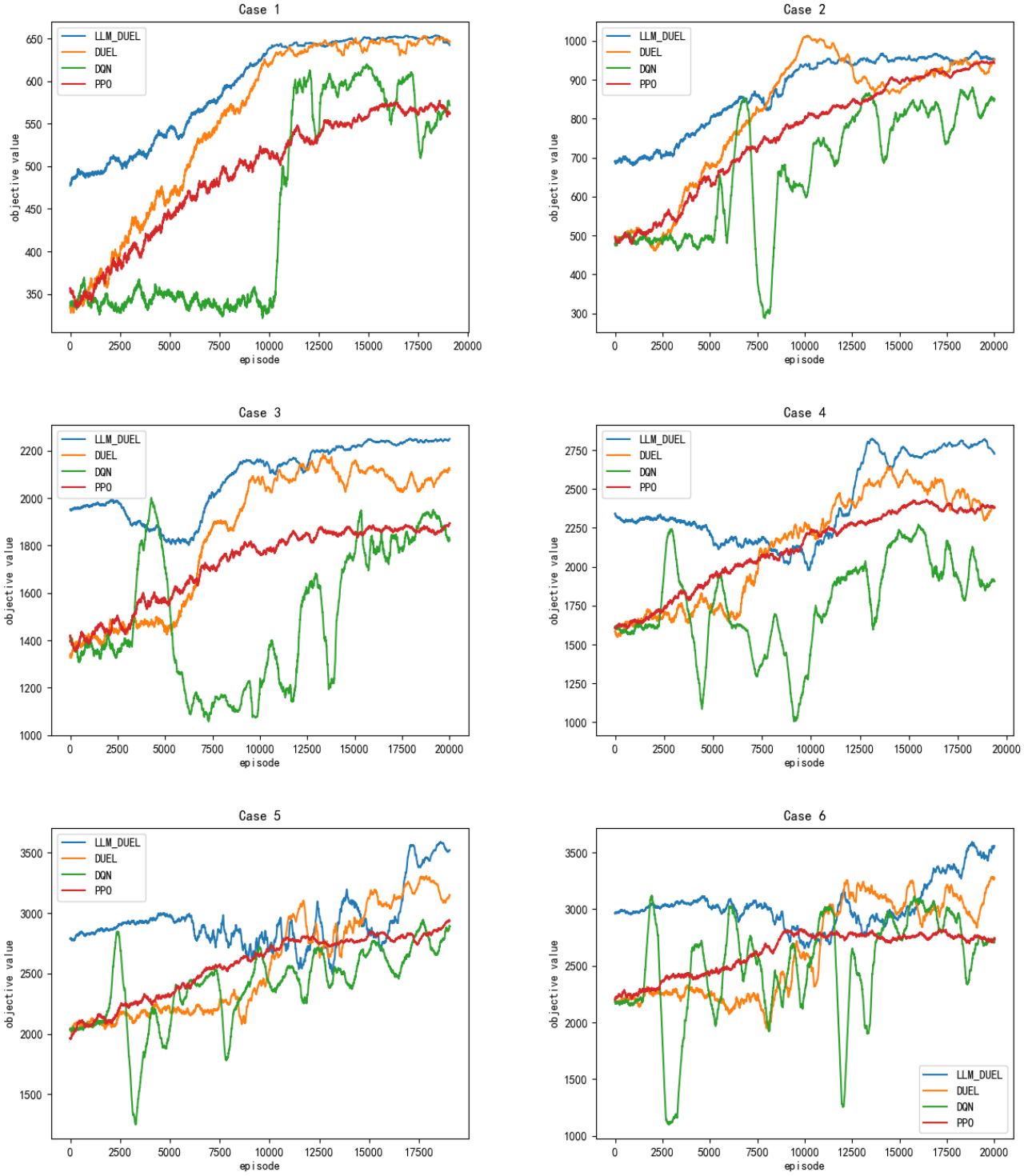


Fig. 4. Comparison of training maps for various algorithms.

TABLE VI Accuracy comparison between base and fine-tuned Qwen2.5-7B models

Product Type	Number of Valid Sequences	Base Model Accuracy	Fine-tuned Model Accuracy
Washing Machine	5	70.00%	100%
Computer	6	80.00%	100%
Clutch	9	85.00%	100%
Radio	25	60.00%	100%
Overall	**-**	**73.75%**	**100%**

lowest accuracy (60.00%), indicating that without task-specific fine-tuning, the model struggles with complex combinatorial optimization tasks. The clutch and computer, with relatively simpler disassembly constraints, achieved higher accuracy rates of 85.00% and 80.00%, respectively.

These results highlight two key insights: First, the base Qwen2.5-7B model possesses inherent sequence generation capabilities but lacks the specialized knowledge required for consistent performance in disassembly tasks. Second, the LoRA fine-tuning approach effectively injects domain-specific knowledge, enabling the model to perfectly comprehend and generate valid disassembly sequences regardless of product complexity.

The 26.25% improvement in overall accuracy demonstrates the critical importance of domain adaptation for LLMs in manufacturing optimization tasks. This performance enhancement directly contributes to the LLM-DUEL algorithm's effectiveness by ensuring that the reinforcement learning agent always receives feasible disassembly sequences, thereby accelerating policy search and improving solution quality.

Analysis of the base model's errors revealed two primary failure modes: (1) Format errors where the model generated malformed sequence representations (e.g., "[[1, 2, 5, 10, 11]" with missing brackets), and (2) Logical errors where the model produced sequences that violated precedence or conflict constraints. The fine-tuned model eliminated both error types, confirming that the LoRA adaptation successfully aligned the model's output patterns with the structural requirements of disassembly sequencing.

V. Conclusion

This paper proposes LLM-Duel for the Multi-Factory Remanufacturing Optimization Problem with hybrid disassembly lines and cross-factory drone delivery. By combining a LoRA-fine-tuned LLM that generates feasible disassembly sequences with a Dueling DQN policy that selects sequences and assigns tasks, LLM-Duel reduces invalid actions, compresses the RL action space, and achieves faster convergence, better training stability, and higher objective values compared with standard RL baselines and traditional methods.

While LLM-Duel demonstrates promising performance on the synthetic, structurally realistic cases studied here, several practical constraints should be noted before industrial deployment. First, the current implementation relies on offline LoRA fine-tuning of a single base model using curated disassembly examples; applying the approach to new product families will typically require additional domain-specific data or more advanced prompt/transfer strategies. Second, our experiments make simplifying assumptions (deterministic parameters and a single drone per factory) that abstract away stochastic arrivals, variable processing times, heterogeneous drone fleets, regulatory/airspace constraints, and other operational uncertainties common in practice. Third, computational and latency considerations matter: despite LoRA's parameter efficiency, large LLMs still incur non-trivial inference and (re)training costs that may limit on-device or real-time deployment without further model compression or distillation. Finally, safe and effective rollout will require human-in-the-loop validation and operator acceptance mechanisms to handle edge cases and ensure safety. Addressing these issues motivates extensions such as continual or online fine-tuning, explicit stochastic modeling, support for heterogeneous multi-drone coordination, and human-centric verification layers.

Future research will focus on extending the approach along several directions: (i) exploring additional parameter-efficient fine-tuning strategies to improve LLM sequence quality and generalization; (ii) incorporating multiple and heterogeneous drones, stochastic dynamics, and more flexible remanufacturing environments; (iii) integrating human-centric considerations such as fatigue, ergonomics, and safety-aware allocation; and (iv) validating the model on diverse factory layouts and considering advanced optimization methods and formal-model approaches to further enhance solution quality [32, 33, 34, 35, 36].

References

[1] S. Qin, S. Dai, J. Wang, S. Liu, X. Guo, L. Qi, B. Hu, and Y. Ji, "Improved carnivorous plant algorithm for human–robot collaborative u-shaped disassembly line balancing with mobile workers," *IEEE Transactions on Computational Social Systems*, pp. 1–13, 2025, early Access.

[2] Z. Bi, X. Guo, J. Wang, S. Qin, and G. Liu, "Truck-drone delivery optimization based on multi-agent reinforcement learning," *Drones*, vol. 8, no. 1, p. 27, 2024.

[3] S. Dai, Z. Zhang, W. Wang, C. Li, J. Parron, and E. Herrera, "Human-robot collaborative disassembly profit maximization via improved grey wolf optimizer," *International Journal of Artificial Intelligence and Green Manufacturing*, vol. 1, no. 2, pp. 12–22, 2025.

[4] Y. Feng, S. Dai, Z. Zhang, X. Guo, S. Qin, Q. Kang, and Y. Liu, "A disassembly and assembly line balancing problem via an improved double q-learning," in *2025 37th Chinese Control and Decision Conference (CCDC)*, IEEE. Xiamen, China: IEEE, 2025, pp. 890–895.

[5] J. Wang, M. Zhou, X. Guo, and L. Qi, "Multiperiod asset allocation considering dynamic loss aversion behavior of investors," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 1, pp. 73–81, 2018.

[6] X. Guo, F. Guo, L. Qi, J. Wang, S. Liu, S. Qin, and W. Wang, "Modeling and optimization of multiproduct human–robot collaborative hybrid disassembly line balancing with resource sharing," *IEEE Transactions on Computational Social Systems*, pp. 1–16, Mar 2025, early Access.

[7] S. Dai, Y. Feng, Z. Zhang, X. Guo, S. Qin, Q. Kang, and Y. Liu, "Solving disassembly and assembly line balancing problem with robot direction switching," in *2025 37th Chinese Control and Decision Conference (CCDC)*, IEEE. Xiamen, China: IEEE, may 2025, pp. 896–901.

[8] L. Qi, M. Li, X. Guo, and W. Luan, "Multi-objective optimization for robotaxi dispatch with safety-carpooling mode in pandemic era," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 1, pp. 878–891, 2024.

[9] Y. Jing, W. Li, X. Wang, and L. Deng, "Production planning with remanufacturing and back-ordering in a cooperative multi-factory environment," *International Journal of Computer Integrated Manufacturing*, vol. 29, no. 6, pp. 692–708, 2016.

[10] F. Marandi and S. Fatemi Ghomi, "Integrated multi-factory production and distribution scheduling applying vehicle routing approach," *International Journal of Production Research*, vol. 57, no. 3, pp. 722–748, 2019.

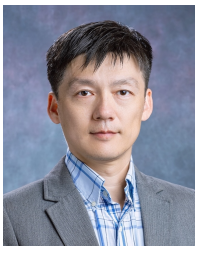
[11] L. Qi, Q. Zeng, S. Liu, J. Wang, S. Qin, and X. Guo, "Twin delayed deep deterministic policy gradient algorithm for a heterogeneous multifactory remanufacturing optimization problem," *IEEE Transactions on Computational Social Systems*, pp. 1–12, 2025, early Access.
[12] X. Guo, L. Chen, L. Qi, J. Wang, S. Qin, M. Chatterjee, and Q. Kang, "Multifactory disassembly process optimization considering worker posture," *IEEE Transactions on Computational Social Systems*, pp. 1–13, March 2025, early Access.
[13] L. Zhou, H. Zhu, and B. Akbari, "Multi-objective optimization of multifactory remanufacturing process considering worker fatigue," *International Journal of Artificial Intelligence and Green Manufacturing*, vol. 1, no. 2, pp. 36–50, 2025.
[14] J. Lohmer and R. Lasch, "Production planning and scheduling in multifactory production networks: a systematic literature review," *International Journal of Production Research*, vol. 59, no. 7, pp. 2028–2054, 2021.
[15] N. Karimi and H. Davoudpour, "Integrated production and delivery scheduling for multi-factory supply chain with stage-dependent inventory holding cost," *Computational and Applied Mathematics*, vol. 36, no. 4, pp. 1529–1544, 2017.
[16] C. Moon, J. Kim, and S. Hur, "Integrated process planning and scheduling with minimizing total tardiness in multi-plants supply chain," *Computers & Industrial Engineering*, vol. 43, no. 1-2, pp. 331–349, 2002.
[17] F. T. Chan, V. Kumar, and N. Mishra, "Resolving multi plant supply chain problem: A novel swarm intelligence based approach," in *2008 4th IEEE International Conference on Management of Innovation and Technology*, IEEE. Bangkok, Thailand: IEEE, Sep. 2008, pp. 1066–1071.
[18] B. Madani and M. Ndiaye, "Hybrid truck-drone delivery systems: A systematic literature review," *IEEE Access*, vol. 10, pp. 92 854–92 878, 2022.
[19] O. Dukkanci, J. F. Campbell, and B. Y. Kara, "Facility location decisions for drone delivery: A literature review," *European Journal of Operational Research*, vol. 316, no. 2, pp. 397–418, 2024.
[20] J. Wang, G. Xi, X. Guo, S. Liu, S. Qin, and H. Han, "Reinforcement learning for hybrid disassembly line balancing problems," *Neurocomputing*, vol. 569, p. 127145, 2024.
[21] H. Qin, T. Li, Y. Teng, and K. Wang, "Integrated production and distribution scheduling in distributed hybrid flow shops," *Memetic Computing*, vol. 13, no. 2, pp. 185–202, 2021.
[22] J. Cai, D. Lei, J. Wang, and L. Wang, "A novel shuffled frog-leaping algorithm with reinforcement learning for distributed assembly hybrid flow shop scheduling," *International Journal of Production Research*, vol. 61, no. 4, pp. 1233–1251, 2023.
[23] R. Liu, R. Piplani, and C. Toro, "Deep reinforcement learning for dynamic scheduling of a flexible job shop," *International Journal of Production Research*, vol. 60, no. 13, pp. 4049–4069, 2022.
[24] J.-J. Wang and L. Wang, "A cooperative memetic algorithm with learning-based agent for energy-aware distributed hybrid flow-shop scheduling," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 3, pp. 461–475, 2021.
[25] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian, "A comprehensive overview of large language models," *ACM Transactions on Intelligent Systems and Technology*, vol. 16, no. 5, pp. 1–72, 2025.
[26] Y. Cao, H. Zhao, Y. Cheng, T. Shu, Y. Chen, G. Liu, G. Liang, J. Zhao, J. Yan, and Y. Li, "Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 6, pp. 9737–9757, jun 2024.
[27] N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C.-M. Chan, W. Chen *et al.*, "Parameter-efficient fine-tuning of large-scale pre-trained language models," *Nature machine intelligence*, vol. 5, no. 3, pp. 220–235, 2023.
[28] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen *et al.*, "Lora: Low-rank adaptation of large language models." *ICLR*, vol. 1, no. 2, p. 3, 2022.
[29] X. Guo, S. Liu, M. Zhou, and G. Tian, "Disassembly sequence optimization for large-scale products with multiresource constraints using scatter search and petri nets," *IEEE Transactions on Cybernetics*, vol. 46, no. 11, pp. 2435–2446, 2016.
[30] Z. Zhao, S. Liu, M. Zhou, D. You, and X. Guo, "Heuristic scheduling of batch production processes based on petri nets and iterated greedy algorithms," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 1, pp. 251–261, 2022.
[31] S.-e. Zhao, Y.-l. Li, R. Fu, and W. Yuan, "Fuzzy reasoning petri nets and its application to disassembly sequence decision-making for the end-of-life product recycling and remanufacturing," *International Journal of Computer Integrated Manufacturing*, vol. 27, no. 5, pp. 415–421, 2014.
[32] L. Qi, M. Li, X. Guo, and W. Luan, "Multi-objective optimization for robotaxi dispatch with safety-carpooling mode in pandemic era," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 1, pp. 878–891, Jan. 2025.
[33] S. Qin, F. Guo, J. Wang, L. Qi, J. Wang, X. Guo, and Z. Zhao, "Expanded discrete migratory bird optimizer for circular disassembly line balancing with tool deterioration and replacement," *International Journal of Artificial Intelligence and Green Manufacturing*, vol. 1, no. 1, pp. 14–24, 2025.
[34] J. Gu, Z. Guo, J. Wang, L. Qi, S. Qin, and S. Zhang, "Optimization of multi-factory remanufacturing processes with shared transportation resources using the alns algorithm," *International Journal of Artificial Intelligence and Green Manufacturing*, vol. 1, no. 1, pp. 1–13, 2025.
[35] L. Qi, Y. Su, M. Zhou, and A. Abusorrah, "A state-equation-based backward approach to a legal firing sequence existence problem in petri nets," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 8, pp. 4968–4979, 2023.
[36] Y. Su, M. Zhou, L. Qi, and R. Wiśniewski, "A reachability-decidable petri net modeling method for discrete event systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 55, no. 1, pp. 453–464, Jan. 2025.

Shujin Qin received the Ph.D degree in system engineering from the Northeastern University, Shenyang, China, in 2019.

He joined Shangqiu Normal University, Shangqiu, China, in 2019, and is currently a Lecturer of artifcial intelligence. His research interests include large-scaled integer programming vehicle routing problem,reinforcement learning and intelligent optimization algorithm.

Shaokang Dai received his degree in software engineering from Southeast University Chengxian College, China, in 2022.

He is currently a graduate student at the School of Artificial Intelligence and Software, Liaoning Shihua University. His current research interests include remanufacturing, intelligent optimization algorithm, and graph neural networks.



Bin Hu received a Ph.D. degree in Electrical and Computer Engineering at Rutgers University.

Currently, he is an Assistant Professor at the Computer Science and Technology Department at Kean University. His research interests include mobile computing and sensing, cybersecurity and privacy, and efficient deep learning.