Hybrid-Line Remanufacturing Process Optimization for Multi-Type Factories with Twin Delayed Deep Deterministic Policy Gradient Algorithm

Jinlei Gu, Yujie Feng, and Vladislav D. Veksler

***Abstract*—In response to the growing complexity of remanufacturing systems, this work investigates a novel Multi-objective Hybrid-line Multi-type Factory Remanufacturing Optimization Problem. The proposed model takes into account the disassembly technologies used by different factories, the selection of heterogeneous disassembly lines, and task-related constraints such as precedence and conflicts. The goal is to assign end-of-life products to appropriate disassembly factories and schedule tasks on optimal lines to achieve high scalability and efficiency in large-scale sequential decision environments. To solve this problem, we formulate a multi-objective mixed-integer programming model that simultaneously maximizes overall profit and minimizes factory cycle time. The model is validated using a commercial solver to ensure feasibility and correctness. Due to the large-scale and sequential decision nature of the problem, we also employ the Twin Delayed Deep Deterministic Policy Gradient Algorithm (TD3), TD3 for short, to learn optimal strategies through interaction with the environment. Experimental studies in various benchmark cases show that TD3 significantly outperforms baseline reinforcement learning algorithms such as Deep Deterministic Policy Gradient (DDPG), Soft Actor-Critic (SAC), and Advantage Actor-Critic (A2C) in both convergence stability and solution quality. TD3 also demonstrates superior capability in approximating Pareto-optimal solutions, which makes it suitable for real-world remanufacturing scenarios.**

***Key Words*—Multi-objective optimization, reinforcement learning, twin delayed deep deterministic policy gradient algorithm, remanufacturing process.**

I. Introduction

With the acceleration of economic globalization and rapid socio-economic development, issues of resource consumption and environmental pollution have become increasingly prominent. The traditional linear economic model of "production–consumption–disposal" fails to meet the requirements of sustainable development, resulting in significant resource waste and ecological pressure. In response, the concepts of circular economy and green manufacturing have attracted widespread attention [1, 2]. The circular economy emphasizes efficient resource utilization and recycling, while green manufacturing aims to reduce environmental impact across the entire product lifecycle. In this context, remanufacturing serves as a key approach for resource regeneration. By repairing and restoring used products to near-new performance, remanufacturing offers substantial economic and environmental benefits compared to traditional manufacturing [3–5].

Multi-type factory remanufacturing constitutes a complex systems engineering challenge, with the core objective of efficiently integrating heterogeneous factory resources to recover, disassemble, and remanufacture end-of-life (EOL) products [6–8]. In practice, different products have diverse requirements for disassembly technologies, tools, and production line capabilities, and factories differ in terms of resources, capacities, and technological [9–11]. This heterogeneity poses challenges in demand variability, resource allocation, and coordinated scheduling across multiple factories.

Although extensive research has explored optimization in remanufacturing systems, most existing studies focus on single-factory scenarios or assume homogeneous disassembly equipment and line structures [12–15], mainly addressing local issues such as task scheduling, workstation balancing, or resource allocation, while neglecting the heterogeneity of real industrial networks. More recent studies have begun to investigate broader system-level perspectives. For example, Tolio *et al.* [16] reviewed advanced de- and remanufacturing systems and emphasized the need for flexible and adaptive system designs to support circular economy operations. Mejia *et al.* [17] proposed a smart architecture for sustainable manufacturing–remanufacturing systems based on Industry 4.0 technologies, focusing on information integration and system-level coordination. Ke *et al.* [18] developed a multi-objective optimization framework for remanufacturing process design that simultaneously considers performance, cost, and carbon emissions. However, these studies mainly address system architecture or process design rather than operational scheduling decisions in multi-factory environments[19, 20]. In particular, disassembly technology selection, factory assignment, and hybrid-line task scheduling are typically modeled separately. Consequently, the joint optimization of heterogeneous factory technologies, hybrid production line balancing, and task precedence/conflict constraints remains insufficiently explored. In addition, many existing approaches rely on heuristic or local search methods, which struggle to scale in high-dimensional sequential decision spaces.

To address these challenges, this study introduces the Hybrid-line Multi-type Factory Remanufacturing Optimization Problem (HMRO), a unified multi-objective framework that simultaneously considers factory allocation, disassembly technology selection, task scheduling, and hybrid-line balancing,

Manuscript received February 18, 2026; revised February 25 and March 4, 2026; accepted March 20, 2026. This article was recommended for publication by Associate Editor Shujin Qin upon evaluation of the reviewers' comments.

J. Gu is with the New Jersey of Institute of Technology, Newark, NJ, USA (e-mail: gujinlei0707@gmail.com).

Y. Feng is with the Artificial Intelligence and Software College, Liaoning Petrochemical University, Fushun 113001, China (e-mail: xf1553@163.com).

V. Veksler is with the Department of Computer Science, Caldwell University, Caldwell, NJ 07006, USA (e-mail: vdv718@gmail.com).

Corresponding Author: Jinlei Gu.

aiming to maximize profit, minimize production cycle time, and ensure feasible disassembly sequences.

In multi-type factory remanufacturing, products must be allocated to appropriate factories according to disassembly technology requirements to ensure safety and efficiency [21–24]. Factory production lines, including layout, equipment, and personnel scheduling, must achieve balanced task assignment and cycle synchronization while satisfying precedence and workstation constraints. Recovered components are further inspected, classified, and transported to higher-value remanufacturing facilities [25, 26], with logistics planning considering quantity, distance, and cost [27].

HMRO problem exhibits three core structural challenges that exacerbate its complexity [18, 28]: (1) Combinatorial explosion due to multi-dimensional decision spaces, where product-factory-line assignments and task sequencing generate exponentially growing solution spaces. (2) Interdependence across decision layers, as factory selection dictates feasible disassembly technologies, which in turn constrain task precedence and workstation allocation. (3) Sensitivity to technological feasibility, where incompatibility between product requirements and factory capabilities invalidates entire solution branches. To solve this high-dimensional, continuous, and coupled optimization problem, we employ the Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm [29, 30]. Compared with DDPG [31], TD3 leverages double Q-networks, delayed policy updates, and target policy smoothing to improve training stability and generalization [32–35], enabling efficient learning of near-optimal allocation and scheduling policies in complex scheduling environments.

The main contributions of this work are as follows.

1) Problem Formulation: We introduce the Multi-objective Hybrid-line Multi-type Factory Remanufacturing Optimization Problem (HMRO), a novel formulation that integrates the practical complexities of disassembly technology selection and line type balancing within a multi-factory remanufacturing environment. This work extends the literature by bridging a significant gap between theoretical models and real-world industrial challenges.
2) Mathematical Modeling: A multi-objective mixed-integer programming model is formulated to simultaneously maximize profit and minimize factory cycle time. The correctness and feasibility of this model are rigorously verified using a commercial solver, providing a solid mathematical foundation for the HMRO problem.
3) Methodological Innovation: We propose a novel application framework that pioneers the use of the TD3 algorithm to solve the complex HMRO. Our work demonstrates that the advanced design of TD3 enables it to better approximate Pareto-optimal solutions and effectively handle the problem's complexity. Experimental results across various benchmark cases show that our TD3-based approach achieves superior convergence and performance compared to other state-of-the-art DRL methods like DDPG, Soft Actor-Critic (SAC), and Advantage Actor-Critic (A2C).

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

II. Problem Description

A. Problem Statement

This work investigates the HMRO, aiming to maximize overall profit and minimize factory cycle time, while considering the technological capabilities of disassembly factories and the selection of disassembly line layouts. Each disassembly factory can choose between linear or U-shaped disassembly lines for processing products, and the layout type significantly impacts disassembly efficiency and cost. Furthermore, the technological capacity of each factory imposes constraints on the types of products it can process.

Profit calculation takes into account the acquisition value of components, transportation costs, disassembly costs, and the fixed costs of opening disassembly factories and workstations. The factory cycle time is determined by the workstation with the longest processing time in each factory. The goal is to balance disassembly times between workstations, thus achieving load balance between factories.

Fig. 1 illustrates the overall framework of the HMRO. The mixed-layout multi-type factory remanufacturing process extends the conventional multi-type factory remanufacturing optimization problem [36–38] by considering the disassembly lines operated within disassembly factories. Accordingly, HMRO includes the following four key components.

1) Product allocation

The HMRO involves multiple disassembly plants and remanufacturing plants. In real-world situations, different disassembly plants possess distinct disassembly technologies. These technologies determine which products a plant can process. For instance, the disassembly of certain complex products may require advanced specific technologies that not all plants possess. Therefore, the allocation process must ensure that the technological capabilities of the plants match the technical requirements for disassembling the products. In this work, we consider five categories of disassembly technologies, and products are assigned to plants that meet their technical needs.

In order to better describe the relationship between product and technology, we define the following two matrices.

The product-technology association matrix $B = [\beta_{pn}]$ describes the relationship between product and disassembly technology. Where p stands for product number and n stands for disassembly technology number.

$$\beta_{pn} = \begin{cases} 1, & \text{The disassembly of product } p \text{ requires} \\ & \text{technology } n. \\ 0, & \text{Otherwise.} \end{cases}$$

The factory-technology association matrix $Y = [\gamma_{kn}]$ describes the relationship between disassembly factory and disassembly technology. Where k represents the disassembly factory number and n represents the disassembly technology number.

$$\gamma_{kn} = \begin{cases} 1, & \text{Disassembly factory } k \text{ has technology } n. \\ 0, & \text{Otherwise.} \end{cases}$$

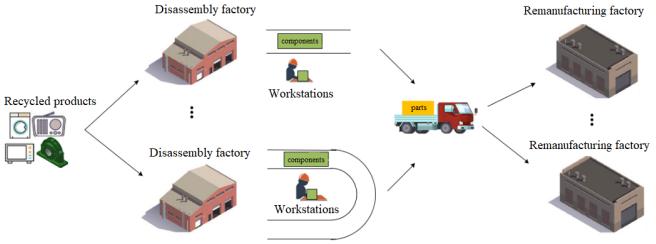

Fig. 1. Framework of the HMRO.

2) Selection of disassembly lines

Each disassembly plant can select either a straight-line or U-shaped disassembly line for the disassembly operation based on actual needs. The straight-line layout is suitable for disassembly tasks with simple structures and continuous processes, while the U-shaped layout is more appropriate for complex disassembly scenarios that involve the coordination of multiple tasks. The choice of layout will significantly impact disassembly efficiency, operating costs, and workstation utilization. The plant will choose the most suitable disassembly line layout according to the type of products assigned and the disassembly requirements.

3) Task scheduling

The disassembly tasks of products must be effectively allocated to suitable workstations. The process takes into account factors such as the plant's cycle time, priority, and task conflicts, for example, different tasks take different amounts of time to disassemble. This phase ensures that each workstation has sufficient time to complete the assigned tasks [39].

4) Transportation of disassembled components

The components obtained from disassembly will be transported to the remanufacturing plants to maximize the overall profit. The transportation process takes into account the recovery value of the components at different remanufacturing locations, as well as the transportation costs between the disassembly plants and the remanufacturing plants [40–42].

The interdependence of subproblems is formally modeled as follows: Product allocation (via z_{pk}) determines the feasible factory set K_c for disassembly. Factory selection activates specific line types (y^Z_k, y^U_k), which constrain task-workstation assignments (x^Z_{pjkw}, x^U_{pjkwl}). Task scheduling further depends on precedence/conflict matrices (Q^M, C^M) that vary by product. This coupling necessitates integrated optimization rather than sequential decomposition.

To better describe the relationships between products and technologies, as well as between product components and tasks, this work defines five matrices, which are specified as follows.

1) The incidence matrix $D^M = [d^M_{pij}]$ describes the relationship between components and disassembly tasks, where i is components, j is tasks and p stands for product number.

$$d^M_{pij} = \begin{cases} 1, & \text{Executing task } j \text{ of product } p \text{ obtains component } i. \\ -1, & \text{Executing task } j \text{ of product } p \text{disassembles component } i. \\ 0, & \text{Otherwise.} \end{cases}$$

2) The precedence matrix $Q^M = [q^M_{pj_1j_2}]$ describes the relationship between two tasks, where j_1 and j_2 represent disassembly tasks and p represents product number.

$$q^M_{pj_1j_2} = \begin{cases} 1, & \text{If task } j_1 \text{ can be performed before task } j_2 \text{ in product } p. \\ 0, & \text{Otherwise.} \end{cases}$$

3) The conflict matrix $C^M = [c^M_{pj_1j_2}]$ describes the conflict between two tasks, where j_1 and j_2 represent disassembly tasks and p represents product number.

$$c^M_{pj_1j_2} = \begin{cases} 1, & \text{If task } j_1 \text{ is conflict with task } j_2 \text{ in product } p. \\ 0, & \text{Otherwise.} \end{cases}$$

B. Mathematical Model

1) Notations

$\mathbb{K}$ Set of all disassembly factory, $\mathbb{K}$={1,2,...,K}.
$\mathbb{R}$ Set of all remanufacturing factory, $\mathbb{R}$={1,2,...,R}.
$\mathbb{P}$ Set of all products, $\mathbb{P}$={1,2,...,P}.
$\mathbb{I}_p$ Set of all subassemblies in product p, $\mathbb{I}_p = \{1, 2, ..., I_p\}$.
$\mathbb{J}_p$ Set of all tasks in product p, $\mathbb{J}_p = \{1, 2, \ldots, J_p\}$.
$\mathbb{W}^U_k$ Set of U-shaped disassembly workstations in disassembly factory k, $\mathbb{W}^U_k = \{1, 2, ..., W^U_k\}$.
$\mathbb{W}^Z_k$ Set of linear disassembly workstations in disassembly factory k, $\mathbb{W}^Z_k = \{1, 2, ..., W^Z_K\}$.
$\mathbb{L}$ Set of side of U-shaped workstations, $\mathbb{L} = \{1, 2\}$.
$\mathbb{N}$ Set of disassembly techniques, $\mathbb{N}$={1,2,...,N}.
v_{rpi} The price at which remanufacturing factory r acquires the i-th component of product p.
c^T_{krpi} The cost of transporting the i-th component of product p from disassembly factory k to remanufacturing factory r.
t_{kpj} Time spent by workers in disassembly factory k to perform task j of product p.
c^D_{kpj} Unit time cost for workers in disassembly factory k to perform task j of product p.
c^O_k Unit time cost of opening of disassembly factory k.
c^U_{kw} Fixed cost of opening the w-th U-shaped workstation in disassembly factory k.
c^Z_{kw} Fixed cost of opening the w-th linear workstation in disassembly factory k.
T^T_k, The cycle time of disassembly factory k.

2) Decision variables

$$z_{pk} = \begin{cases} 1, & \text{Product } p \text{ is performed at disassembly factory } k. \\ 0, & \text{Otherwise.} \end{cases}$$

$$x^Z_{pjkw} = \begin{cases} 1, & \text{Disassembly task } j \text{ of product } p \\ & \text{is to linear workstation } W^Z_k \\ & \text{in disassembly factory } k. \\ 0, & \text{Otherwise.} \end{cases}$$

$$x^U_{pjkwl} = \begin{cases} 1, & \text{Disassembly task } j \text{ of product } p \\ & \text{is performed on side } l \text{ of workstation } W^U_k \\ & \text{in disassembly factory } k. \\ 0, & \text{Otherwise.} \end{cases}$$

$$y^Z_k = \begin{cases} 1, & \text{Open linear disassembly line of} \\ & \text{disassembly factory } k. \\ 0, & \text{Otherwise.} \end{cases}$$

$$y^U_k = \begin{cases} 1, & \text{Open U-shaped disassembly line of} \\ & \text{disassembly factory } k. \\ 0, & \text{Otherwise.} \end{cases}$$

$$u^Z_{kw} = \begin{cases} 1, & \text{Open the linear workstation } w \text{ of disassembly} \\ & \text{factory } k. \\ 0, & \text{Otherwise.} \end{cases}$$

$$u^U_{kw} = \begin{cases} 1, & \text{Open the U-shaped workstation } w \text{ of} \\ & \text{disassembly factory } k. \\ 0, & \text{Otherwise.} \end{cases}$$

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

3) Assumption

Excluding the interference of other non-essential factors, the following assumptions are made in this work.

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

4) Objective function

$$\begin{aligned} f_1 = \max\Big(& \sum_{k\in\mathbb{K}}\sum_{r\in\mathbb{R}}\sum_{p\in\mathbb{P}}\sum_{i\in\mathbb{I}_p}(v_{rpi} - c^T_{krpi})a_{krpi} \\ & - \sum_{k\in\mathbb{K}}\sum_{p\in\mathbb{P}}\sum_{j\in\mathbb{I}_p}\sum_{w\in\mathbb{W}^Z_k} c^D_{kpj}t_{kpj}x^Z_{pjkw} \\ & - \sum_{k\in\mathbb{K}}\sum_{p\in\mathbb{P}}\sum_{j\in\mathbb{J}_p}\sum_{w\in\mathbb{W}^U_k}\sum_{l\in\mathbb{L}} c^D_{kpj}t_{kpj}x^U_{pjkwl} - \sum_{k\in\mathbb{K}} c^O_k T^T_k \\ & - \sum_{k\in\mathbb{K}}\sum_{w\in\mathbb{W}^Z_k} c^Z_{kw}u^Z_{kw} - \sum_{k\in\mathbb{K}}\sum_{w\in\mathbb{W}^U_k} c^U_{kw}u^U_{kw}\Big) \end{aligned} \tag{1}$$

$$f_2 = \min(\max_{k\in\mathbb{K}} T^T_k) \tag{2}$$

The objective function f_1 represents the maximization of disassembly profit. The first term is the profit obtained by subtracting the transportation cost from the product components. The second and third terms are the costs of disassembling products on straight and U-shaped disassembly lines. The fourth term is the fixed cost of starting up the disassembly plant. The fifth and sixth terms are the fixed costs of starting up the workstations on the two disassembly lines. The objective function f_2, represents the minimization of the disassembly factory cycle time.

5) Constraints related to linear disassembly lines

$$u^Z_{kw} \le y^Z_k, \forall w \in \mathbb{W}^Z_k, \forall k \in \mathbb{K} \tag{3}$$

$$x^Z_{pjkw} \le u^Z_{kw}, \forall p \in \mathbb{P}, \forall j \in \mathbb{J}_p, \forall k \in \mathbb{K}, \forall w \in \mathbb{W}^Z_k \tag{4}$$

$$\sum_{w\in\mathbb{W}^Z_k} w\left(x^Z_{pj_1kw} - x^Z_{pj_2kw}\right) + W^Z_k\left(\sum_{w\in\mathbb{W}^Z_k} x^Z_{pj_2kw} - 1\right) \le 0, \quad \forall k \in \mathbb{K}, \forall p \in \mathbb{P}, \forall j_1, j_2 \in \mathbb{J}_p \text{ and } q^M_{pj_1j_2} = 1 \tag{5}$$

$$q^M_{pj_1j_2}\left(\sum_{w\in\mathbb{W}^Z_k} x^Z_{pj_2kw} \sum_{w\in\mathbb{W}^Z_k} x^Z_{pj_1kw}\right) \le 0, \forall k \in \mathbb{K}, \quad \forall p \in \mathbb{P}, \forall j_1, j_2 \in \mathbb{J}_p \tag{6}$$

$$\sum_{w\in W^Z_k}\left(x^z_{pj_1kw} + x^z_{pj_2kw}\right) \le 1, \forall k \in \mathbb{K}, \forall p \in \mathbb{P}, \quad \forall j_1, j_2 \in \mathbb{J}_p \text{ and } c^M_{pj_1j_2} = 1 \tag{7}$$

$$\sum_{p\in\mathbb{P}}\sum_{j\in\mathbb{J}_p} t_{kpj}x^Z_{pjkw} \le T^T_k, \forall k \in \mathbb{K}, \forall w \in \mathbb{W}^Z_k \tag{8}$$

Constraint (3) ensures that the straight disassembly line in the disassembly plant where the straight workstation is opened is also opened. Constraint (4) ensures that products are assigned to the opened straight workstations. Constraint (5) ensures that the assignment of disassembly tasks on the disassembly line meets the priority relationship constraints. Constraint (6) ensures that the previous task of the disassembly task is assigned before it. Constraint (7) ensures that the

disassembly tasks meet the conflict relationship constraints. Constraint (8) ensures that the working time of each straight workstation in the disassembly plant cannot exceed the cycle time of the disassembly plant.

6) Constraints related to U-shaped disassembly lines

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

$$x_{pjkwl}^{U} \le u_{kw}^{U}, \forall p \in \mathbb{P}, \forall j \in \mathbb{J}_p, \forall k \in \mathbb{K}, \forall w \in \mathbb{W}_k^{U}, \forall l \in \mathbb{L} \tag{10}$$

$$\sum_{w \in \mathbb{W}_k^{U}} (w(x_{pj_1kw_1}^{U} - x_{pj_2kw_1}^{U}) + (2W_k^{U} - w) (x_{pj_1kw_2}^{U} - x_{pj_2kw_2}^{U})) + 2W_k^{U}\Big(\sum_{w \in \mathbb{W}_k^{U}} \sum_{l \in \mathbb{L}} x_{pj_2kwl}^{U} - 1\Big) \le 0, \forall k \in \mathbb{K}, \forall p \in \mathbb{P}, \forall j_1, j_2 \in \mathbb{J}_p \text{ and } q_{pj_1j_2}^{M} = 1 \tag{11}$$

$$\sum_{w \in \mathbb{W}_k^{U}} \sum_{l \in \mathbb{L}} (x_{pj_1kwl}^{U} + x_{pj_2kwl}^{U}) \le 1, \forall k \in \mathbb{K}, \forall p \in \mathbb{P}, \forall j_1, j_2 \in \mathbb{J}_p \text{ and } c_{pj_1j_2}^{M} = 1 \tag{12}$$

$$q_{pj_1j_2}^{M}\left(\sum_{w \in \mathbb{W}_k^{U}} \sum_{l \in \mathbb{L}} x_{pj_2kwl}^{U} - \sum_{w \in \mathbb{W}_k^{U}} \sum_{l \in \mathbb{L}} x_{pj_1kwl}^{U}\right) \le 0, \forall k \in \mathbb{K}, \forall p \in \mathbb{P}, \forall j_1, j_2 \in \mathbb{J}_p \tag{13}$$

$$\sum_{p \in \mathbb{P}} \sum_{j \in \mathbb{J}_p} \sum_{l \in \mathbb{L}} t_{kpj} x_{pjkwl}^{U} \le T_k^{T}, \quad \forall k \in \mathbb{K}, \forall w \in \mathbb{W}_k^{U} \tag{14}$$

Constraint (9) ensures that the U-shaped disassembly line in the disassembly plant where the U-shaped workstation is located is open. Constraint (10) ensures that the product is assigned to an open U-shaped disassembly workstation. Constraint (11) ensures that the assignment of disassembly tasks on the U-shaped disassembly line meets the priority relationship constraints. Constraint (12) ensures that the disassembly tasks meet the conflict relationship constraints. Constraint (13) ensures that task j_1 has been completed before task j_2 is executed. Constraint (14) ensures that the working time of each U-shaped workstation in the disassembly plant cannot exceed the cycle time of the disassembly plant.

7) Constraints related to hybrid disassembly lines

$$\sum_{r \in \mathbb{R}} \alpha_{krpi} \le \sum_{w \in \mathbb{W}_k^{Z}} \sum_{j \in \mathbb{J}_p} d_{pij}^{M} x_{pjkw}^{Z} + \sum_{w \in \mathbb{W}_k^{U}} \sum_{j \in \mathbb{J}_p} \sum_{l \in \mathbb{L}} d_{pij}^{M} x_{pjkwl}^{U}, \forall k \in \mathbb{K}, \forall p \in \mathbb{P}, \forall i \in \mathbb{I}_p \tag{15}$$

$$\sum_{k \in \mathbb{K}} z_{pk} = 1, \forall p \in \mathbb{P} \tag{16}$$

$$z_{pk} = 0, \forall k \in \mathbb{K}, \forall p \in \mathbb{P}, \forall n \in \mathbb{N} \text{ and } \gamma_{kn} < \beta_{pn} \tag{17}$$

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

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

$$\sum_{w \in \mathbb{W}_k^{Z}} x_{pjkw}^{Z} + \sum_{w \in \mathbb{W}_k^{U}} \sum_{l \in \mathbb{L}} x_{pjkwl}^{U} \le z_{pk}, \forall p \in \mathbb{P}, \forall k \in \mathbb{K}, \forall j \in \mathbb{J}_p \tag{20}$$

$$\sum_{k \in \mathbb{K}} \left(\sum_{w \in \mathbb{W}_k^{Z}} x_{pjkw}^{Z} + \sum_{w \in \mathbb{W}_k^{U}} \sum_{l \in \mathbb{L}} x_{pjkwl}^{U}\right) \le 1, \forall p \in \mathbb{P}, \forall j \in \mathbb{J}_p \tag{21}$$

$$z_{pk} \in \{0, 1\}, \forall p \in \mathbb{P}, \forall k \in \mathbb{K} \tag{22}$$

$$x_{pj_jkw}^{Z} \in \{0, 1\}, \forall p \in \mathbb{P}, \forall j \in \mathbb{J}_p, \forall w \in \mathbb{W}_k^{Z}, \forall k \in \mathbb{K}, \tag{23}$$

$$x_{pj_jkwl}^{U} \in \{0, 1\}, \forall p \in \mathbb{P}, \forall j \in \mathbb{J}_p, \forall w \in \mathbb{W}_k^{U}, \forall k \in \mathbb{K}, \forall l \in \mathbb{L} \tag{24}$$

$$y_k^{Z} \in \{0, 1\}, \forall k \in \mathbb{K} \tag{25}$$

$$y_k^{U} \in \{0, 1\}, \forall k \in \mathbb{K} \tag{26}$$

$$u_{kw}^{Z} \in \{0, 1\}, \forall k \in \mathbb{K}, \forall w \in \mathbb{W}_k^{Z} \tag{27}$$

$$u_{kw}^{U} \in \{0, 1\}, \forall k \in \mathbb{K}, \forall w \in \mathbb{W}_k^{U} \tag{28}$$

$$T_k^{T} \in \mathbb{R}_+, \forall k \in \mathbb{K} \tag{29}$$

Constraint (15) ensures that the components obtained by disassembling products can only be transported to one manufacturing plant. Constraint (16) ensures that each product can only be assigned to one disassembly plant. Constraint (17) ensures that each product can only be assigned to a disassembly plant that meets the required disassembly technology. Constraint (18) ensures that each disassembly plant can only open one type of disassembly line. Constraint (19) ensures that each product can only be assigned to an open disassembly plant. Constraint (20) ensures that the product disassembly tasks are carried out on open disassembly line workstations. Constraint (21) ensures that each disassembly task of each product is performed at most once. Constraints (22)–(29) specify the range of decision variables.

III. Algorithm Design

In this section, we introduce the algorithm design for solving HMRO based on the TD3 algorithm, including the design of states and actions, the construction of the reward mechanism, and the overall framework and training process of the algorithm. The design of states and actions takes into account the dynamic changes of factories, tasks, workstations, and disassembly lines, while the reward mechanism guides the algorithm to learn better strategies by comparing the current and historical objective function values. The entire algorithm improves training efficiency and solution quality through experience replay and network update mechanisms.

A. Design of States and Actions

1) State

In the TD3 algorithm, the state $S_t = [p, k, w, j_p]$ includes the disassembly product p, disassembly factory k, workstation w, and disassembly task j_p, where $S_t[0] = p$, $S_t[1] = k$, $S_t[2] = w$, and $S_t[3] = j_p$. The state is updated each time a valid disassembly step is performed. The initial state of S_t is $[0, 0, 0, 0]$, indicating that disassembly has not yet started. During disassembly, products are processed sequentially starting from $p = 1$. When one product is completely disassembled, the state switches by updating $S_t[0] = p + 1$.

2) Action

In the TD3 algorithm, the action $A = [k, j_p, w, l]$ is used to schedule product p to disassembly factory k, select the active disassembly line l, choose the disassembly task j_p for the product, and assign the workstation w. Specifically, $A[0] = k$, $A[1] = j_p$, $A[2] = w$, and $A[3] = l$.

When disassembling product p, a set K_c is used to store factories capable of the required disassembly technology for that product. The state $S_t[1]$ is checked; if $S_t[1] = 0$, it indicates that the product has not yet been assigned to a factory. Then, $A[0]$ selects a disassembly factory k from K_c to assign the product, and the state is updated to $S_t[1] = k$. If $S_t[1] \neq 0$, it means the product is already assigned to factory k, so $S_t[1]$ remains unchanged.

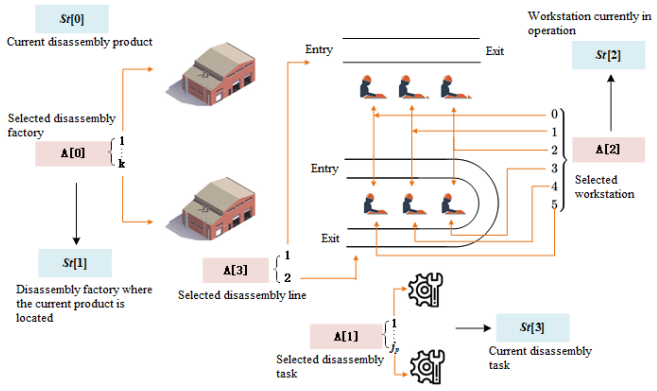

Fig. 2. Example diagram of action and state selection.

Similarly, for disassembly tasks, based on the current task state $S_t[3]$, the subsequent disassembly tasks of product p are determined and stored in the set J_c. Then, $A[1]$ selects the next disassembly task from J_c, and $S_t[3]$ is updated accordingly.

To determine which disassembly line is active in a factory, an array L_k records the activation status: $L_k[k-1] = 0$ indicates that factory k has no active disassembly line; $L_k[k-1] = 1$ indicates a linear disassembly line is active; $L_k[k-1] = 2$ indicates a U-shaped disassembly line is active. If $L_k[k-1] = 0$, the disassembly line activation is determined by $A[3]$. If $L_k[k-1] \neq 0$, the activation remains unchanged.

Finally, action $A[2]$ assigns the disassembly task to a workstation based on the current workstation state $S_t[2]$.

These operations effectively avoid invalid decisions caused by position transitions and Petri net transitions, thereby improving training efficiency. The definitions of actions and states are illustrated in Fig. 2.

B. Reward Design

This work aims to solve a multi-objective optimization problem. Consequently, when designing the reward R_e, both objectives f_1 and f_2 must be considered simultaneously. Objective f_1' denotes the maximum attainable profit under the current state, whereas objective f_2' denotes the maximum cycle time of the plant in that state. Note that the second objective is to minimize the maximum cycle time; hence, the smaller the value of f_2, the better it performs in the computation of R_e.

Because the algorithm relies on reward signals to reflect action preferences, we compare the current values of f_1' and f_2' with those of the previous iteration (denoted as f_1 and f_2) to steer the algorithm toward better solutions. The reward is computed as follows.

$$R_e = \begin{cases} 2, & \text{If } f_1' > f_1 \text{ and } f_2' < f_2. \\ 1, & \text{If } f_1' > f_1 \text{ or } f_2' < f_2. \\ 0, & \text{Otherwise.} \end{cases} \tag{30}$$

Although the reward function is relatively simple, it provides a stable learning signal for guiding the agent toward improving the optimization objectives. More sophisticated reward shaping strategies and ablation studies will be explored in future work to further enhance training efficiency.

C. Algorithm Description

Fig. 3 illustrates the framework for solving the HMRO using the TD3 algorithm. Algorithm 1 presents the training and testing procedures of the TD3 algorithm. During training, the TD3 algorithm randomly samples a batch of data from the replay buffer to obtain the current state S_t, action A, reward Re, and the next state S_{t+1}.

Although TD3 is primarily designed for continuous control problems, it can be applied to discrete decision-making problems through an action mapping mechanism. In this paper, the Actor network outputs a continuous action vector, where each element represents the preference score of the corresponding candidate action.

At each decision epoch, the set of feasible actions is first constructed based on the current system state, which is determined by task precedence constraints, workstation availability,

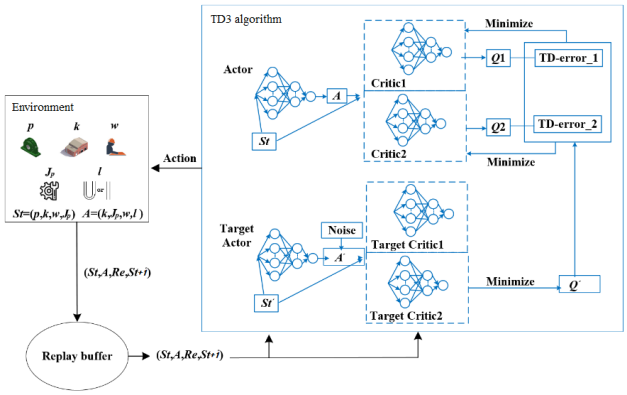

Fig. 3. Framework for solving the HMRO using TD3.

Algorithm 1 TD3 Training and Evaluation Process

Environment *env*, total time steps T, log interval L Cumulative reward, episode cost, evaluation metrics
Initialize the environment *env* and initial state *state*, TD3 model with hyper parameters
Set total training steps T, log interval L

Training Phase:
Train the model: $model.learn(total\_timesteps = T, log\_interval = L)$

Evaluation Phase:
Reset the environment: $obs \leftarrow env.reset()$
Set the done flag: $done \leftarrow False$
$done \neq True$ Select action using the trained policy:
$action, \_ \leftarrow model.predict(obs)$ Execute the action in the environment:
$obs, reward, done, info \leftarrow env.step(action)$ Record reward, cost, and other evaluation metrics
Output cumulative reward, episode cost, and final performance statistics

and system scheduling rules. Subsequently, the continuous output is mapped to a final discrete scheduling decision by selecting the action with the highest score from the set of feasible actions. This process can be expressed as follows: $a^* = \arg\max_{a \in A_{\text{valid}}} \pi_\theta(s)_a$. Where A_{valid} represents the set of feasible actions in the current state.

Furthermore, to ensure the feasibility of the scheduling decisions, this paper adopts an action masking strategy. During the action selection phase, all infeasible actions that violate the system constraints are masked, thereby ensuring that the final executed action always satisfies the system constraints.

IV. Experimental Design and Analysis

All code programs are implemented using PyCharm Community Edition 2022.2.1 and are compiled using Python 3.9.7. The experiments and tests are conducted on a computer equipped with an AMD Ryzen 7 5800H CPU operating at 3.20 GHz and 16 GB RAM. This hardware setup ensures that experimental conditions remain consistent.

TABLE I 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 II Disassembly technology information

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

A. Case Design

In this work, washing machines, radios, lithium-ion batteries, and roller bearing housings are selected as disassembly products for experimental evaluation. As shown in Table I, six experimental cases are designed. Table II presents the disassembly technologies required by each product and the disassembly technologies available in each disassembly factory in the six cases. Each product is represented by the first letter of its name in the table. A value of 1 in brackets [ ] indicates the use of a specific disassembly technology, with its position corresponding to the technology number.

For example, in case 2, there are two disassembly factories and two remanufacturing factories. Three products are to be disassembled: washing machine, bearing seat, and radio. Disassembly factory 1 is equipped with disassembly technologies 1, 2, 3, and 5, while Disassembly factory 2 has Technologies 1, 4, and 5. The washing machine requires Technologies 1 and 2, the bearing housing unit requires Technologies 2 and 4, and the radio requires Technologies 2 and 5.

B. Experimental Design

To evaluate the performance of the TD3 algorithm in solving the HMRO, we compare TD3 with DDPG, SAC, and A2C algorithms. The experimental setup is based on the open-source framework provided by Stable-Baselines3. All algorithms share the same initialization settings, with a learning rate of 1×10^{-5} and a batch size of 100. Each algorithm is

executed in five independent trials, with 1000 iterations per trial.

To validate the superiority of TD3 in addressing HMRO, it is implemented and trained alongside DDPG, A2C, and SAC using the Stable-Baselines3 framework. The experiments adopt the default parameter configurations provided by the framework.

The evaluation metrics used in this work include Spread, Epsilon, GD$^+$, and IGD. Spread measures the diversity of the obtained solutions, Epsilon reflects the convergence of the solutions, and GD$^+$ and IGD represent the overall performance of the algorithms.

C. Experimental Analysis

First, the Pareto front is obtained from the solution set generated by each algorithm. Then, the Pareto fronts from all algorithms are merged to form a combined front, which serves as the reference true Pareto front.

The evaluation results for each algorithm under each test case, including the values of Spread, Epsilon, GD$^+$, and IGD, are shown in Table III, Table IV, Table V, and Table VI, respectively.

In terms of the Spread indicator, TD3 performs well in cases 1, 2, and 4 with the lowest Spread values. DDPG performs poorly across all cases. SAC shows good performance in cases 3 and 6, but performs poorly in cases 2 and 5. A2C performs poorly in case 6 but well in the other cases.

TABLE III Values of the spread indicator

Case Number	TD3	DDPG	SAC	A2C
1	1.0	5.5	23.5	28.5
2	17.5	64.5	123.5	20.5
3	54.0	40.0	7.5	39.5
4	165.0	191.0	188.5	199.0
5	494.5	360.5	543.0	333.5
6	393.5	395.5	192.0	495.5

TABLE IV Values of the epsilon indicator

Case Number	TD3	DDPG	SAC	A2C
1	0.0000	10.0000	58.0000	64.0702
2	0.0000	41.9761	239.0000	43.0000
3	0.0000	138.0036	27.1661	30.8058
4	0.0000	94.1912	105.0428	187.3232
5	0.0000	216.9469	271.0000	271.3705
6	0.0000	1126.44263	185.6475	547.3664

TABLE V Values of the GD$^+$ indicator

Case Number	TD3	DDPG	SAC	A2C
1	0.0000	7.1063	41.8867	45.7438
2	0.0000	97.8493	125.1918	23.8379
3	0.0000	35.8608	26.2583	20.3899
4	0.0000	55.1588	58.1957	113.6276
5	0.0000	141.0093	165.8372	181.7195
6	0.0000	928.6074	132.8674	384.7588

For the Epsilon indicator, TD3 achieves a value of 0 in all cases, representing the best performance. DDPG performs well in cases 1, 4, and 5, but poorly in cases 3 and 6. SAC performs well in case 6 but poorly in case 2. A2C shows good performance in cases 2 and 3, but performs poorly in cases 1, 4, and 5.

TABLE VI Values of the IGD indicator

Case Number	TD3	DDPG	SAC	A2C
1	0.0000	1.0	12.6885	9.3094
2	0.0000	29.9081	20.2361	30.2241
3	0.0000	47.5674	43.2357	27.6767
4	2.0816	45.9836	47.0053	51.2249
5	25.8779	320.3609	183.6368	413.9824
6	0.0000	1027.1109	412.435	465.9883

Regarding the GD$^+$ indicator, TD3 consistently achieves a value of 0 in all cases, indicating optimal performance. DDPG performs well in cases 1 and 5, but poorly in cases 2 and 6. SAC performs well in case 6 but poorly in cases 1 and 2. A2C shows good performance in cases 2 and 3, but performs poorly in cases 4 and 5.

As for the IGD indicator, TD3 demonstrates the best performance in all cases. DDPG performs well in case 1, but poorly in cases 5 and 6. SAC shows good results in case 3, but performs poorly in cases 1 and 3. A2C performs well in cases 1 and 2, but poorly in cases 2, 4, and 5.

Overall, TD3 outperforms the other three algorithms across most indicators and cases, particularly showing stable and low values in the Epsilon, GD$^+$, and IGD indicators. While DDPG and SAC show good performance on certain metrics, they are unstable and perform poorly on others. A2C generally performs worse across most metrics and cases. Therefore, TD3 is superior in solving the HMRO, offering better stability and performance.

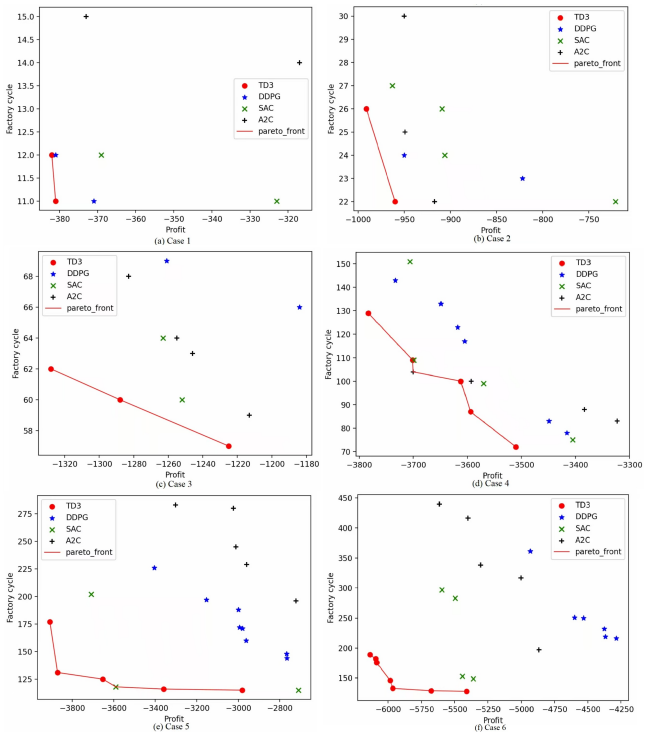

Fig. 4. Pareto solutions.

Fig. 4 shows the Pareto solutions obtained by the four algorithms in six different cases. The Pareto front represents

the set of non-dominated solutions in a multi-objective optimization problem, where no objective can be improved without degrading another.

Since the problem focuses on minimizing objectives, factory profit values are converted to negative numbers so that maximizing profit corresponds to minimizing the objective. As shown in the figures, the TD3 algorithm performs well in all cases, producing smooth Pareto fronts that span a wide range of factory cycle times and profit values. The DDPG algorithm performs poorly in all cases, with solutions concentrated in regions of high factory cycle time and low profit. The A2C algorithm shows moderate performance, with solutions distributed in mid-range factory cycles and profits. The SAC algorithm performs well, with solutions concentrated in areas of lower factory cycle time and higher profit.

Therefore, TD3 demonstrates superior performance in solving the multi-objective optimization problem, as it can identify better Pareto-optimal solutions.Comparisons with evolutionary multi-objective algorithms such as NSGA-II will be investigated in future work.

D. Computational Complexity and Runtime Analysis

During the training phase, the main computational overhead of each reinforcement learning algorithm stems from the forward and backward propagation processes of the neural network, with its overall complexity being primarily related to the scale of the network parameters and the number of training iterations. Building upon DDPG, TD3 introduces a double Q-network and a delayed policy update mechanism, which requires an additional critic network to be updated during training, resulting in a slightly higher training cost than DDPG. However, this design effectively mitigates the overestimation of Q-values, thereby enhancing policy stability and the quality of the final solution.

In comparison, SAC generally has a higher training complexity than TD3 due to the need to simultaneously learn a policy network, two Q-networks, and a temperature parameter. A2C, with its relatively simple network structure, has a lower training cost but exhibits relatively limited optimization capabilities in complex scheduling problems.

In the inference phase, all algorithms only need to perform a single forward pass of the neural network to generate a scheduling decision, resulting in a small computational overhead that can meet the real-time decision-making requirements of practical scheduling systems.

V. Conclusion

This work investigates the Hybrid-line Multi-type Factory Remanufacturing Optimization Problem (HMRO), considering the disassembly technologies used in disassembly factories and the selection of different disassembly lines. A multi-objective hybrid-line multi-type factory remanufacturing model is developed with the objectives of maximizing profit and minimizing factory cycle time. The HMRO problem involves assigning products to different factories that employ distinct disassembly technologies. While considering constraints such as precedence relationships and task conflicts, this study schedules tasks on optimal disassembly lines to ensure scalability and high performance in complex and large-scale scheduling environments. To address this problem, the TD3 algorithm is adopted and compared with several baseline reinforcement learning algorithms. Experimental results demonstrate that TD3 achieves superior performance in solving the HMRO problem.

In future work, we will consider optimization in the following directions. First, multi-type factory remanufacturing requires more strategies targeting inter-factory collaboration and resource optimization. Second, research on disassembly line balancing requires more models and algorithms to demonstrate better efficiency. Regarding reinforcement learning, more complex algorithms can be explored to address problems in large-scale and multi-modal environments, and deep learning methods can be integrated to improve learning efficiency and generalization capabilities.

References

[1] D. Singhal, S. Tripathy *et al.*, "Remanufacturing for the circular economy: Study and evaluation of critical factors," *Resources, Conservation and Recycling*, vol. 156, p. 104681, 2020.

[2] A. Haleem, M. Javaid *et al.*, "A pervasive study on green manufacturing towards attaining sustainability," *Green Technologies and Sustainability*, vol. 1, no. 2, p. 100018, 2023.

[3] M. Baballe, M. Yusif *et al.*, "Advantages and challenges of remanufactured products," *Acta Energetica*, pp. 1–7, Jan. 2023.

[4] R. Fofou, Z. Jiang *et al.*, "A review on the lifecycle strategies enhancing remanufacturing," *Applied Sciences*, vol. 11, no. 13, p. 5937, 2021.

[5] X. Zhang, Y. Tang *et al.*, "Remanufacturability evaluation of end-of-life products considering technology, economy and environment: A review," *Science of The Total Environment*, vol. 764, p. 142922, 2021.

[6] X. Niu, X. Guo *et al.*, "Hybrid disassembly line balancing of multi-factory remanufacturing process considering workers with government benefits," *Mathematics*, vol. 13, no. 5, p. 880, 2025.

[7] L. Qi, Q. Zeng *et al.*, "Twin delayed deep deterministic policy gradient algorithm for a heterogeneous multifactory remanufacturing optimization problem," *IEEE Transactions on Computational Social Systems*, 2025.

[8] Q. Zeng, X. Guo *et al.*, "Optimization of product remanufacturing process across multifactories with reinforcement learning," in *Proc. 2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2024, pp. 1–6.

[9] J. Wang, M. Zhou *et al.*, "Multiperiod asset allocation considering dynamic loss aversion behavior of investors," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 1, pp. 73–81, 2019.

[10] K. Zhang, R. Zhou *et al.*, "Transmission line component defect detection based on UAV patrol images: A self-supervised HC-ViT method," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 54, no. 11, pp. 6510–6521, 2024.

[11] Z. Zhang, X. Guo *et al.*, "Multi-objective discrete grey wolf optimizer for solving stochastic multi-objective disassembly sequencing and line balancing problem," in *Proc. 2020 IEEE Int. Conf. Systems, Man, and Cybernetics (SMC)*, 2020, pp. 682–687.

[12] S. Parsa and M. Saadat, "Human-robot collaboration disassembly planning for end-of-life product disassembly process," *Robotics and Computer-Integrated Manufacturing*, vol. 71, p. 102170, 2021.

[13] M. Lee, X. Liang *et al.*, "A review of prospects and opportunities in disassembly with human–robot collaboration," *Journal of Manufacturing Science and Engineering*, vol. 146, no. 2, 2024.

[14] X. Guo, F. Guo *et al.*, "Modeling and optimization of multiproduct human–robot collaborative hybrid disassembly line balancing with resource sharing," *IEEE Transactions on Computational Social Systems*, vol. 12, no. 5, pp. 2848–2863, 2025.

[15] X. Guo, L. Chen *et al.*, "Multifactory disassembly process optimization considering worker posture," *IEEE Transactions on Computational Social Systems*, vol. 12, no. 5, pp. 3049–3061, 2025.

[16] T. Tolio, A. Bernard *et al.*, "Design, management and control of demanufacturing and remanufacturing systems," *CIRP Annals*, vol. 66, no. 2, pp. 585–609, 2017.
[17] C. Mejía-Moncayo, J. Kenné *et al.*, "On the development of a smart architecture for a sustainable manufacturing-remanufacturing system: a literature review approach," *Computers & Industrial Engineering*, vol. 180, p. 109282, 2023.
[18] C. Ke, Y. Chen *et al.*, "An integrated design method for used product remanufacturing process based on multi-objective optimization model," *Processes*, vol. 12, no. 3, p. 518, 2024.
[19] X. Wang, M. Zhou *et al.*, "A branch and price algorithm for crane assignment and scheduling in slab yard," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 1122–1133, 2021.
[20] S. Qin, S. Zhang *et al.*, "Multiobjective multiverse optimizer for multirobotic U-shaped disassembly line balancing problems," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 2, pp. 882–894, 2024.
[21] L. Zhou, X. Guo *et al.*, "Multifactory remanufacturing process optimization considering worker scheduling," *IEEE Transactions on Computational Social Systems*, 2025.
[22] M. Han, L. Yun *et al.*, "Deep reinforcement learning-based approach for dynamic disassembly scheduling of end-of-life products with stimuli-activated self-disassembly," *Journal of Cleaner Production*, vol. 423, p. 138758, 2023.
[23] F. Ming, W. Gong *et al.*, "Constrained multi-objective optimization with deep reinforcement learning assisted operator selection," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 4, pp. 919–931, 2024.
[24] S. Qin, Y. Feng *et al.*, "Optimization of circular disassembly lines with human-assisted robotic workstations using two-stage greedy PPO algorithm," *IEEE Transactions on Computational Social Systems*, pp. 1–13, 2025.
[25] G. Ferrer and D. Whybark, "From garbage to goods: Successful remanufacturing systems and skills," *Business Horizons*, vol. 43, no. 6, pp. 55–55, 2000.
[26] J. Guo and G. Ya, "Optimal strategies for manufacturing/remanufacturing system with the consideration of recycled products," *Computers & Industrial Engineering*, vol. 89, pp. 226–234, 2015.
[27] X. Zhang, X. Ao *et al.*, "A sustainability evaluation method integrating the energy, economic and environment in remanufacturing systems," *Journal of Cleaner Production*, vol. 239, p. 118100, 2019.
[28] Y. Feng, X. Xia *et al.*, "Multi-objective optimization of recycling and remanufacturing supply chain logistics network with scalable facility under uncertainty," *Production & Manufacturing Research*, vol. 10, no. 1, pp. 641–665, 2022.
[29] P. Li, D. Chen *et al.*, "Path planning of mobile robot based on improved TD3 algorithm in dynamic environment," *Heliyon*, vol. 10, no. 11, 2024.
[30] B. Sun, M. Song *et al.*, "Multi-objective solution of optimal power flow based on TD3 deep reinforcement learning algorithm," *Sustainable Energy, Grids and Networks*, vol. 34, p. 101054, 2023.
[31] E. Sumiea, S. Abdulkadir *et al.*, "Deep deterministic policy gradient algorithm: A systematic review," *Heliyon*, 2024.
[32] Z. Bi, X. Guo *et al.*, "Deep reinforcement learning for truck-drone delivery problem," *Drones*, vol. 7, no. 7, p. 445, 2023.
[33] H. Shi, G. Liu *et al.*, "MARL sim2real transfer: Merging physical reality with digital virtuality in metaverse," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 4, pp. 2107–2117, 2023.
[34] J. Gu, J. Wang *et al.*, "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.
[35] J. Wang, G. Xi *et al.*, "Reinforcement learning for hybrid disassembly line balancing problems," *Neurocomputing*, vol. 569, p. 127145, 2024.
[36] Y. Ren, X. Lu *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.
[37] M. Kerin and D. Pham, "Smart remanufacturing: a review and research framework," *Journal of Manufacturing Technology Management*, vol. 31, no. 6, pp. 1205–1235, 2020.
[38] Y. Jing, W. Li *et al.*, "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.
[39] J. Mao, D. Hong *et al.*, "Disassembly sequence planning of waste auto parts," *Journal of the Air & Waste Management Association*, vol. 71, no. 5, pp. 607–619, 2021.
[40] L. Zhou, X. Guo *et al.*, "Distributed factory disassembly scheduling problem," in *Proc. 2023 International Conference on Cyber-Physical Social Intelligence (ICCSI)*, 2023, pp. 429–434.
[41] K. Sycara, S. Roth *et al.*, "Resource allocation in distributed factory scheduling," *IEEE Expert*, vol. 6, no. 1, pp. 29–40, 1991.
[42] N. Karimi and H. Davoudpour, "A knowledge-based approach for multi-factory production systems," *Computers & Operations Research*, vol. 77, pp. 72–85, 2017.

Jinlei Gu received his B.S. degree in computer science from Changshu Institute of Technology, China, in 2021, M.S. degree in computer science, from Monmouth University, New Jersey, U.S., in 2023.

He is currently a Ph.D student in the New Jersey Institute of Technology. His research interests include reinforcement learning and intelligent optimization algorithms.

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.

Vladislav D. Veksler Vladislav D. Veksler received his Ph.D. degree from the Rensselaer Polytechnic Institute, Troy, NY, in 2009. Currently, he is an Assistant Professor of Computer Science with Caldwell University, Caldwell, NJ, where he also serves as the Director of C-STEM Laboratories and the Director of the CogAI Research Laboratory. He has authored more than 50 technical papers in journals and conference proceedings. His research interests include artificial intelligence, pedagogical methods in computer science, cognitive and behavioral modeling, and human-computer interaction.