Mixed-Layout Multi-Type Factory Remanufacturing System Optimization via LLM-TD3

Xiwang Guo, Yujie Feng and Mengchu Zhou

***Abstract*—This work presents a mixed-layout, multi-type factory remanufacturing system optimization problem that considers both linear and U-shaped disassembly lines, with the goal of maximizing profit, and formulates its corresponding mathematical model. Its solution has four stages: product allocation, disassembly line selection, task allocation, and component transportation. Based on the characteristics of each stage, Large Language Model (LLM) is responsible for product allocation and disassembly line selection, while Twin Delayed Deep Deterministic Policy Gradient optimizes task allocation and component transportation according to the LLM's results. By providing the estimated profits of products under different factory and disassembly line configurations and designing tailored action-state space, the proposed method interacts with the environment to solve the problem. By using various experimental cases, we compare it with CPLEX, Deep Deterministic Policy Gradient, Soft Actor-Critic, and Advantage Actor-Critic to verify its feasibility and effectiveness, demonstrating its potential as a novel solution method.**

***Key Words*—Large language models, twin delayed deep deterministic policy gradient, remanufacturing optimization, mixed integer programming, Petri nets.**

I. Introduction

AGAINST the backdrop of accelerating global industrialization, resources are being increasingly depleted while the pressure on the ecological environment continues to escalate. Under these circumstances, the traditional linear economic model of 'extraction-production-disposal' has become inadequate to meet current industrial requirements. With the deep integration of a global industrial chain and continuous expansion of economic scale, the shortage of raw materials and surge in waste have become the main bottlenecks restricting sustainable development, forcing academic and industrial communities to examine the transformation path of production models.

Under this circumstance, the concepts of circular economy and green manufacturing have gradually taken shape. The former, with the core principle of maximizing resource utilization and minimizing carbon emissions, promotes closed-loop material flow. The latter, on the other hand, integrates the concept of environmental friendliness throughout the entire life cycle of a product, from design, production, and use to recycling. Together, they form a solid foundation for sustainable development.

In the context of this macrotrend, as a key practice linking circular economy and green manufacturing, remanufacturing has become a focal area of global industrial transformation. By systematically disassembling, inspecting, repairing, and upgrading used products, remanufacturing achieves the dual goals of resource regeneration and value reshaping, bringing the technical performance and reliability of these products to or even beyond the standards of new ones. Compared to traditional manufacturing, remanufacturing offers significant synergistic benefits in alleviating resource constraints, reducing production cost, and protecting the ecological environment. As a result, it provides a viable solution for the global manufacturing industry to tackle the dual challenges of development and environmental protection.

In recent years, with the rapid development of artificial intelligence and machine learning technologies, the combination of Large Language Models (LLMs) and deep reinforcement learning algorithms has provided new ideas and methods for solving complex optimization problems [1]. LLMs can offer efficient modeling and decision-making support for optimization problems through their natural language processing and simulated logical reasoning capabilities [2, 3]. Meanwhile, the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm, through its reinforcement learning mechanism [4, 5, 6], is able to learn optimal strategies in dynamic environments, thereby achieving optimization of complex systems [7, 8, 9]. The integration of these two technologies is expected to offer a new solution for the optimization problem of mixed-line layout and multi-type factory remanufacturing systems.

The LLM-based TD3 strategy has opened up new ways to handle the remanufacturing adjustment issues of various factories. With its excellent data processing and pattern recognition capabilities [10, 11, 12], LLM can simulate and evaluate complicated remanufacturing production processes. As a cutting-edge reinforcement learning technique, TD3 is capable of properly addressing the challenges of continuous action spaces and high-dimensional state space, and it features good stability and fast convergence [13, 14, 15]. The combination of LLM and TD3 can fully utilize the strengths of both, thereby improving the efficiency and accuracy in solving our concerning optimization problem.

Manuscript received October 5, 2025; revised October 12 and October 19, 2025; accepted December 29, 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 Liaoning Revitalization Talents Program under Grant XLYC1907166, in part by the Natural Science Foundation of Shandong Province under Grant ZR2024BF140, and in part by Archival Science and Technology Project of Liaoning Province under Grant 2021-B-004.

X. Guo and Y. Feng are with the College of Artificial Intelligence and Software, Liaoning Petrochemical University, Fushun 113001, China (e-mail: x.w.guo@163.com, xf1553@163.com).

M. Zhou is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102-1982 USA (e-mail: zhou@njit.edu).

Corresponding Author: Xiwang Guo

The contributions of this work are as follows.

1) This work combines LLM with TD3 to solve a newly proposed optimization problem of remanufacturing in mixed-line layout and multi-type factories.

2) It establishes a mathematical model aimed at the maximizing revenue of a remanufacturing system. The correctness of the model is verified by using CPLEX as a commercially available solver.

3) The designed LLM-TD3 algorithm is compared with TD3, Deep Deterministic Policy Gradient (DDPG), Advantage Actor-Critic (A2C), and Soft Actor-Critic (SAC). The results show that it has more advantages than its peers in solving the proposed optimization problem.

The remainder of this paper is organized as follows: Section II describes the problem. Section III elaborates on the proposed algorithm. Section IV presents the experimental results and analysis. Finally, Section V concludes the work and discusses directions for future work.

II. Problem Description

A. Problem Statement

This work studies a Mixed-layout multi-type factory Remanufacturing system Optimization Problem (MROP), aiming to maximize the profit while considering the technical capabilities of different disassembly factories and the choice of disassembly line layouts [16]. Each disassembly factory can choose either a linear or U-shaped disassembly line to dismantle products, and the choice of layout affects both disassembly efficiency and cost [17, 18, 19]. In addition, the technical capabilities of each factory limit the types of products it can process. Therefore, the optimization problem needs to consider both technical capability matching and disassembly line layout selection to achieve the maximum profit. Fig. 1 shows the overall framework of MROP.

A MROP extends the conventional multi-type factory remanufacturing optimization problem in [20, 21, 22] by considering different disassembly lines operated disassembly factories. Its includes the following four key issues:

1) Product allocation

Optimization MROP encompasses a network of multiple disassembly factories, each one being uniquely equipped with a distinct set of disassembly technologies. It is these specific technologies that dictate the range of products a particular factory is capable of handling. The disassembly of certain highly complex products may necessitate the use of advanced and specialized technologies, which may not be available across all factories. Consequently, the allocation process is of paramount importance, as it must meticulously ensure that the technical capabilities of a factory are in precise alignment with the technical prerequisites of the tasks assigned to it. In the context of this study, disassembly factories are categorized based on distinct types of disassembly technologies. Products are then assigned to factories that possess the requisite technical capabilities to meet their specific disassembly needs.

In order to describe the problem more clearly, we have established two matrices.

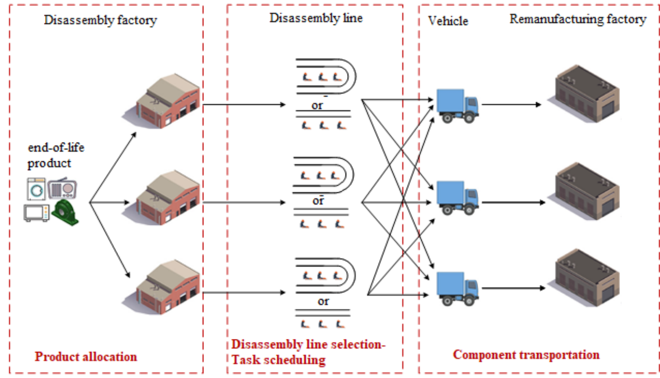


Fig. 1. Framework of the MROP.

1) Product-technology association matrix $\Theta = [\theta_{pn}]$ describes the relationship between product p and disassembly technology n.

$$\theta_{pn} = \begin{cases} 1, & \text{disassembling product } p \text{ requires} \\ & \text{technology } n. \\ 0, & \text{otherwise.} \end{cases}$$

2) Factory-technology association matrix $Y = [\delta_{kn}]$ describes the relationship between disassembly factory k and disassembly technology n.

$$\delta_{kn} = \begin{cases} 1, & \text{disassembly factory } k \text{ has technology } n. \\ 0, & \text{otherwise.} \end{cases}$$

2) Selection of disassembly lines

Each disassembly factory has the option to select either a linear or U-shaped disassembly line for processing products. linear layouts are particularly well-suited for handling simple and continuous tasks, whereas U-shaped layouts are more advantageous for dealing with complex disassembly scenarios that involve multiple tasks. The choice of layout has a significant impact on disassembly efficiency, cost, and workstation utilization.

3) Task scheduling

The assignment of product disassembly tasks to workstations must be carried out in an efficient manner. This process takes into account a variety of factors, including the duration of a task, its priority level, and any potential conflicts, such as the differing amounts of time required to dismantle each specific task. By considering these factors, it ensures that each workstation is allocated sufficient time to successfully complete the tasks assigned to it [23].

4) Component transportation

The transportation of disassembled components from disassembly factories to remanufacturing factories should be well planned to maximize profitability. This process must carefully consider the recovery value of components at various remanufacturing sites and transportation costs incurred between disassembly and remanufacturing factories [24, 25, 26].

The following is the correlation matrices are required to describe our problem:

1) Incidence matrix $A = [\alpha_{pij}]$ describes the relationship between components and disassembly tasks, where i means the i-th components, j disassembly the j-th task and p the p-th product.

$$\alpha_{pij} = \begin{cases} 1, & \text{if component } i \text{ of product } p \text{ is obtained} \\ & \text{by task } j. \\ -1, & \text{if component } i \text{ of product } p \text{ is} \\ & \text{disassembled by task } j. \\ 0, & \text{otherwise.} \end{cases}$$

2) Precedence matrix $B = [\beta_{pjj'}]$ describes the relationship between two tasks, where j and j' represent disassembly tasks.

$$\beta_{pjj'} = \begin{cases} 1, & \text{if task } j \text{ must be executed before task } j' \\ & \text{to disassemble product } p. \\ 0, & \text{otherwise.} \end{cases}$$

3) Conflict matrix $\mathbf{\Gamma} = [\gamma_{pjj'}]$ describes the conflict between two tasks, j and j'.

$$\gamma_{pjj'} = \begin{cases} 1, & \text{if task } j \text{ and task } j' \text{ to disassemble product } p \text{ are} \\ & \text{in conflict.} \\ 0, & \text{otherwise.} \end{cases}$$

B. Petri Nets

A Petri net is a mathematical tool for modeling and analyzing discrete event systems, widely used in fields such as production systems, computer science, and communication networks [27, 28, 29, 30, 31]. It describes system state changes and event triggering in a graphical manner, capable of intuitively representing system characteristics such as concurrency, synchronization, and resource sharing [32, 33]. It consists of places, transitions, arcs, and tokens. Places represent the states or resources of the system, typically depicted as circles. Transitions represent events or actions in the system, typically shown as vertical lines or rectangles. Arcs represent the relationships between places and transitions, indicated by arrows, with the direction of the arrows showing the flow direction of tokens. Tokens represent the availability of resources or the occurrence of events, and the presence and flow of tokens indicate state changes in the system.

Petri nets are applied to disassembly systems for modeling and analyzing disassembly processes [34, 35]. A DPN consists of tokens, places, transitions, and arcs. Tokens represent the availability of products, components, or subcomponents. Places represent the states of product components. Transitions correspond to disassembly operations, and arcs correspond to the flow between component tasks, which can represent priority relationships between actions. The DPN of a product is established based on the geometric constraint relationships and/or logical relationships among all subcomponents/parts that constitute the product, as well as disassembly states and disassembly resource information. This work uses DPN to describe the relationships between tasks and components in a product. Fig. 2 shows a simple DPN. The product disassembly starts from M_0, and when S_1 is present, T_1 can be executed. After executing T_1, S_1 is consumed and S_2 and S_3 are produced, meaning that by performing disassembly task T_1, component S_1 is disassembled to obtain components S_2 and S_3.

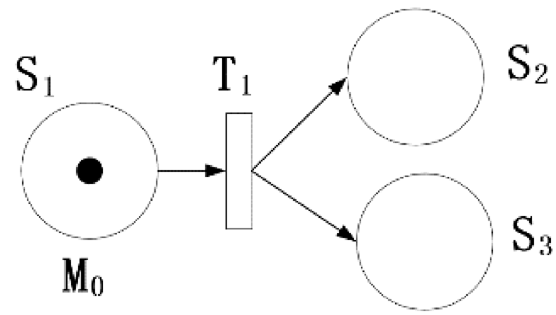


Fig. 2. A simple DPN.

C. Mathematical Model

1) Notations

k	Number of disassembly factories.
r	Number of remanufacturing factories.
p	Number of products.
I_p	Maximum number of components for product p.
J_p	Maximum number of tasks for product p.
w	Number of workstations in a disassembly factory.
n	Index of disassembly techniques.
s	Index on a side of U-shaped line.
$\mathbb{K}$	Set of disassembly factories, $\mathbb{K}$={1,2,...,k}.
$\mathbb{R}$	Set of remanufacturing factories, $\mathbb{R}$={1,2,...,r}.
$\mathbb{P}$	Set of products, $\mathbb{P}$={1,2,...,p}.
$\mathbb{I}_p$	Set of components in product p, $\mathbb{I}_p = \{1, 2, ..., I_p\}$.
$\mathbb{J}_p$	Set of tasks in product p, $\mathbb{J}_p$={1,2,...,J_p}.
$\mathbb{W}_k^U$	Set of workstations on U-shaped line of disassembly factory k, $\mathbb{W}_k^U = \{1, 2, ..., w_k^U\}$.
$\mathbb{W}_k^L$	Set of workstations on linear line of disassembly factory k, $\mathbb{W}_k^L = \{1, 2, ..., w_K^L\}$.
$\mathbb{S}$	Set of sides of U-shaped line, $\mathbb{S} = \{1, 2\}$.
$\mathbb{N}$	Set of disassembly techniques, $\mathbb{N}$={1,2,...,n}.
v_{rpi}	The price at which remanufacturing factory r purchases disassembled component i of product p.
c_{krpi}	The cost of transporting the i-th component of product p from disassembly factory k to remanufacturing factory r.
t_{kpj}	Time required by workers in disassembly factory k to complete task j for product p.
c_{kpj}^D	Unit time cost incurred by workers in disassembly factory k for performing task j of product p.
c_k^O	Unit time cost associated with the operation of disassembly factory k.
c_{kw}^U	Fixed cost to activate workstation w on U-shaped line of disassembly factory k.
c_{kw}^L	Fixed cost to activate workstation w on linear line of disassembly factory k.
T_k^t	The cycle time of disassembly factory k.

2) Decision variables

$$z_{pk} = \begin{cases} 1, & \text{if product } p \text{ is disassembled at factory } k. \\ 0, & \text{Otherwise.} \end{cases}$$

$$x^{L}_{pjkw} = \begin{cases} 1, & \text{if disassembly task } j \text{ of product } p \text{ is assigned to linear workstation } w^{L}_{k} \text{ in disassembly factory } k. \\ 0, & \text{otherwise.} \end{cases}$$

$$x^{U}_{pjkws} = \begin{cases} 1, & \text{if disassembly task } j \text{ of product } p \text{ is performed on side } s \text{ of workstation } w^{U}_{k} \text{ in disassembly factory } k. \\ 0, & \text{otherwise.} \end{cases}$$

$$y^{L}_{k} = \begin{cases} 1, & \text{if the linear disassembly line of disassembly factory } k \text{ is opened.} \\ 0, & \text{otherwise.} \end{cases}$$

$$y^{U}_{k} = \begin{cases} 1, & \text{if the U-shaped disassembly line of disassembly factory } k \text{ is opened.} \\ 0, & \text{otherwise.} \end{cases}$$

$$u^{L}_{kw} = \begin{cases} 1, & \text{if workstation } w \text{ on linear line of disassembly factory } k \text{ is opened.} \\ 0, & \text{otherwise.} \end{cases}$$

$$u^{U}_{kw} = \begin{cases} 1, & \text{if workstation } w \text{ on U-shaped line of disassembly factory } k \text{ is opened.} \\ 0, & \text{otherwise.} \end{cases}$$

$$\eta_{krpi} = \begin{cases} 1, & \text{if the } i\text{-th component of product } p \text{ is shipped from disassembly factory } k \text{ to remanufacturing factory } r. \\ 0, & \text{otherwise.} \end{cases}$$

3) Assumptions

To focus solely on the core aspects of this work, makes problem solving the following assumptions.

a. The parameters of the disassembled products in different disassembly factories are known, encompassing the time and cost of disassembly tasks for the disassembled products, the profits obtained, and the Petri net of products.
b. Each disassembly factory operates independently.
c. The operating cost of each disassembly factory is known, including the factory startup cost.
d. The parameters related to the disassembly lines of each factory are also known, such as the costs of workstations for linear or U-shaped disassembly lines in different disassembly factories and the costs of starting disassembly lines.
e. The distances and transportation cost between disassembly factories and remanufacturing factories are known.
f. The recycled parts possess certain remanufacturing and reuse value.
g. The disassembly technology required for disassembling products and the disassembly technology owned by disassembly factories are known.

4) Objective is to maximize disassembly profit

$$\begin{aligned} f = & \sum_{k\in\mathbb{K}}\sum_{r\in\mathbb{R}}\sum_{p\in\mathbb{P}}\sum_{i\in\mathbb{I}_p}(v_{rpi}-c_{krpi})\eta_{krpi} \\ & -\sum_{k\in\mathbb{K}}\sum_{p\in\mathbb{P}}\sum_{j\in\mathbb{I}_p}\sum_{w\in\mathbb{W}^{L}_{k}} c^{D}_{kpj}t_{kpj}x^{L}_{pjkw} \\ & -\sum_{k\in\mathbb{K}}\sum_{p\in\mathbb{P}}\sum_{j\in\mathbb{J}_p}\sum_{w\in\mathbb{W}^{t}_{k}}\sum_{s\in\mathbb{S}} c^{D}_{kpj}t_{kpj}x^{U}_{pjkws} - \sum_{k\in\mathbb{K}} c^{O}_{k}T^{t}_{k} \\ & -\sum_{k\in\mathbb{K}}\sum_{w\in\mathbb{W}^{L}_{k}} c^{L}_{kw}u^{L}_{kw} - \sum_{k\in\mathbb{K}}\sum_{w\in\mathbb{W}^{U}_{k}} c^{U}_{kw}u^{U}_{kw} \end{aligned} \tag{1}$$

The first term in (1) captures the profit derived from the difference between the recovery value of disassembled components and their transportation cost. The second and third terms account for the cost associated with disassembling products on linear and U-shaped disassembly lines, respectively. The fourth term represents the fixed cost of operating disassembly factories. Lastly, the fifth and sixth terms correspond to the fixed cost of activating workstations on the linear and U-shaped disassembly lines, respectively.

5) Constraints

$$\begin{aligned} & \beta_{pjj'}(\sum_{w\in\mathbb{W}^{U}_{k}}\Big(w(x^{U}_{pjkw_1}-x^{U}_{pj'kw_1})+(2W^{U}_{k}-w)\Big) \\ & (x^{U}_{pjkw_2}-x^{U}_{pj'kw_2})+2W^{U}_{k}(\sum_{w\in\mathbb{W}^{U}_{k}}\sum_{s\in\mathbb{S}}x^{U}_{pj'kws}-1))\leq 0, \\ & \forall p\in\mathbb{P}, j,j'\in\mathbb{J}_p, k\in\mathbb{K} \end{aligned} \tag{2}$$

$$u^{U}_{kw}\leq y^{U}_{k}, \forall w\in\mathbb{W}^{U}_{k}, k\in\mathbb{K} \tag{3}$$

$$x^{U}_{pjkws}\leq u^{U}_{kw}, \forall p\in\mathbb{P}, j\in\mathbb{J}_p, k\in\mathbb{K}, w\in\mathbb{W}^{U}_{k}, s\in\mathbb{S} \tag{4}$$

$$\begin{aligned} & \gamma_{pjj'}(\sum_{w\in\mathbb{W}^{U}_{k}}\sum_{s\in\mathbb{S}}(x^{U}_{pjkws}+x^{U}_{pj'kws}))\leq 1, \\ & \forall p\in\mathbb{P}, j,j'\in\mathbb{J}_p, k\in\mathbb{K} \end{aligned} \tag{5}$$

$$\sum_{p\in\mathbb{P}}\sum_{j\in\mathbb{J}_p}\sum_{s\in\mathbb{S}} t_{kpj}x^{U}_{pjks}\leq T^{t}_{k}, \quad \forall k\in\mathbb{K}, w\in\mathbb{W}^{U}_{k} \tag{6}$$

$$\begin{aligned} & \beta_{pjj'}(\sum_{w\in\mathbb{W}^{U}_{k}}\sum_{s\in\mathbb{S}}x^{U}_{pj'kws} - \sum_{w\in\mathbb{W}^{U}_{k}}\sum_{s\in\mathbb{S}}x^{U}_{pjkws})\leq 0, \\ & \forall p\in\mathbb{P}, j,j'\in\mathbb{J}_p, k\in\mathbb{K} \end{aligned} \tag{7}$$

When assigning products to U-shaped disassembly lines, constraints (2)-(7) must be satisfied. constraint (2) ensures that the sequence of disassembly tasks on the U-shaped line meets the required priority relationships. (3) ensures that the workstations on the U-shaped disassembly line can be started only after the U-shaped disassembly line is opened. (4) ensures that products are assigned only to workstations that have been started. (5) requires that if a disassembly task has prerequisite tasks, these prerequisite tasks must be completed before the task is executed. (6) stipulates that the working time of each U-shaped workstation in the disassembly factory shall not exceed the factory's cycle time. (7) ensures that task assignments comply with the specified conflict relationship constraints.

$$\beta_{pjj'}\Big(\sum_{w\in\mathbb{W}_k^L} w(x_{pjkw}^L - x_{pj'kw}^L) + W_k^L\Big(\sum_{w\in\mathbb{W}_k^L} x_{pj'kw}^L - 1\Big)\Big) \le 0,$$
$$\forall k\in\mathbb{K}, p\in\mathbb{P}, j, j'\in\mathbb{J}_p \tag{8}$$

$$u_{kw}^L \le y_k^L, \forall w\in\mathbb{W}_k^L, k\in\mathbb{K} \tag{9}$$

$$\sum_{p\in\mathbb{P}}\sum_{j\in\mathbb{J}_p} t_{kpj}x_{pjkw}^L \le T_k^t, \forall k\in\mathbb{K}, w\in\mathbb{W}_k^L \tag{10}$$

$$\beta_{pjj'}\Big(\sum_{w\in\mathbb{W}_k^L} x_{pj'kw}^L - \sum_{w\in\mathbb{W}_k^L} x_{pjkw}^L\Big) \le 0, \forall p\in\mathbb{P}, j, j'\in\mathbb{J}_p, k\in\mathbb{K} \tag{11}$$

$$\gamma_{pjj'}\Big(\sum_{w\in\mathbb{W}_k^L}\left(x_{pjkw}^L + x_{pj'kw}^L\right)\Big) \le 1, \forall p\in\mathbb{P}, j, j'\in\mathbb{J}_p, k\in\mathbb{K} \tag{12}$$

$$x_{pjkw}^L \le u_{kw}^L, \forall p\in\mathbb{P}, j\in\mathbb{J}_p, k\in\mathbb{K}, w\in\mathbb{W}_k^L \tag{13}$$

In the process of assigning products to the linear disassembly lines, constraints (8)-(13) must be satisfied to ensure operational efficiency and feasibility. Constraint (8) restricts the task assignment order to meet the priority relationship. (9) requires that after the linear disassembly line in the disassembly factory is opened, its workstations can perform disassembly tasks. (10) stipulates that the cumulative working time of each linear workstation in the disassembly factory shall not exceed the specified cycle time of the factory, so as to ensure that all tasks are completed within the allocated time. (11) ensures that the prerequisite tasks for disassembly tasks have been completed. (12) ensures that task execution causes no conflict. (13) ensures that products are assigned to workstations that have already been started.

$$\sum_{r\in\mathbb{R}}\eta_{krpi} \le \sum_{w\in\mathbb{W}_k^L}\sum_{j\in\mathbb{J}_p}\alpha_{pij}x_{pjkw}^L + \sum_{w\in\mathbb{W}_k^U}\sum_{j\in\mathbb{J}_p}\sum_{s\in\mathbb{S}}\alpha_{pij}x_{pjkws}^U, \forall p\in\mathbb{P}, i\in\mathbb{I}_p, k\in\mathbb{K} \tag{14}$$

$$\sum_{k\in\mathbb{K}} z_{pk} = 1, \forall p\in\mathbb{P} \tag{15}$$

$$z_{pk} = 0, \forall k\in\mathbb{K}, p\in\mathbb{P}, n\in\mathbb{N} \text{ and } \delta_{kn} < \theta_{pn} \tag{16}$$

$$\sum_{w\in\mathbb{W}_k^L} x_{pjkw}^L + \sum_{w\in\mathbb{W}_k^U}\sum_{s\in\mathbb{S}} x_{pjkws}^U \le z_{pk}, \forall p\in\mathbb{P}, k\in\mathbb{K}, j\in\mathbb{J}_p \tag{17}$$

$$y_k^L + y_k^U \le 1, \forall k\in\mathbb{K} \tag{18}$$

$$z_{pk} \le y_k^L + y_k^U, \forall p\in\mathbb{P}, k\in\mathbb{K} \tag{19}$$

$$\sum_{k\in\mathbb{K}}\left(\sum_{w\in\mathbb{W}_k^Z} x_{pjkw}^L + \sum_{w\in\mathbb{W}_k^U}\sum_{s\in\mathbb{S}} x_{pjkws}^U\right) \le 1, \forall p\in\mathbb{P}, j\in\mathbb{J}_p \tag{20}$$

Constraints (14)-(20) must ensure normal operation of the disassembly lines in the factory, while considering the constraints related to the scenario of a remanufacturing factory. (14) stipulates that the components generated from product disassembly can only be transported to one manufacturing factory. (15) stipulates that each product can only be assigned to one disassembly factory. (16) ensures that each product is assigned to a disassembly factory that has the necessary disassembly technology. (17) ensures that the disassembly tasks of products are carried out on the workstations of the activated disassembly lines. (18) ensures that each disassembly plant can only open one type of disassembly line. (19) ensures that products are only assigned to disassembly factories that are open. (20) requires that each disassembly task of each product is performed at most once.

III. Proposed Algorithm

A. Design of Actions and States

State $S_t = [p, w, j_p]$ includes disassembled product p, workstation w, and disassembly task j_p, where $S_t[0] = p$, $S_t[1] = w$, and $S_t[2] = j_p$. The state is updated each time an effective disassembly step is taken. The initial state of S_t is $[0, 0, 0]$, indicating that disassembly has not yet begun. When disassembling products, it starts in order from $p = 1$, and when a product is disassembled, the state is switched and updated with $S_t[0] = p + 1$.

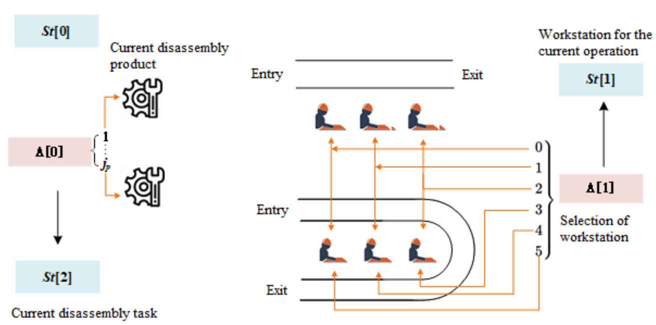


Fig. 3. States and actions.

Action $A = [j_p, w]$ is used to select a product disassembly task j_p and assign it to workstation w. Here, $A[0] = j_p$ and $A[1] = w$. When disassembling product p, based on the current state's disassembly task of product p, the subsequent disassembly task for product p is determined and stored in set J_c. Then, $A[0]$ selects the next disassembly task from J_c and updates $S_t[2]$. Finally, action $A[1]$ determines the workstation based on state $S_t[1]$ and assigns the disassembly task to it. The factory k where the product is assigned and the factory's choice of disassembly line x are provided by the LLM decision.

To determine which disassembly line should be opened in a factory, array L_k is used to record the factory's status, where $L_k[k-1]$ being 1 indicates that factory k opens a linear disassembly line and 2 indicates that factory k opens a U-shaped one. These operations effectively avoid invalid decisions based on transitions between locations and transitions in the Petri net, thereby improving the training effect. The definitions of actions and states are shown in Fig. 3.

B. Reward Design

In the proposed algorithm, the reward is designed as the difference between the maximum profit f_1 obtained in the current state S_t and the maximum profit f_2 obtained in the previous state S_{t-1}, i.e., $R_t = f_1 - f_2$.

Generate results: **Q:** There are 2 disassembly factories and 3 products. Each factory can open at most one linear or U-shaped disassembly line. Each product can only be assigned to one factory. Product 1 yields a profit of 311 on the linear line or 308 on the U-shaped line in factory 1, and 229 on the linear line or 230 on the U-shaped line in factory 2. Product 2 yields a profit of 300 on the linear line or 280 on the U-shaped line in factory 1, and 243 on the linear line or 310 on the U-shaped line in factory 2. Product 3 yields a profit of 324 on the linear line or 301 on the U-shaped line in factory 1, and 235 on the linear line or 301 on the U-shaped line in factory 2. In the format "[factory F opens X disassembly line to disassemble products P.]", where F means factory F, X is the line type, and P means product P. Provide an initial assignment plan. Later, I will give you the total profit under that plan, and you need to adjust the assignments to maximize total profit.
A: [factory 1 opens the linear disassembly line to disassemble products 1 and 3. factory 2 opens the U-shaped disassembly line to disassemble product 2.]

C. LLM Decision-making

LLM employed in this work is a language model pre-trained on large-scale textual data [36, 37]. Its knowledge originates from a broad array of Internet sources, including academic papers, technical documentation, and news reports. Although LLM is not explicitly exposed to the specific scenario of "disassembly line balancing" during training, its knowledge base encompasses several key concepts relevant to this problem—such as factory scheduling, task allocation, resource optimization, and reinforcement learning. Consequently, LLM possesses a certain degree of domain versatility and is capable of applying logical reasoning and generating strategic recommendations related to disassembly-line issues.

In this work, LLM serves as an auxiliary decision-making tool. Through natural-language interactions with users [38], it provides recommendations for the assignment of factories and disassembly lines. Even though it has not been fine-tuned on the specific disassembly-line problem, its strong capabilities in language understanding and inference enable it to perform logical reasoning and generate strategies based on existing knowledge [39]. Below is an illustrative example of our interaction with LLM. This work uses ChatGPT-3.5.

LLM parses the text to extract structured information, such as products (Product 1, Product 2, etc.), factories (factory 1 and factory 2), linear, U-shaped, etc., to construct a profit matrix. It then solves the problem through internal optimization algorithms (such as heuristic search and reinforcement learning.) to find the optimal allocation plan. LLM then outputs the result in natural language, e.g. [factory 1 opens the linear disassembly line to disassemble products 1 and 3. Factory 2 opens the U-shaped disassembly line to disassemble product 2]. The LLM decision result in text is extracted by using regular expressions, and the factories to which the products are allocated are stored in an array K_p with a length equal to the number of products, and the opened disassembly lines are placed in array L_k with a length equal to the number of factories. This process addresses the product allocation and disassembly line selection part of MROP. Calling these two arrays in TD3 simplifies the actions in the TD3 algorithm, providing it with a better initial allocation and enhancing the optimization effect of the TD3 algorithm.

Through repeated interactions with LLM, we feed it with the allocation results of different products such that it continuously refines its memory. Ultimately, LLM will record which factory and disassembly line yields the maximum profit for each product. Consequently, when multiple products are to be disassembled, LLM can immediately provide a tailored allocation plan for each product.

D. Description of the LLM-TD3 algorithm

Fig. 4 illustrates the framework for solving the MROP using LLM-TD3. During training, the TD3 algorithm randomly samples a batch of data from the replay buffer to obtain the state S_t, action A, reward Re, and subsequent state S_t'. Then, the Actor network selects actions A and A' based on the current state. The target Critic network calculates the target Q-values based on the state and actions, taking the smaller of the two as the target value Q'. Meanwhile, It calculates the current estimated Q-values Q_1 and Q_2 based on the state and actions. Finally, the parameters of the Critic1 and Critic2 networks are updated using the mean squared error (TD error) between target value Q' and the current estimated values Q_1 and Q_2. The parameters of the target network are softly updated to gradually approach the parameters of the current network. During the training process, LLM makes some decisions that are provided to the environment and the agent provides action feedback to the environment. The environment then decomposed on the basis of the decisions and actions is obtaining a new state, and placing it into the buffer.

In Algorithm, each episode consists of multiple steps. An episode generates a solution for MROP, and at each step of the episode, the environment sends a new state to the agent based on LLM's decision. The agent observes this state and inputs the feasible disassembly factories, disassembly workstations,

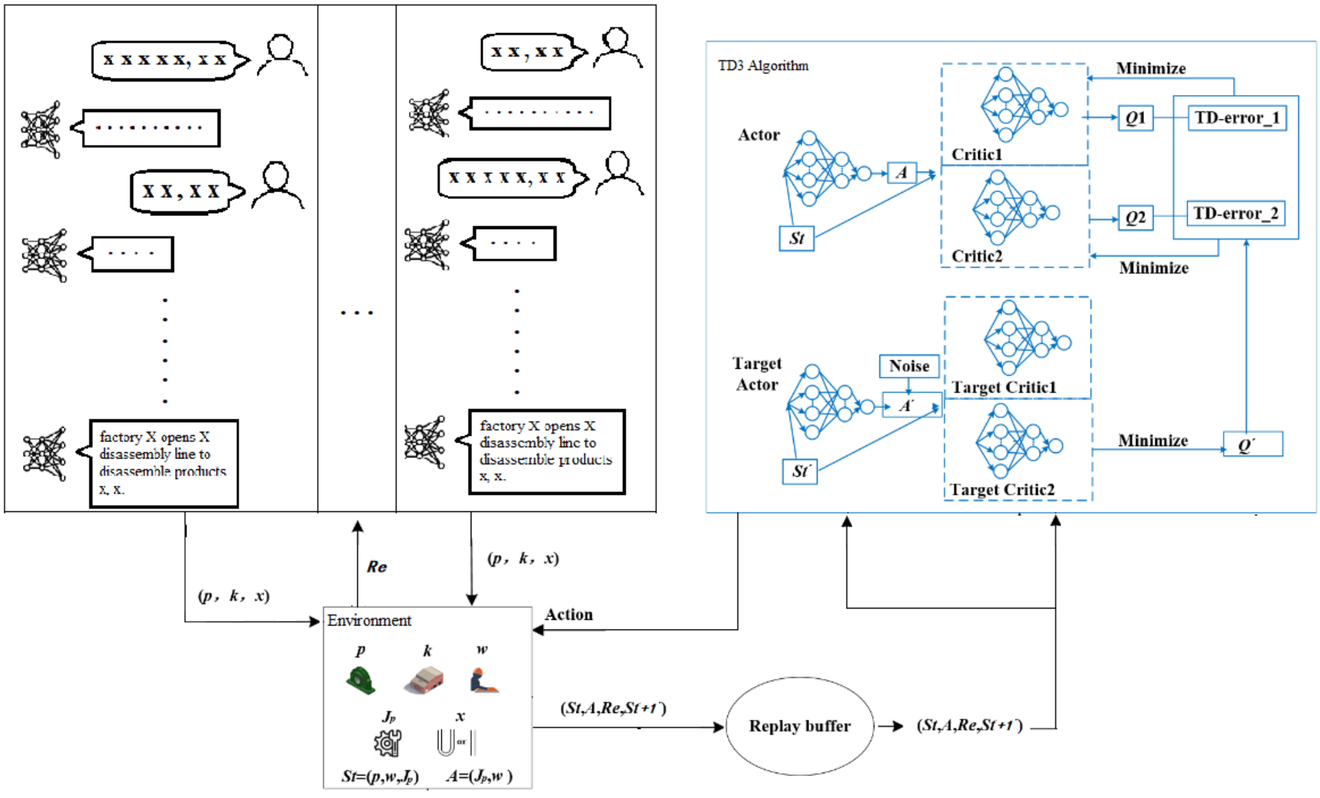


Fig. 4. Framework for solving MROP using LLM-TD3.

and disassembly tasks into its Q-network. Subsequently, it obtains the Q-value of a state and action to guide the agent in selecting the action with a higher Q-value. At the end of the episode, the environment is updated, and then the agent can choose better actions based on the states stored in the replay buffer. The algorithm 1 presents the pseudocode of the LLM-TD3 procedure to solve MROP.

IV. Experimental design and analysis

All algorithms in this work are implemented in PyCharm Community Edition 2022.2.1 and are compiled with Python 3.9.7. Experiments and tests are conducted on a computer equipped with a 3.20 GHz AMD Ryzen 7 5800H CPU and 16 GB of RAM to ensure consistent experimental conditions.

A. Case Design

This work uses washing machines, radios, lithium-ion batteries, and bearing housings as disassembly products for experiments. Table I presents the relevant data input to LLM, where '-' indicates that the disassembly factory does not meet the technical requirements for disassembling a given product. Table II shows the experimental case design. Table III provides the disassembly technology information required for product disassembly.

B. Experimental Design

To evaluate the performance of LLM-TD3 in solving MROP, we compare it with the CPLEX and four existing algorithms:

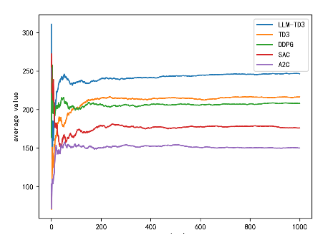


Fig. 5. Training progress for case 1.

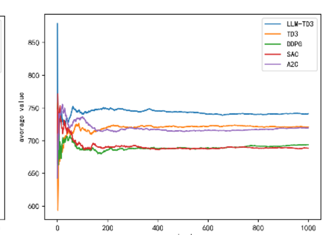


Fig. 6. Training progress for case 2.

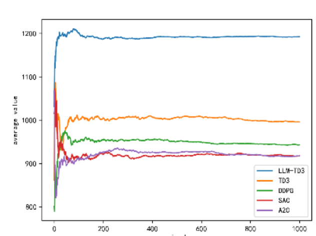


Fig. 7. Training progress for case 3.

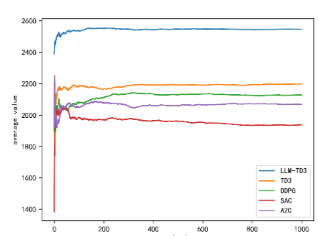


Fig. 8. Training progress for case 4.

TD3 [40], DDPG [41], SAC [5], A2C [42]. The experimental settings of the algorithms in this study are based on the open-source framework provided by Stable-Baselines3. The initialization settings of the algorithms are kept consistent, with a learning rate of 1×10^{-5} and a batch size of 100. The algorithms are run for five trials, with each trial consisting of 1000 iterations.

TABLE I Profits of products on different disassembly lines in factories

Case Number	Product	Factory 1		Factory 2		Factory 3	
		Linear Line	U-shaped	Linear Line	U-shaped	Linear Line	U-shaped
1	Washing Machine	-	-	258	266	-	-
	Bearing Seat	139	135	-	-	-	-
2	Washing Machine	311	308	229	230	-	-
	Bearing Seat	-	-	245	255	-	-
	Radio	443	442	393	399	-	-
3	Washing Machine	-	-	129	139	-	-
	Bearing Seat	-	-	229	239	-	-
	Radio	265	262	-	-	-	-
	Battery	707	711	788	794	-	-
4	Washing Machine 1	106	114	129	139	-	-
	Bearing Seat 1	208	214	229	239	-	-
	Radio 1	265	262	-	-	244	247
	Battery 1	-	-	788	794	-	-
	Washing Machine 2	119	127	142	152	-	-
	Bearing Seat 2	214	220	235	245	-	-
	Radio 2	282	280	-	-	262	266
	Battery 2	-	-	806	812	-	-
5	Washing Machine 1	190	186	216	220	176	180
	Washing Machine 2	216	212	148	152	108	112
	Washing Machine 3	202	198	151	155	263	267
	Bearing Seat 1	-	-	201	205	146	150
	Bearing Seat 2	-	-	264	268	45	49
	Bearing Seat 3	-	-	187	191	46	50
	Radio 1	336	330	323	336	-	-
	Radio 2	272	274	308	312	-	-
	Radio 3	273	274	333	364	-	-
	Battery 1	521	537	634	674	-	-
	Battery 2	580	570	301	305	-	-
	Battery 3	487	483	467	489	-	-
6	Washing Machine 1	227	223	245	249	248	252
	Washing Machine 2	259	256	179	183	180	184
	Washing Machine 3	255	251	172	176	329	333
	Washing Machine 4	233	229	215	219	312	316
	Bearing Seat 1	167	163	186	190	233	237
	Bearing Seat 2	258	254	165	169	224	228
	Bearing Seat 3	173	169	301	305	167	171
	Bearing Seat 4	120	116	177	181	85	89
	Radio 1	403	401	347	358	383	387
	Radio 2	343	340	325	329	369	373
	Radio 3	365	361	381	404	317	321
	Radio 4	425	425	267	276	310	314
	Battery 1	711	717	726	758	770	782
	Battery 2	714	710	507	525	691	707
	Battery 3	939	935	523	541	734	750
	Battery 4	653	659	652	684	802	814

TABLE II Case design

Case Number	Product Quantity	Product				Factory	
		Washing Machine	Bearing Seat	Radio	Lithium Battery	Disassembly Factory	Remanufacturing Factory
1	2	1	1	0	0	2	2
2	3	1	1	1	0	2	2
3	4	1	1	1	1	2	2
4	8	2	2	2	2	3	3
5	12	3	3	3	3	3	6
6	16	4	4	4	4	3	6

TABLE III Disassembly technology information

Case Number	Product Required Disassembly Technology	Factory Owned Disassembly Technology
1	Washing Machine: [1, 0, 0, 1], Bearing Seat: [0, 1, 1, 0]	Factory 1: [1, 1, 1, 1], Factory 2: [1, 0, 0, 1]
2	Washing Machine: [1, 1, 0, 0], Bearing Seat: [0, 1, 0, 1], Radio: [0, 1, 1, 1]	Factory 1: [1, 1, 1, 1], Factory 2: [1, 1, 1, 1]
3	Washing Machine: [1, 0, 0, 1], Bearing Seat: [0, 1, 1, 0], Radio: [0, 1, 1, 1], Lithium Battery: [0, 1, 1, 1]	Factory 1: [0, 1, 1, 1], Factory 2: [1, 1, 1, 1]
4	Washing Machine: [1, 0, 0, 0], Bearing Seat: [0, 1, 0, 1], Radio: [0, 1, 1, 1], Lithium Battery: [0, 1, 1, 1]	Factory 1: [1, 1, 1, 1], Factory 2: [1, 1, 1, 1], Factory 3: [1, 1, 1, 1]
5	Washing Machine: [0, 1, 1, 0], Bearing Seat: [0, 1, 1, 1], Radio: [0, 1, 1, 1], Lithium Battery: [0, 1, 1, 1]	Factory 1: [1, 1, 1, 1]
6	Washing Machine: [0, 0, 1, 0], Bearing Seat: [0, 1, 1, 1], Radio: [0, 1, 1, 1], Lithium Battery: [0, 1, 1, 1]	Factory 2: [1, 1, 1, 1], Factory 3: [1, 1, 1, 1]

C. *Experimental Result Analysis*

Table IV shows the product allocation results for different cases of MROP solved by CPLEX and LLM-TD3. The number before Z or U indicates the disassembly factory number, Z represents that the factory opens a linear disassembly line, and U represents that it opens a U-shaped disassembly one. The number in the parentheses in cases 5 and 6 indicates the corresponding product number. When multiple products of the

TABLE IV Product assignment by CPLEX and LLM-TD3

Case Number	CPLEX Allocation Plan	LLM-TD3 Allocation Plan
1	1Z:{Bearing Seat}, 2U:{Washing Machine}	1Z:{Bearing Seat}, 2U:{Washing Machine}
2	1Z:{Washing Machine, Radio}, 2U:{Bearing Seat}	1Z:{Washing Machine, Radio}, 2U:{Bearing Seat}
3	1Z:{Radio}, 2U:{Washing Machine, Bearing Seat, Lithium Battery}	1Z:{Radio}, 2U:{Washing Machine, Bearing Seat, Lithium Battery}
4	-	1Z:{Radio}, 2U:{Washing Machine, Bearing Seat, Lithium Battery}
5	-	1Z:{Washing Machine(2), Radio(1), Lithium Battery(2, 3)}, 2U:{Washing Machine(1), Bearing Seat, Radio(2, 3), Lithium Battery(1)}, 3U:{Washing Machine(3)}
6	-	1Z:{Washing Machine(2), Bearing Seat(2), Radio(1, 4), Lithium Battery(2, 3)}, 2U:{Washing Machine(1), Bearing Seat(3, 4), Radio(3)}, 3U:{Washing Machine(3, 4), Bearing Seat(1), Radio(2), Lithium Battery(1, 4)}

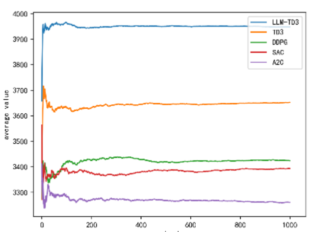


Fig. 9. Training progress for case 5.

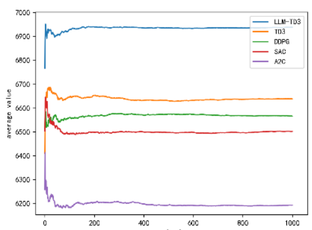


Fig. 10. Training progress for case 6.

TABLE V Maximum profit achieved by different algorithms

Case Number	LLM-TD3	TD3	DDPG	SAC	A2C	CPLEX
1	400	397	382	385	367	405
2	1013	995	956	992	967	1041
3	1420	1392	1317	1381	1332	1480
4	2925	2892	2817	2879	2789	-
5	4267	4067	4026	4111	3983	-
6	7259	7164	7007	7040	6733	-

TABLE VI Mean and standard deviation of profits from different algorithms

Case Number	LLM-TD3	TD3	DDPG	SAC	A2C
1	203.17 ±0.42	187.78 ±0.39	183.04 ±0.41	164.98 ±0.54	145.86 ±0.51
2	728.07 ±0.13	710.09 ±0.14	675.45 ±0.15	685.02 ±0.15	713.19 ±0.13
3	1046.08 ±0.09	919.76 ±0.22	874.97 ±0.18	889.96 ±0.22	914.99 ±0.18
4	2386.10 ±0.07	1994.62 ±0.18	1826.94 ±0.18	1900.94 ±0.15	2068.64 ±0.14
5	3791.34 ±0.04	3417.93 ±0.06	3296.91 ±0.08	3376.49 ±0.07	3248.72 ±0.08
6	6794.91 ±0.02	6489.66 ±0.03	6398.39 ±0.03	6478.58 ±0.02	6181.38 ±0.03

Algorithm 1 State transition process based on LLM

Input: Action $A = [j_p, w]$, state $S_t = [p, w, j_p]$
Output: Next state $S_{t+1} = [p', w', j'_p]$, R_t
Begin
Obtain K_p, L_k from LLM.
if product p has been fully disassembled **then**
  $p' \leftarrow p + 1$
else
  $p' \leftarrow S_t[0]$
end if
if $L_k[K_p[p-1]] == 0$ **then**
  $L_k[K_p[p-1]] \leftarrow A[0]\%2 + 1$
end if
if $L_k[K-1] == 1$ **then**
  $w' \leftarrow A[1] \text{mod number of workstations}$
else
  $w' \leftarrow A[1] \text{ mod } (2\times \text{ number of workstations}) + 1$
end if
if $w' < S_t[1]$ **then**
  $w' \leftarrow S_t[1]$
end if
$j'_p \leftarrow A[0]\%(\text{remaining disassembly tasks}) + 1$
Record obtained components
Obtain $S_{t+1} = [p', w', j'_p]$
if $p', K_p[p'-1], w', j'_p \neq 0$ **then**
  $z_{pk} \leftarrow 1$, $x^u_{pjkws} \leftarrow 1$, $x^L_{pjkw} \leftarrow 1$, $y^u_k \leftarrow 1$, $y^L_k \leftarrow 1$, $u^u_{kw} \leftarrow 1$, $u^L_{kw} \leftarrow 1$
end if
Calculate R_t
return S_{t+1}, R_t
End

same type are assigned to the same disassembly factory, no number is marked. '-' indicates that CPLEX fails to obtain an optimal solution after running for three hours. Looking at the allocation results in Table IV in cases 1-3, the results of the two allocation schemes are exactly the same. This indicates that in these cases, LLM-TD3 can make the best allocation decisions based on the given data, effectively solving the product allocation problem and achieving the same optimal results as CPLEX. Therefore, the allocation schemes provided by LLM-TD3 in cases 4-6. Table IV shows that CPLEX and LLM-TD3 yield the same results in Cases 1-3, whereas Table V reports different results. This discrepancy is attributed to the different workstations selected by the two approaches during task execution.

Next, the performance and effectiveness of LLM-TD3 are evaluated by comparing the maximum values, average values, and standard deviations obtained by different algorithms in solving MROP, as well as the convergence of the average values during training.

Figs. 5 to 10 show the convergence of average rewards for different cases of the MROP solved by five algorithms over 1000 iterations. Each chart represents a case, with the x-axis indicating the number of training episodes and the y-axis indicating the average reward. In all cases, since the differences among the solutions in MROP are not particularly large, the algorithms start with a baseline value during training, followed by some value fluctuations. Note that, describing the training process with the average value can better visualize the training process.

It can be seen from the figures that LLM-TD3 has relatively high stability and a higher final average value in all cases. This is attributed to the excellent decisions made by LLM, which provides a high-quality initial solution for TD3. In cases 1 and 2, although it performs better than its peers, it is not as outstanding as in cases 3-6. This is because cases 1 and 2 have a smaller scale and are easy to solve, while cases 3-6 have a larger scale and are more difficult to solve. Since LLM-TD3 operates based on the excellent decisions made by LLM, it achieves the best results.

Table V shows the maximum profit obtained by the algorithms in each case, where '-' indicates that CPLEX fails after running for three hours. The values obtained by LLM-TD3 are close to those of CPLEX, indicating that LLM-TD3 is effective in solving MROP. Compared to its four peers, LLM-TD3 achieves the highest values in all cases. Table VI shows the total average values and standard deviations after five trials and 1000 iterations in each case. LLM-TD3 has higher average reward values and smaller standard deviations in all cases, indicating its better overall performance than its peers. Therefore, LLM-TD3 has better stability and effectiveness in solving MROP.

V. Conclusion

This work addresses the optimization problem of a remanufacturing system with multiple types of factories and mixed-line layouts. It establishes a collaborative optimization mechanism between the balancing of the disassembly line and the remanufacturing in multiple types of factories. Based on the disassembly technological characteristics of different factories, a product-factory matching decision model is constructed to achieve the optimal assignment of disassembly tasks. The work proposes a mixed-line layout optimization model for remanufacturing in multiple types of factories with the objective of maximizing system profit.

By introducing the intelligent decision-making capability of LLM, the convergence speed and optimization effect of TD3 as a reinforcement learning algorithm are significantly enhanced. Experimental results demonstrate that the proposed LLM-TD3 hybrid optimization framework achieves superior solution efficiency and quality compared to several competitive peers within this specific problem context. These findings validate the potential effectiveness of integrating LLMs with reinforcement learning for addressing complex scheduling challenges in remanufacturing, presenting a promising methodological exploration. Future research will focus on enhancing the framework's robustness, such as addressing LLM output variability and handling dynamic disturbances, to bridge the gap toward real-world application and explore other directions [43, 44, 45].

References

[1] W. X. Zhao and et al., "A survey of large language models," *arXiv preprint arXiv:2303.18223*, vol. 1, no. 2, 2023.

[2] S. Liu, C. Chen, X. Qu, K. Tang, and Y.-S. Ong, "Large language models as evolutionary optimizers," in *2024 IEEE Congress on Evolutionary Computation (CEC)*, 2024, pp. 1–8.

[3] E. Eigner and T. Händler, "Determinants of LLM-assisted decision-making," *arXiv preprint arXiv:2402.17385*, 2024.

[4] J. Gu, J. Wang, X. Guo, G. Liu, S. Qin, and Z. Bi, "A metaverse-based teaching building evacuation training system with deep reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 4, pp. 2209–2219, 2023.

[5] 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.

[6] Z. Bi, X. Guo, J. Wang, S. Qin, and G. Liu, "Deep reinforcement learning for truck-drone delivery problem," *Drones*, vol. 7, no. 7, 2023.

[7] X. Shen, "Comparison of DDPG and TD3 algorithms in a walker2d scenario," in *2023 Int Conf on Data Science, Advanced Algorithm and Intelligent Computing (DAI 2023)*. Atlantis Press, 2024, pp. 148–155.

[8] H. Li, J. Zhu, Y. Zhou, Q. Feng, and D. Feng, "Charging station management strategy for returns maximization via improved TD3 deep reinforcement learning," *International Transactions on Electrical Energy Systems*, vol. 2022, no. 1, p. 6854620, 2022.

[9] E. Altman, *"Constrained Markov Decision Processes"*, 2021.

[10] Y. Yao and et al., "A survey on large language model (LLM) security and privacy: The good, the bad, and the ugly," *High-Confidence Computing*, p. 100211, 2024.

[11] D. Nam and et al., "Using an LLM to help with code understanding," in *Proc. of the IEEE/ACM 46th International Conference on Software Engineering*, 2024, pp. 1–13.

[12] X. Lin and et al., "Data-efficient fine-tuning for LLM-based recommendation," in *Proc. of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024, pp. 365–374.

[13] J. Zhou and et al., "A novel energy management strategy of hybrid electric vehicle via an improved TD3 deep reinforcement learning," *Energy*, vol. 224, p. 120118, 2021.

[14] F. Zhang, J. Li, and Z. Li, "A TD3-based multi-agent deep reinforcement learning method in mixed cooperation-competition environment," *Neurocomputing*, vol. 411, pp. 206–215, 2020.

[15] H. Yin and et al., "Deep reinforcement learning with multi-critic TD3 for decentralized multi-robot path planning," *IEEE Transactions on Cognitive and Developmental Systems*, 2024.

[16] S. Wang, C. Zhang, and J. Wan, "A smart factory solution to hybrid production of multi-type products with reduced intelligence," in *2016 IEEE Information Technology, Networking, Electronic and Automation Control Conference*, 2016, pp. 848–853.

[17] X. Guo, T. Wei, J. Wang, S. Liu, S. Qin, and L. Qi, "Multiobjective U-shaped disassembly line balancing problem considering human fatigue index and an efficient solution," *IEEE Transactions on Computational Social Systems*, vol. 10, no. 4, pp. 2061–2073, 2022.

[18] Z. Li and M. N. Janardhanan, "Modelling and solving profit-oriented U-shaped partial disassembly line balancing problem," *Expert Systems with Applications*, vol. 183, p. 115431, 2021.

[19] F. T. Altekin, L. Kandiller, and N. E. Ozdemirel, "Profit-oriented disassembly-line balancing," *International Journal of Production Research*, vol. 46, no. 10, pp. 2675–2693, 2008.

[20] Y. Ren and et al., "A review of combinatorial optimization problems in reverse logistics and remanufacturing for end-of-life products," *Mathematics*, vol. 11, no. 2, p. 298, 2023.

[21] M. Kerin and D. T. Pham, "Smart remanufacturing: a review and research framework," *Journal of Manufacturing Technology Management*, vol. 31, no. 6, pp. 1205–1235, 2020.

[22] 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.

[23] J. Mao and et al., "Disassembly sequence planning of waste auto parts," *Journal of the Air & Waste Management Association*, vol. 71, no. 5, pp. 607–619, 2021.

[24] 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, 2024.

[25] K. P. Sycara, S. F. Roth, N. Sadeh, and M. S. Fox, "Resource allocation in distributed factory scheduling," *IEEE Expert*, vol. 6, no. 1, pp. 29–40, 1991.

[26] N. Karimi and H. Davoudpour, "A knowledge-based approach for multi-factory production systems," *Computers & Operations Research*, vol. 77, pp. 72–85, 2017.

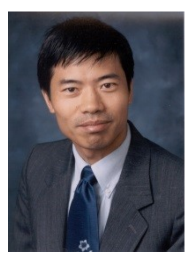
[27] I. Grobelna and A. Karatkevich, "Challenges in application of petri nets

in manufacturing systems," *Electronics*, vol. 10, no. 18, 2021.
[28] B. Huang, M. Zhou, X. S. Lu, and A. Abusorrah, "Scheduling of resource allocation systems with timed petri nets: A survey," *ACM Comput. Surv.*, vol. 55, no. 11, 2023.
[29] F. Liu, G. Assaf, M. Chen, and M. Heiner, "A petri nets-based framework for whole-cell modeling," *Biosystems*, vol. 210, p. 104533, 2021.
[30] M. C. Zhou and F. Dicesare, *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. Kluwer Academic, 1993.
[31] Y. Fan, "Modeling, simulation, and control of flexible manufacturing systems – a petri net approach; meng chu zhou; kurapati venkatesh; yushun fan; world scientific, singapore, 1999," *Mechatronics*, vol. 11, no. 7, pp. 947–950, 2001.
[32] J. Wang, "Charging information collection modeling and analysis of gprs networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 4, pp. 473–481, 2007.
[33] J. Zhou, J. Wang, and J. Wang, "A simulation engine for stochastic timed petri nets and application to emergency healthcare systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 4, pp. 969–980, 2019.
[34] Y. Ren and et al., "A self-adaptive learning approach for uncertain disassembly planning based on extended petri net," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 12, pp. 11 889–11 897, 2023.
[35] J. Mao and et al., "Disassembly sequence planning of waste auto parts," *Journal of the Air & Waste Management Association*, vol. 71, no. 5, pp. 607–619, 2021.
[36] P. Kumar, "Large language models (LLMs): survey, technical frameworks, and future challenges," *Artificial Intelligence Review*, vol. 57, no. 10, p. 260, 2024.
[37] J. Wu and et al., "A survey on LLM-generated text detection: Necessity, methods, and future directions," *Computational Linguistics*, pp. 1–66, 2025.
[38] J. Yang and et al., "Harnessing the power of llms in practice: A survey on chatgpt and beyond," *ACM Transactions on Knowledge Discovery from Data*, vol. 18, no. 6, pp. 1–32, 2024.
[39] N. Karanikolas and et al., "Large language models versus natural language understanding and generation," in *Proc. of the 27th Pan-Hellenic Conference on Progress in Computing and Informatics*, 2023, pp. 278–290.
[40] L. Qi and et al., "Twin delayed deep deterministic policy gradient algorithm for a heterogeneous multifactory remanufacturing optimization problem," *IEEE Transactions on Computational Social Systems*, pp. 1–12, 2025.
[41] E. H. Sumiea and et al., "Deep deterministic policy gradient algorithm: A systematic review," *Heliyon*, vol. 10, no. 9, 2024.
[42] S. Qin, X. Xie, J. Wang, X. Guo, L. Qi, W. Cai, Y. Tang, and Q. T. A. Talukder, "An optimized advantage actor-critic algorithm for disassembly line balancing problem considering disassembly tool degradation," *Mathematics*, vol. 12, no. 6, p. 836, 2024.
[43] Q. Zhang, Y. Xing, C. Zhang, X. Sun, B. Hu, and A. Das, "Column generation algorithms for two-dimensional cutting problem with surface defects," *International Journal of Artificial Intelligence and Green Manufacturing*, vol. 1, no. 2, pp. 23–35, 2025.
[44] L. Zhou, H. Zhu, and B. Akbari, "Multi-objective optimization of multi-factory remanufacturing process considering worker fatigue," *International Journal of Artificial Intelligence and Green Manufacturing*, vol. 1, no. 2, pp. 36–50, 2025.
[45] H. Zhang, D. Pham, and Q. Kang, "Improved fruit fly algorithm for multi-objective disassembly line balancing problem considering learning effect," *International Journal of Artificial Intelligence and Green Manufacturing*, vol. 1, no. 2, pp. 51–62, 2025.

Xiwang Guo received his Ph.D. degree in System Engineering from Northeastern University, Shenyang, China, in 2015. He is currently an Associate Professor of the College of Computer and Communication Engineering at Liaoning Petrochemical University. From 2016 to 2018, he was a visiting scholar of Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA. He has published 200+ papers in journals and conference proceedings, including the IEEE, Transactions on Intelligent Transportation Systems (TITS), CAA Journal of Automatica Sinica (JAS), Transactions on Systems, Man, and Cybernetics: Systems (TSMCS), Transactions on Computational Social Systems (TCSS), Transactions on Automation Science and Engineering (TASE), Transactions on Cybernetics (TCYB), Internet of Things Journal (IoT-J), Transactions on Artificial Intelligence (TAI), and Robotics and Automation Letters (RA-L). His current research interests include optimization algorithms, reinforcement learning, manufacturing, and intelligent transportation.

Yujie Feng received her Bachelor 's degree in Computer Science and Technology from Jilin Jianzhu University in China in 2023.

She is now a graduate student of the School of Artificial Intelligence and Software, Liaoning University of Petrochemical Technology. Her research interests include robotics learning.



MengChu Zhou (Fellow, IEEE) received his B.S. degree in Control Engineering from Nanjing University of Science and Technology, Nanjing, China in 1983, M.S. degree in Automatic Control from Beijing Institute of Technology, Beijing, China in 1986, and Ph. D. degree in Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY in 1990. He joined the Department of Electrical and Computer Engineering, New Jersey Institute of Technology in 1990, and is now a Distinguished Professor. His interests are in intelligent automation/transportation, robotics, Petri nets, Internet of Things, edge/cloud computing, and big data analytics. He has over 1400 publications including 17 books, over 900 journal papers including over 700 IEEE Transactions papers, 31 patents and 32 book-chapters. He is a recipient of Excellence in Research Prize and Medal from NJIT, Humboldt Research Award for US Senior Scientists from Alexander von Humboldt Foundation, and Franklin V. Taylor Memorial Award and the Norbert Wiener Award from IEEE Systems, Man, and Cybernetics Society, and Edison Patent Award from the Research & Development Council of New Jersey. He is a life member of Chinese Association for Science and Technology-USA and served as its President in 1999. He is Fellow of IEEE, International Federation of Automatic Control (IFAC), American Association for the Advancement of Science (AAAS), Chinese Association of Automation (CAA) and National Academy of Inventors (NAI).