Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Multi-objective reinforcement learning for dynamic balancing of two-sided human–robot collaborative disassembly lines

Jinlong Wang [a], Min Li [a], Fanyun Meng [a], Haoran Zhao [a,*], Xin Sun [b,c]

[a] School of Information and Control Engineering of Qingdao University of Technology, 777 Jialingjiang East Road, Huangdao District, Qingdao, China
[b] Faculty of Data Science, City University of Macau, Macau, China
[c] School of Data Science, Chinese University of Hong Kong, Shenzhen, China

ARTICLE INFO

Keywords:
Disassembly line balancing problem
Human–robot collaboration
Disassembly uncertainty
Multi-objective reinforcement learning

ABSTRACT

With the deepening of resource recycling and sustainable manufacturing concepts, the efficient and safe disassembly of end-of-life (EOL) large-size products has become a critical component of green manufacturing systems. However, current research seldom addresses disassembly line layout design tailored for large-size products, and studies on managing disassembly uncertainties remain limited. To bridge these gaps, this study proposes a two-sided human–robot collaborative disassembly line system that improves space utilization and disassembly efficiency through coordinated bilateral operations. A task classification mechanism allocates tasks to robots or human workers based on hazard levels and demand priorities. A human intervention mechanism is incorporated to handle robot-side failures, such as those caused by rusted fasteners, ensuring process continuity and accounting for additional labor costs. A multi-objective proximal policy optimization (MO-PPO) algorithm is developed to enhance decision-making. By randomly sampling preferences during training, the agent learns a unified policy model that generalizes across different preferences and approximates the pareto front. The proposed approach addresses the static nature of traditional heuristic methods and the fixed-weight limitations of weighted reinforcement learning. To evaluate its efficacy, experiments were conducted on a disassembly line for large-sized refrigerators, comparing the proposed MO-PPO algorithm with other multi-objective reinforcement learning algorithms and classical heuristic methods. The results indicate that the MO-PPO algorithm consistently achieves near-optimal solutions for large-scale disassembly tasks. Under conditions of disassembly uncertainty, the MO-PPO algorithm completes its execution in less than 1 s, whereas traditional heuristic algorithms require re-execution, with total run times exceeding 30 s.

1. Introduction

In recent decades, rapid economic and technological growth, along with the rise of mass production and consumption (Caterino et al., 2025), has led to a surge in EOL products, presenting significant challenges for their management. These products often contain hazardous substances that, if mishandled, pose serious risks to environmental and human health (Kerin & Pham, 2020). Simultaneously, EOL products are a valuable source of recoverable materials, and their efficient recycling enables resource circulation and supports the development of the circular economy (Bai, Zhou, & Sarkis, 2023; El Jaouhari, Samadhiya, Benbrahim, Kumar, & Luthra, 2025). In response, many countries have implemented policies that leverage regulatory incentives and technological innovation to promote sustainable EOL product recovery and reuse, aiming to balance economic gains with environmental protection (Liu, Wang, & Luo, 2025; Vahedi-Nouri, Rohaninejad, Hanzálek, & Foumani, 2025).

As a critical intermediary stage in the recycling and remanufacturing of EOL products, disassembly plays a pivotal role in bridging upstream collection and downstream material recovery. However, efficiently disassembling large volumes of discarded products remains a major challenge for recycling enterprises. Existing research on EOL product disassembly primarily focuses on two key problems: disassembly line balancing problem (DLBP) and disassembly scheduling problem (Ilgin, Gupta, & Battaïa, 2015; Rosenberg, Huster, Rudi, & Schultmann, 2025). The DLBP aims to develop optimized task allocation schemes that meet specific constraints while aligning with the operational needs of disassembly facilities, thereby ensuring the

* Corresponding author.
E-mail addresses: wangjinlong@qut.edu.cn (J. Wang), limin@stu.qut.edu.cn (M. Li), mengfanyun@qut.edu.cn (F. Meng), zhaohaoran@qut.edu.cn (H. Zhao), sunxin1984@ieee.org (X. Sun).

https://doi.org/10.1016/j.cie.2025.111703
Received 27 June 2025; Received in revised form 11 September 2025; Accepted 23 November 2025
Available online 24 November 2025

efficient distribution of tasks across workstations (Zhu, Zhang, & Guan, 2020). Disassembly scheduling addresses the sequencing of disassembly tasks based on incoming product orders (Zhou, Xu, Chen, Liao, & Xu, 2025). Compared to scheduling, DLBP represents a mid-term decision-making problem that directly influences line layout and has a substantial impact on sequence planning and overall scheduling performance.

The DLBP, as a core issue in disassembly optimization, has been extensively studied in terms of line configurations – such as straight, U-shaped, parallel, and bilateral layouts – and has evolved from manual operations to robotic disassembly for improved flexibility and safety. In terms of solution methods, research has progressed from exact and heuristic approaches to metaheuristics and reinforcement learning, with multi-objective optimization emerging as a key trend.

Despite significant progress, existing research on the DLBP reveals several critical limitations. Traditional automated disassembly systems often struggle with poor adaptability and instability when handling complex tasks. In contrast, purely manual disassembly lines face challenges such as low efficiency and safety risks. This study introduces a dual-station human–robot collaborative disassembly line. In this design, manual and robotic workstations perform disassembly tasks simultaneously. It combines the efficiency of robots with the flexibility of human intervention. This approach optimizes the disassembly line layout, reduces its length, and significantly improves efficiency, safety, and space utilization.

Existing studies often fail to adequately address the inherent uncertainties in the disassembly process, such as task failures caused by rusted bolts or variations in task durations due to component deformation. These uncertainties can significantly impact the stability and throughput of the system, thereby reducing overall disassembly efficiency. However, current research and methods generally lack robust strategies to handle these uncertainties, resulting in unstable system performance and difficulty maintaining high operational efficiency when faced with real-world variations. To address this issue, this study proposes a human–robot collaboration intervention mechanism. When robots are unable to complete tasks autonomously, human operators can step in to ensure the continuity of disassembly operations. This mechanism effectively mitigates task failures caused by uncertainties such as rusted bolts or deformed components, ensuring the system's stability and efficiency in uncertain environments.

Traditional optimization methods typically lack sufficient adaptability when responding to dynamic changes. When failures or disturbances occur, heuristic algorithms often require reinitialization, resulting in delayed responses and an inability to quickly adjust the system to handle sudden situations. Moreover, reinforcement learning methods based on weighted sums have limitations in multi-objective optimization, making it difficult to effectively balance conflicts between different objectives. In disassembly tasks, frequent changes in tasks and environmental conditions make decision requirements complex and variable. Existing methods often struggle to adapt to these changes in real-time, failing to provide flexible decision support, which affects the system's efficiency and stability. To address this issue, this study proposes a preference-based MO-PPO approach, which can flexibly adjust optimization strategies in dynamic, multi-objective environments, enabling efficient, real-time decision support.

This study aims to design an efficient human–robot collaborative bilateral disassembly line tailored to large-scale EOL product scenarios, balancing space utilization, safety, and economic performance. It further proposes an intelligent disassembly line balancing optimization method capable of adapting to dynamic multi-objective requirements. Addressing key limitations in existing research, including unsuitable layouts for large-size products, outdated manual disassembly practices, and inadequate handling of process uncertainties, this work introduces systematic innovations across three dimensions: line layout, task allocation, and dynamic balancing strategies. The main contributions of this study are as follows:

1 For the layout design, a two-sided configuration is adopted, facilitating simultaneous operations on both sides of the disassembly line. This configuration not only reduces the overall line length but also enhances spatial efficiency and throughput. To address the practical uncertainties encountered in disassembly processes, such as task failures caused by corroded fasteners, a human-intervention fallback mechanism is incorporated. In instances where robots are unable to complete tasks autonomously, human operators step in to provide necessary assistance, ensuring seamless operations without interruptions. The labor costs associated with human intervention are explicitly factored into the objective function, strengthening the model's industrial relevance and operational resilience. This adaptive mechanism ensures that the system remains stable and efficient, even under uncertain conditions, thereby promoting continuous operations and optimized resource utilization.
2 In the task allocation process, disassembly tasks are categorized into three distinct types: hazardous, high-demand, and general. Hazardous tasks are assigned to robotic workstations to ensure operational safety, while high-demand components are allocated to human operators to maximize economic returns. The remaining tasks are dynamically assigned based on the availability of resources. This classification system not only enhances safety by appropriately distributing risky tasks but also ensures efficient task execution by matching resource capabilities with the specific demands of each task. As a result, the allocation strategy optimizes resource use, aligning task complexity with available capabilities and industrial constraints.
3 Unlike previous approaches that primarily rely on heuristic methods or single-weighted reinforcement learning, this study introduces a preference-based MO-PPO framework. This novel framework enables the training of a unified policy that adapts to various preference vectors, allowing for real-time decision-making and the flexible balancing of conflicting objectives. In contrast to heuristic methods, which often require complete reinitialization, and weighted-sum reinforcement learning approaches, which suffer from limited flexibility, the proposed method significantly enhances adaptability and decision-making efficiency. By accommodating multiple objectives and operating under uncertain conditions, the MO-PPO framework provides a more dynamic and responsive solution to disassembly line balancing.

The remainder of this study is structured as follows. Section 2 presents related research on the DLBP. Section 3 defines the proposed two-sided human–robot collaborative disassembly line dynamic balancing problem (THCDLDB). Section 4 presents the enhanced MO-PPO algorithm in detail. Section 5 applies the proposed method to a real-world case study. Finally, Section 6 concludes the paper and outlines directions for future research.

2. Related work

This chapter provides a systematic review from three dimensions: disassembly line layout, disassembly methods, and DLBP solution approaches. It focuses on core literature from 2013 onwards, with particular emphasis on the past three years, especially since 2022. Through a comparative analysis of disassembly line layouts, optimization objectives, and solution methods, this chapter reveals the evolution of the field and identifies research gaps. The relevant content is summarized in Table 1.

2.1. Disassembly line layout

As a critical optimization task in the disassembly process, the DLBP has attracted significant attention since its introduction by Güngör and Gupta (Özceylan, Kalayci, Güngör, & Gupta, 2019), with growing

Table 1
Literature analysis of DLBP.

Studies	Line layout				Objective function						Method			
	SL	UL	TL	OL	CT	NW	SI	DC	DP	EC	EM	HM	MHM	RL
Paksoy, Güngör, Özceylan, and Hancilar (2013)	✓				✓	✓	✓				✓		✓	
Hezer and Kara (2015)				✓		✓					✓			
Bentaha, Battaïa, and Dolgui (2015)	✓							✓			✓			
Li, Chen, Zhu, Yang, and Chu (2019)				✓	✓						✓			
Cevikcan, Aslan, and Yeni (2020)				✓		✓					✓			
Dong et al. (2021)	✓								✓	✓			✓	
Dalle Mura, Pistolesi, Dini, and Lazzerini (2021)	✓					✓	✓		✓				✓	
Yılmaz et al. (2022)				✓	✓		✓	✓					✓	
Guo et al. (2022)				✓					✓				✓	
Liang, Zhang, Yin, Zhang, and Wu (2023)				✓		✓	✓						✓	
Wang et al. (2023)	✓					✓	✓						✓	
Chu and Chen (2024)				✓				✓			✓		✓	✓
Kalaycilar, Azizoğlu, and Batun (2024)	✓								✓		✓	✓		
Yeni, Cevikcan, Yazici, and Yilmaz (2024)				✓		✓					✓	✓		
Zhu, Chen, and Mumtaz (2024)				✓		✓	✓						✓	
Huang and Zhou (2025)				✓	✓					✓			✓	✓
Öksüz, Yılmaz, Öksüz, and Gürsoy Yılmaz (2025)				✓		✓					✓			
This study			✓			✓		✓		✓				✓

SL: straight disassembly line; UL: U-shaped disassembly line; TL: Two-sided disassembly line; OL: other disassembly line; CT: circle time; NW: number of workstations; SI: smoothness index; DC: disassembly cost; DP: disassembly profit; EC: Disassembly energy consumption; EM: exact method; HM: heuristic; MHM: meta-heuristic method; RL: reinforcement learning method.

research interest in recent years (Agrawal & Tiwari, 2008; Aydemir-Karadag & Turkbey, 2013; Dong et al., 2021; Guo et al., 2022; Riggs, Battaïa, & Hu, 2015). Existing studies on DLBP primarily focus on several disassembly line layouts, including straight, parallel, two-sided, and U-shaped configurations. Gungor and Gupta (2001) initially proposed the straight-line layout, which was also adopted by Wang et al. (2023). In subsequent work, Wang, Li, Guo, Gao, and Li (2024) explored a U-shaped workstation layout to address DLBP. To mitigate capacity constraints in the disassembly of high-demand components, Ketzenberg, Souza, and Guide Jr (2003) introduced parallel disassembly lines. Additionally, to reduce unnecessary worker movement and enhance efficiency, Wang, Li, and Gao (2019) examined the two-sided layout under high-volume disassembly scenarios. Each of these configurations presents distinct advantages, and selecting an appropriate layout is essential. Given the focus of this study on large-scale EOL products, a two-sided disassembly line is adopted to improve efficiency and minimize line length.

2.2. Disassembly method

Traditional disassembly lines primarily rely on manual operations (Guo et al., 2024), typically utilizing single-worker workstations, which results in low efficiency and high costs. To address this issue, Cevikcan et al. and Yılmaz et al. proposed disassembly line layouts with multi-manned workstations, incorporating worker skill heterogeneity. This approach effectively overcomes the limitations of single-worker workstations and significantly improves disassembly efficiency. Recently, some studies have explored fully automated robotic disassembly systems (Hu et al., 2024). However, existing robotic technologies still face limitations (Dalle Mura et al., 2021; Lee et al., 2024; Qin et al., 2024) in handling common uncertainties during the disassembly process, particularly when confronted with unexpected situations. When disassembly fails, it directly impacts the operation of the disassembly line, reducing overall efficiency. To address this, several studies have proposed human–robot collaborative disassembly approaches (Qu, Li, Zhang, Liu, & Bao, 2024; Wang, Qiao, Guan, Liu, Ding, et al., 2024; Wu, Zhang, Guo, et al., 2024; Wu, Zhang, Zeng, & Zhang, 2024; Wu, Zhang, Zeng, Zhang, Guo, &, Liu, 2024; Yin, Zhang, Liang, Zeng, & Zhang, 2023). Based on this, we propose assigning hazardous tasks to robots while delegating demand-driven tasks to human workers. This strategy not only ensures worker safety but also enhances economic efficiency. Therefore, this study classifies disassembly tasks based on their attributes and focuses on developing a human–robot collaborative disassembly strategy.

2.3. Optimization method

The DLBP is inherently an NP-hard optimization problem (McGovern & Gupta, 2007). As the problem size increases, its computational complexity grows rapidly. The introduction of two-sided disassembly lines further complicates the problem, as it adds constraints related to the simultaneous operation of tasks on both sides (Xu & Han, 2024). This study addresses a DLBP variant that not only involves the combinatorial optimization challenge but also integrates multi-objective optimization and dynamic environments, significantly increasing the computational difficulty. Over the years, solution approaches to DLBP have evolved considerably. Early research focused primarily on single-objective optimization, while more recent efforts have shifted towards multi-objective formulations. Traditional exact methods (Li et al., 2019; Paksoy et al., 2013; Yin, Zhang, Zhang, Wu, & Liang, 2022; Zheng, He, Chu, & Liu, 2018), heuristic algorithms (Mete, Cil, Ağpak, Özceylan, & Dolgui, 2016), and metaheuristic algorithms (Dalle Mura et al., 2021; Dong et al., 2021) have been widely used. Both Cella, Robin, Faroni, Zanchettin, and Rocco (2025)s upper-layoutlower-scheduling bilevel framework and Zhi and Lien (2025)s single-layer, multi-objective approach that simultaneously optimizes layout and multi-agent path planning represent promising avenues for future research. More recently, the application of reinforcement learning has gained attention as a promising approach, reflecting the increasing complexity and scale of practical problems (Chand & Ravi, 2024). Reinforcement learning, through the use of trained models, can effectively reduce the overhead of repetitive calculations, demonstrating its unique strengths in solving multi-objective optimization problems. This shift towards reinforcement learning highlights its potential to handle the growing complexity of modern DLBP challenges.

Despite progress in evaluation metrics and solution techniques, two major challenges remain in DLBP: real-time rebalancing in dynamic environments and flexible adaptation to multi-objective preferences. Traditional heuristic or metaheuristic algorithms perform well in multi-objective optimization. However, when task parameters are disturbed, these algorithms require reinitialization and recalculation. This leads to decision delays and makes it difficult to adjust production lines in real time.

Regarding preference adaptation, current reinforcement learning methods use fixed-weight strategies. These methods convert multi-objective problems into a single scalar reward. While they enable rapid responses through policy network generalization, they are limited by the strategy-weight relationship. If preferences change, the model

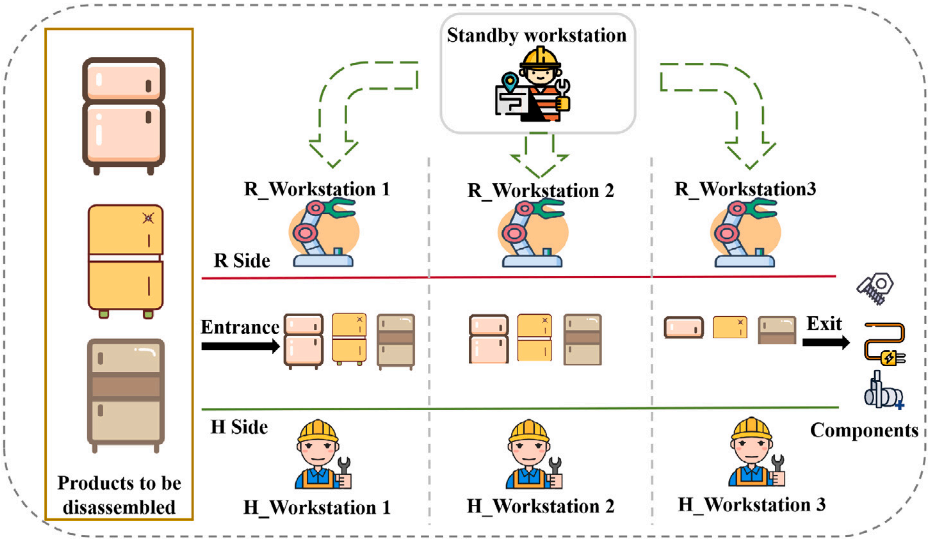


Fig. 1. Disassembly line layout.

must be retrained, which incurs high computational costs and makes it difficult to provide diverse solutions along the Pareto frontier in one run.

Few studies have applied Multi-Objective Reinforcement Learning (MORL) to DLBP. To our knowledge, no research specifically explores MORL's application in disassembly line balancing. This paper aims to fill this gap by proposing an MORL-based DLBP framework that addresses both dynamic responsiveness and flexible multi-objective preference balancing.

3. Problem and mathematical model

3.1. Problem description

This study investigates a two-sided human–robot collaborative disassembly line designed for large-scale EOL products, as depicted in Fig. 1. Disassembly workstations are arranged on both sides of the line, with robotic stations located on the R (robot) side and manual stations on the H (human) side. The system enables synchronous collaboration between robots and human operators, allowing parallel execution of different disassembly tasks. To enhance adaptability under real-world uncertainties, such as task failures on the robot side due to issues like fastener corrosion, a human-assisted fallback mechanism is introduced. Specifically, a flexible manual backup station is deployed to intervene when robotic execution is likely to fail, thereby improving the robustness and reliability of the system (see Fig. 1).

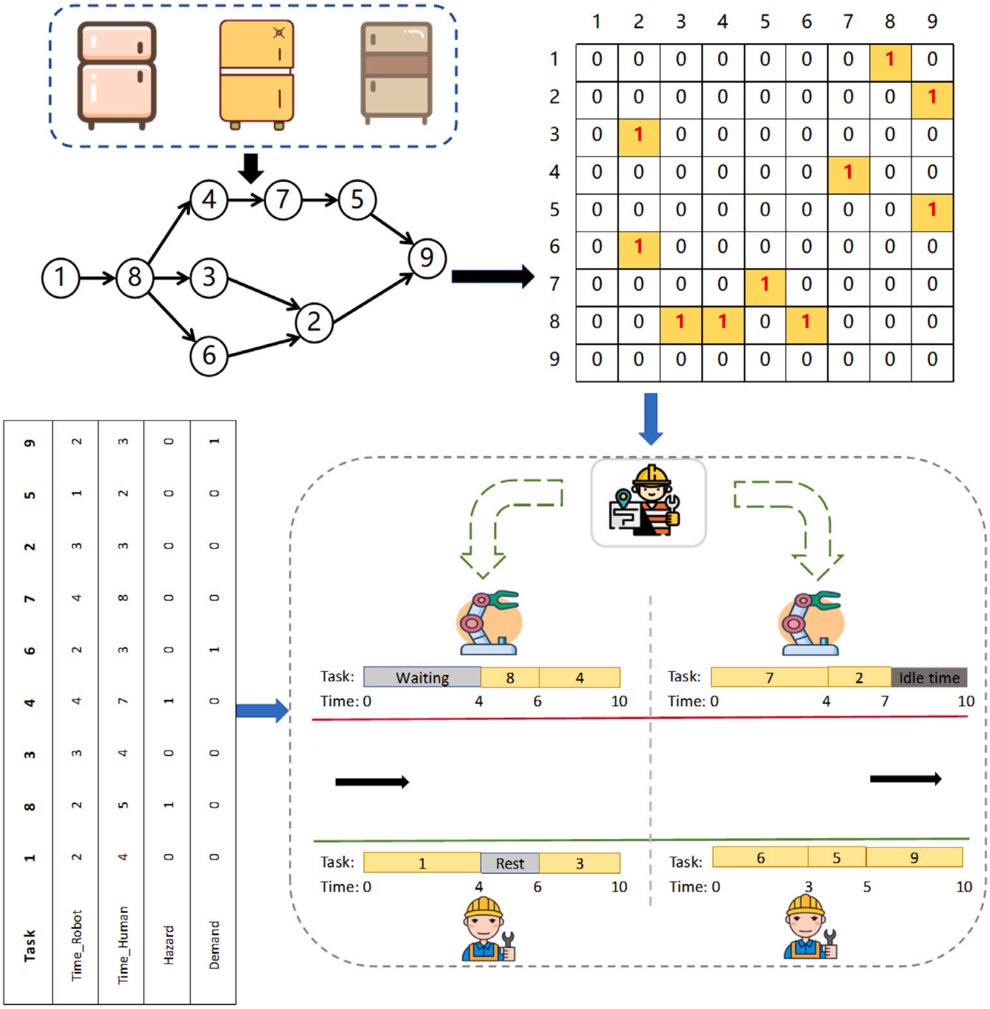
Based on component attributes and operational considerations, disassembly tasks are classified into three categories. Hazardous tasks are assigned to robotic stations to ensure operational safety. Demand-specific tasks, which involve high-value recovery or require specialized handling, are allocated to manual stations to maximize efficiency and resource utilization. General tasks, which lack specific constraints, can be flexibly executed by either station type depending on availability and system conditions.

To accurately model the temporal constraints between disassembly tasks, we construct a directed acyclic graph $G = (V, E)$, where an edge $(i, j) \in E$ indicates that task i must be completed before task j can begin. This graph is then encoded as a boolean precedence matrix $\mathbf{P} \in \{0,1\}^{n\times n}$, where $p_{ij} = 1$ if and only if task i strictly precedes task j. This representation enhances computational efficiency and simplifies constraint handling. The left panel lists each task identifier along with its processing times on robotic and manual workstations, as well as the associated hazard and demand levels.

Table 2
Notations and definitions.

Notation	Definition
Indices	
i	Task index, $i = 1, \ldots, I$
I	The number of tasks to be disassembled
m	Workstation index, $m = 1, \ldots, M$
M	The number of workstations
Parameters	
T_c	Theoretical cycle time
T_P	Practical cycle time
t_i	Disassembly time of task i
t_i^h	Disassembly time of task i in human workstations
t_i^r	Disassembly time of task i in robot workstations
c_i	Disassembly cost of task i
c_i^h	Disassembly cost of task i in human workstations
c_i^r	Disassembly cost of task i in robot workstations
e_i	Disassembly energy consumption of task i
e_i^h	Disassembly energy consumption of task i in human workstations
e_i^r	Disassembly energy consumption of task i in robot workstations
c_u^h	The operating cost of a human workstation per unit time
c_u^r	The operating cost of a robot workstation per unit time
c_u^α	The additional cost per unit time for a robot workstation to handle hazardous tasks
c_u^β	The additional cost per unit time for a flexible workstation to handle failed tasks
e_u^h	The operating energy consumption of a human workstation per unit time
e_u^r	The operating energy consumption of a robot workstation per unit time
e_u^α	The additional energy consumption per unit time for a robot workstation to hazardous tasks
Decision Variables	
r_i^h	1, task i is disassembled on the human workstation; 0, otherwise
r_i^r	1, task i is disassembled on the robot workstation; 0, otherwise
z_m^r	1, the robot sub-workstation of the workstation m handles harmful tasks; 0, otherwise
u_m	1, workstation m is opened; 0, otherwise
u_m^h	1, the human sub-station of the workstation m is opened; 0, otherwise
u_m^r	1, the robot sub-station of the workstation m is opened; 0, otherwise
x_{im}	1, task i is assigned to workstation m; 0, otherwise
h_i	1, task i is hazardous; 0, otherwise
d_i	1, task i is demanded; 0, otherwise

Based on these attributes and the precedence matrix, Fig. 2 illustrates a feasible task assignment: light-gray segments represent waiting times induced by precedence constraints, dark-gray segments correspond to idle times, and yellow segments denote actual disassembly durations. While satisfying all precedence relations, the bilateral



	1	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	0	1
3	0	1	0	0	0	0	0	0	0
4	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	0	0	0	1
6	0	1	0	0	0	0	0	0	0
7	0	0	0	0	1	0	0	0	0
8	0	0	1	1	0	1	0	0	0
9	0	0	0	0	0	0	0	0	0

Task	1	8	3	4	6	7	2	5	9
Time_Robot	2	2	3	4	2	4	3	1	2
Time_Human	4	5	4	7	3	8	3	2	3
Hazard	0	1	0	1	0	0	0	0	0
Demand	0	0	0	0	1	0	0	0	1

Fig. 2. Disassembly precedence constraint graph, disassembly precedence constraint matrix, and a disassembly example.

workstations operate in a coordinated manner. Moreover, a human-intervention mechanism is implemented on the robotic side: if a robot fails—due to a disassembly fault or other anomalies—the operator can intervene immediately, ensuring uninterrupted line operation.

3.2. Mathematical model

The notations in the mathematical model are shown in Table 2. The disassembly time t_i of task i is expressed as follows:

$$t_i = r_i^h t_i^h + r_i^r t_i^r \quad \forall i \in I \tag{1}$$

The disassembly cost c_i of task i is expressed as follows:

$$c_i = r_i^h c_i^h + r_i^r c_i^r \quad \forall i \in I \tag{2}$$

The disassembly energy consumption e_i of task i is expressed as follows:

$$e_i = r_i^h e_i^h + r_i^r e_i^r \quad \forall i \in I \tag{3}$$

The mathematical model for THCDLDB is as follows:

$$\min F = \min \left[f_1, f_2, f_3 \right] \tag{4}$$

$$\min f_1 = \sum_{m \in M} u_m \tag{5}$$

$$\min f_2 = \sum_{i \in I} c_i + c_u^h T_P \sum_{m \in M} u_m^h + c_u^r T_P \sum_{m \in M} u_m^r + c_u^\alpha T_P \sum_{m \in M} z_m^r + c_u^\beta T_P \tag{6}$$

$$\min f_3 = \sum_{i \in I} e_i + e_u^h T_P \sum_{m \in M} u_m^h + e_u^r T_P \sum_{m \in M} u_m^r + e_u^\alpha T_P \sum_{m \in M} z_m^r \tag{7}$$

Subject to:

$$\sum_{m \in M} x_{in} = 1, \forall i \in I \tag{8}$$

$$u_m \geq u_{m+1}, \forall m \in \{1, \dots, M-1\} \tag{9}$$

$$\sum_{m \in M} u_m \leq I \tag{10}$$

$$T_P \leq T_C, \forall m \in M \tag{11}$$

$$r_i^r \geq h_i, \forall i \in I \tag{12}$$

$$r_i^h \geq d_i, \forall i \in I \tag{13}$$

$$r_i^h + r_i^r = 1, \forall i \in I \tag{14}$$

The multi-objective optimization problem is formulated as a joint optimization task with three sub-objectives, as defined in Eq. (4), aiming to simultaneously minimize the number of workstations, disassembly cost, and energy consumption.

Eq. (5) defines the objective of minimizing the number of active workstations required to complete all disassembly tasks, reflecting the compactness of resource allocation. Eq. (6) represents the disassembly cost objective, which accounts for multiple cost components, including task-specific operation costs, operating costs of manual (H-side) and robotic (R-side) workstations, additional costs incurred by robots when handling hazardous tasks, and the extra cost of deploying a flexible manual backup station on the R side. Eq. (7) defines the energy consumption objective, incorporating the base energy of each task, energy usage by different workstation types during execution, and the additional energy required when R-side stations process hazardous tasks.

Together, the three objectives characterize the disassembly system's overall performance in efficiency, cost-effectiveness, and sustainability, forming the core basis for reward function design in the reinforcement learning optimization process.

Constraint (8) ensures that each disassembly task i is assigned to exactly one workstation m, preventing the possibility of task omission or duplication. This constraint guarantees the proper allocation of tasks across the workstations. Constraint (9) establishes a predefined activation sequence for the workstations, ensuring that they are activated in the correct operational order. This ensures the efficient flow of tasks along the disassembly line. Constraint (10) limits the total number of active workstations to be no more than the total number of disassembly tasks. This prevents the unnecessary activation of idle workstations, thereby optimizing resource usage. Constraint (11) sets a limit on the cycle time, ensuring that the operational duration of each workstation does not exceed its maximum allowable capacity. This is crucial for maintaining the balance and efficiency of the disassembly line. Constraint (12) requires that all hazardous tasks be assigned exclusively to robotic (R-side) stations. This is a safety measure, ensuring that dangerous tasks are handled by robots, minimizing risk to human workers. Constraint (13) mandates that tasks driven by demand be assigned to manual (H-side) stations. This ensures flexibility and precision in handling tasks that require human intervention or detailed manual handling. Constraint (14) prohibits any task from being assigned simultaneously to both sides of the disassembly line. This avoids conflicts and ensures that each task is handled by a specific workstation on either the robotic or manual side.

4. Proposed MO-PPO for THCDLDB

4.1. Reinforcement learning

The Markov Decision Process (MDP) is the fundamental theoretical framework in reinforcement learning, widely used to model the dynamic decision-making process of an agent interacting with its environment. An MDP is formally defined by a five-tuple (S, A, P, R, γ), where denotes the state space S representing all possible configurations of the system, A is the action space defining the set of actions available to the agent, P describes the probability distribution over next states given a current state and action, R quantifies the immediate reward received after executing an action in a given state, and γ is the discount factor that modulates the influence of future rewards on current decision-making. Under this framework, the agent continually interacts with the environment, selecting actions based on observed states and adjusting its policy through feedback in order to maximize the expected cumulative return. The MDP thus offers a principled structure for evaluating and optimizing decision strategies in complex, sequential environments.

4.2. MO-PPO

4.2.1. Proximal policy optimization

As a widely adopted reinforcement learning algorithm, proximal policy optimization (PPO) employs a clipped surrogate objective to mitigate the instability often encountered in traditional policy gradient updates. Formally, PPO maximizes the following objective at each iteration:

$$L^{\mathrm{CL}\mathbb{P}}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)\hat{A}_t, \mathrm{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right] \quad (15)$$

Where θ denotes the current policy parameters, $r_t(\theta) = \pi_\theta(a_t \mid s_t)/\pi_{\theta_{\mathrm{old}}}(a_t \mid s_t)$ is the probability ratio between new and old policies, ε is the clipping coefficient that constrains $r_t(\theta)$ within $[1-\varepsilon, 1+\varepsilon]$ to prevent excessive policy shifts, and $\hat{A}_t$ is the advantage estimate – typically computed via generalized advantage estimation (GAE) – which quantifies the relative benefit of action a_t in state s_t. By optimizing this clipped objective without the need for complex constrained procedures, PPO achieves both stable and efficient training, consistently demonstrating superior performance on high-dimensional control and game environments.

4.2.2. MO-PPO

In classical reinforcement learning frameworks, the agent's policy is typically designed to maximize a scalar reward function. However, real-world decision-making, particularly in industrial settings, often involves multiple conflicting objectives that cannot be adequately modeled or optimized through a single metric. In the disassembly line balancing problem, common objectives include minimizing cost, reducing energy consumption, and limiting the number of required workstations—goals that frequently conflict, making traditional reinforcement learning methods insufficient for capturing the full complexity of the task.

To address the aforementioned limitations, this study proposes a MO-PPO algorithm by integrating a preference vector mechanism into the standard PPO framework, as illustrated in Fig. 3. To capture the relative importance of each objective, a preference vector $\mathbf{w} = [w_1, w_2, \ldots, w_k]$ is introduced, where each component w_k denotes the weight assigned to the kth objective. During training, the agent samples a preference vector from a predefined distribution at the beginning of each episode and inputs it alongside the state into the policy network. Then, The agent selects an action based on the output of the policy network and executes the task through interaction with the environment. This mechanism enables the agent to dynamically adapt its behavior based on varying objective preferences, thereby improving its ability to optimize multiple objectives in a flexible and responsive manner.

The policy network (Actor) is responsible for selecting actions based on the current state and preference vector. It receives the joint input of the environment state s and preference vector $\mathbf{w}$, and outputs a conditional policy $\pi(a \mid s, \mathbf{w})$. Implemented as a multilayer perceptron (MLP), the actor network takes both the state and preference vector as input and produces a probability distribution over actions. During training, the network dynamically adjusts its output according to the sampled preference vector, enabling the agent to make adaptive decisions under varying objective weights. This design endows the agent with the flexibility to respond not only to environmental changes but also to shifting user preferences.

Through interaction with the environment, the agent collects experience tuples comprising the current state, preference vector, action taken, reward, next state, and the associated action probability. These experiences are stored in a replay buffer for subsequent policy updates. Once a sufficient number of samples is collected, mini-batches are drawn from the buffer to update the policy, ensuring that training reflects diverse objective preferences. This mechanism allows the agent to effectively manage conflicts and priority shifts in multi-objective tasks.

In multi-objective tasks, the agent must simultaneously optimize multiple objectives, with their relative importance potentially varying across episodes. To accommodate this variability, the reward function is represented as a multi-dimensional vector:

$$r(s,a) = [r_1(s,a), r_2(s,a), \ldots, r_k(s,a)] \quad (16)$$

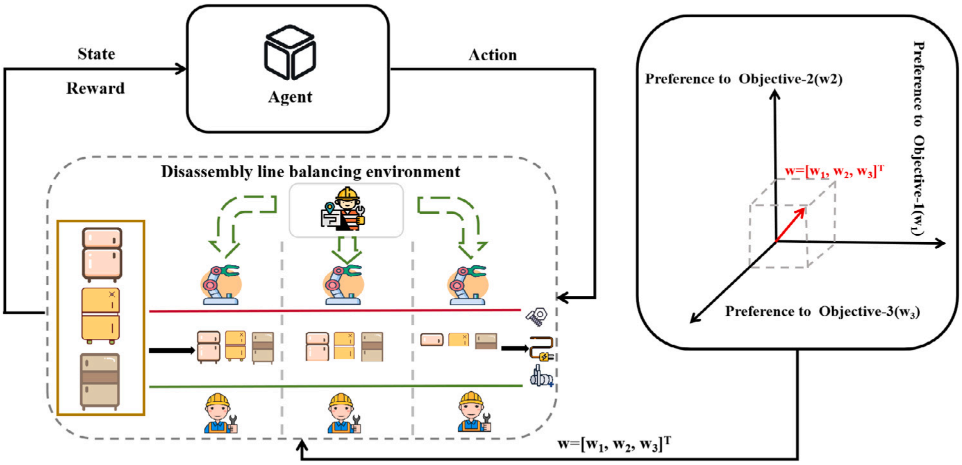

State
Reward
Agent
Action
Disassembly line balancing environment
Preference to Objective-2(w2)
Preference to Objective-1(w1)
w=[w1, w2, w3]T
Preference to Objective-3(w3)
w=[w1, w2, w3]T

Fig. 3. The overall process of the MO-PPO algorithm.

Here, $r_i(s, a)$ denotes the immediate reward for the ith objective obtained by taking action a in state s_t. During training, a scalar reward signal is constructed by computing the weighted sum of the reward vector and the preference vector:

$$r^{\mathbf{w}}(s, a) = \mathbf{w}^{\top}\mathbf{r}(s, a) \tag{17}$$

To implement the above mechanism, the disassembly line balancing problem is formalized as a Multi-objective Markov Decision Process (MOMDP), defined as a seven-tuple $\mathcal{M} = (S, A, T, \mathbf{r}, \gamma, \mathbf{w}, \pi)$, where $\mathbf{r}$ denotes the reward vector function and $\mathbf{w}$ represents the preference vector.

In the early stages of training, preference vectors are randomly sampled from the Dirichlet distribution, generating a vector containing n components, each representing a preference weight for an optimization objective. These weights are constrained to be positive and sum to 1, ensuring that the preference vector can dynamically adjust the relative importance of different objectives. The random sampling approach encourages the model to extensively explore the solution space in its initial phases, preventing premature convergence to local optima, thereby enhancing the coverage of the solution space and improving the model's global search capabilities.

As training progresses, the model dynamically adjusts the preference vectors based on task progress and optimization requirements, flexibly modifying the weights of each objective to maintain an effective balance among them. The model can perform non-dominated sorting to select the preference vectors corresponding to solutions near the Pareto front or conduct targeted sampling based on specific task requirements, while maintaining a degree of randomness to facilitate a more comprehensive exploration of the solution space. This adaptive sampling approach helps the model gain a deeper understanding of the diversity within the solution space.

Additionally, the preference vector is used along with the state space as input to the policy network, enabling the PPO network to implicitly learn conditional policies during training. This design allows the model to adaptively adjust the weights between different objectives, effectively addressing dynamic and complex multi-objective optimization tasks, further enhancing the model's performance and flexibility in practical applications.

To mitigate the instability caused by differences in scale and units across objectives, all reward dimensions are normalized to a common scale. Specifically, we apply min–max normalization to each objective, transforming the values to a range between 0 and 1. This enhances numerical stability in policy gradient estimation and accelerates convergence by ensuring that each objective contributes proportionally to the overall reward, regardless of its original scale.

In the implementation of our MO-PPO algorithm, we primarily utilize a multilayer perceptron (MLP) architecture. Both the actor network and critic network consist of two hidden layers, with each layer containing 256 neurons. The Tanh activation function is applied after each hidden layer, while the output layer of the actor network employs the softmax function.

In summary, the proposed MO-PPO framework integrates multi-objective reinforcement learning, preference modeling, and conditional policy learning into a cohesive and extensible structure. It offers a flexible and generalizable solution particularly well-suited for industrial decision-making problems like disassembly line balancing, where multiple objectives and evolving preferences must be simultaneously addressed. The algorithmic procedure is detailed in Algorithm 1, which outlines the step-by-step implementation of the proposed method.

Algorithm 1 MO-PPO

```
Initialize actor network $\pi(a \mid s; \theta)$ and critic network $V(s; \phi)$
Initialize replay buffer $\mathcal{D}$
for episode = 1 to M do
    Sample a preference vector $\mathbf{w} \sim \mathcal{W}$
    Receive initial state $s_0$ from the environment
    for $t = 0$ to T do
        Construct extended state $s_t = [s_t, \mathbf{w}]$
        Sample action $a_t \sim \pi(a \mid s_t; \theta)$
        Execute action $a_t$ to receive reward vector
        $\mathbf{r}_t = [r_t^1, r_t^2, \dots, r_t^k]$ and next state $s_{t+1}$
        Compute scalarized reward: $r_t^{\mathbf{w}} = \mathbf{w}^{\top}\mathbf{r}_t$
        Store transition $(s_t, \mathbf{w}, a_t, r_t^{\mathbf{w}}, s_{t+1})$ in replay buffer
    end for
    Sample a mini-batch from $\mathcal{D}$
    Compute estimated return $\hat{R}_t$ and $\hat{A}_t$ using GAE
    for $k = 1$ to K do
        Compute probability ratio:
        $r_t(\theta) = \pi_\theta(a_t \mid s_t)/\pi_{\theta_{\text{old}}}(a_t \mid s_t)$
        Compute actor loss (clipped objective):
        $L^{\text{clip}}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right]$
        Compute critic loss:
        $L^{\text{value}}(\phi) = \mathbb{E}_t\left[\left(V(s_t; \phi) - \hat{R}_t\right)^2\right]$
        Update parameters:
        $\theta \leftarrow \theta + \alpha\nabla_\theta L^{\text{clip}}(\theta), \quad \phi \leftarrow \phi - \alpha\nabla_\phi L^{\text{value}}(\phi)$
    end for
end for
```

Table 3
The constant of the model.

Notation	I	T_e	c_u^h	c_u^r	c_u^α	c_u^β	e_u^h	e_u^r	e_u^α
Value	20	70 s	0.03 Yuan/s	0.04 Yuan/s	0.01 Yuan/s	0.01 Yuan/s	0.5 kW	1.2 kW	0.3 kW

4.3. Applied MO-PPO to THCDLDB

4.3.1. State space

The state space constructed in this study is designed to comprehensively capture the key dynamic characteristics of the disassembly environment, encompassing both the status of disassembly task nodes and the operational states of disassembly workstations. Task node states describe the real-time execution status of each node, including whether it has been selected, its selection order, and the workstation side (manual or robotic) it currently occupies. In contrast, workstation states reflect the operational conditions of various stations along the disassembly line, such as whether a station is active, the activation status of manual and robotic sides, and the cumulative processing time at each station. Together, these state features effectively represent resource allocation and task progression across different stages of the process, providing the reinforcement learning agent with precise and informative state inputs for decision-making.

4.3.2. Action space

The agent's action decision process is structured into two stages: task selection and workstation assignment. In the first stage, the agent selects a disassembly task from a set of candidates that strictly satisfy predefined precedence constraints, thereby avoiding infeasible task sequences. At each decision step, the candidate set includes only those tasks whose immediate predecessors have been completed. Upon selecting a task, the precedence constraint matrix is updated to remove the dependencies associated with the completed task, ensuring that all subsequent selections adhere to the required task order.

In the second stage, the selected task is allocated to an appropriate workstation based on its attributes, such as hazard level and demand. High-risk tasks must be assigned to robotic workstations to ensure operational safety, while tasks with high demand priority are preferentially assigned to manual workstations to enhance efficiency. For remaining tasks, the agent autonomously determines optimal workstation allocation by evaluating the current environment and policy model. This two-stage action modeling approach balances physical and logical constraints while enabling adaptive and intelligent decision-making.

4.3.3. Reward function

The reward function in this study integrates three key objectives: the number of disassembly workstations, disassembly cost, and energy consumption. To mitigate the negative impact of scale differences among these objectives on learning efficiency, all objective values are normalized and combined using a weighted sum, yielding the final reward signal for the agent.

During each interaction round, the agent generates an action sequence that, once decoded, determines the workstation allocation and task execution order. Based on this sequence, the actual number of workstations used, total disassembly cost, and energy consumption are calculated to evaluate the reward. The specific objective functions are defined as follows.

$$R = -\left(w_1 f_1 + w_2 f_2 + w_3 f_3\right) \tag{18}$$

5. Experiment

PPO is widely recognized as a representative and efficient algorithm in deep reinforcement learning, with proven success in robotic control, game strategies, and dynamic system scheduling. Meanwhile, classical multi-objective evolutionary algorithms such as NSGA-II, NSGA-III, and SPEA2 have long played a key role in solving multi-objective optimization problems. To comprehensively evaluate the performance of the proposed MO-PPO, we conduct a systematic comparison with both reinforcement learning and evolutionary baselines. By quantifying the solution quality across algorithms, this study aims to provide insightful evaluations in terms of performance, convergence, and adaptability, thereby validating the effectiveness of MO-PPO for complex optimization tasks.

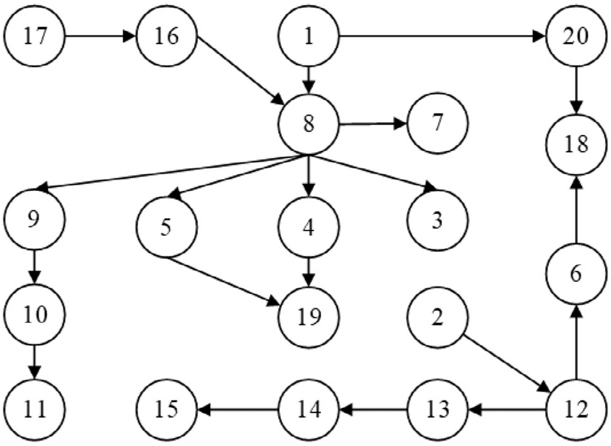


Fig. 4. Precedence relationship of disassembly tasks.

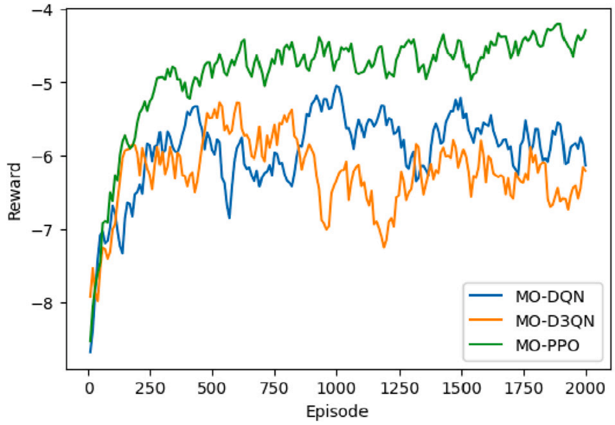


Fig. 5. The reward curves of MO-DQN, MO-D3QN, MO-PPO.

5.1. Disassembly instances

In disassembly scenarios, refrigerators represent a common type of large-scale EOL product. This study uses the refrigerator disassembly line of a dismantling enterprise as a case example, extracting key task-related information including task attributes, disassembly time, cost, and energy consumption. Based on the collected data, model constants are defined as shown in Table 3. Table 4 summarizes the attributes of each disassembly task, along with the corresponding time, cost, and energy consumption for different sides of the disassembly line. The precedence constraints among disassembly tasks are illustrated in Fig. 4.

Table 4
Disassembly task information.

No	Task	Hazard	Demand	time_r	cost_r	energy_r	time_h	cost_h	energy_h
1	Hinge	0	1	13	0.9	0.0028	17	1.1	0.0025
2	Refrigerator shelf	0	1	12	1.1	0.0028	15	1.3	0.0025
3	Drawers of refrigeration door	0	1	14	1.0	0.0042	20	1.2	0.0039
4	Refrigeration door	0	1	9	1.0	0.0028	11	0.5	0.0028
5	The freezer door	0	0	10	0.4	0.0000	13	0.6	0.0000
6	Power line	0	0	10	0.3	0.0000	17	0.5	0.0000
7	Insulation layer of freezer door	1	0	12	0.5	0.0000	15	0.6	0.0000
8	Display panel	0	1	12	0.3	0.0028	14	0.5	0.0025
9	Dry filter	0	1	13	1.0	0.0042	13	1.5	0.0033
10	Bulb	0	0	11	0.3	0.0028	11	1.2	0.0019
11	Junction box assembly	0	0	12	1.1	0.0028	12	1.3	0.0017
12	Compressor	0	0	13	1.0	0.0022	13	1.2	0.0019
13	Refrigerator box	0	0	10	0.8	0.0017	14	1.2	0.0014
14	Gear	0	0	11	0.9	0.0028	12	1.2	0.0019
15	Lamp cover and lampshade	0	0	13	1.0	0.0028	14	1.3	0.0020
16	Thermostat knob	0	0	13	0.8	0.0017	12	1.0	0.0014
17	Screw	0	1	13	1.9	0.0028	12	2.1	0.0025
18	Season switch	0	1	13	2.4	0.0056	14	2.6	0.0061
19	Automatic temperature regulator	0	0	9	1.7	0.0041	13	1.9	0.0045
20	Screw	0	1	11	1.2	0.0056	14	1.4	0.0053

Table 5
The parameters of the algorithm.

Parameters	Discount factor	Clipping parameter	Learning rate	Training episodes	Replay memory
Value	0.99	0.2	0.0001	2000	1000

Table 6
The disassembly scheme of the MOPPO algorithm.

Algorithm	WS	Side	Assignment sequence	T	T_P	F1	F2	F3
MO-PPO	1	H	1-2-4-17-20	69.00	69.00	2.00	32.64	255.35
	1	R	7-(W4s)-8-16-9-12	67.00				
	2	H	6-18-3-15	65.00				
	2	R	13-5-19-14-10-11	66.00				

5.2. Comparative experiment

5.2.1. Comparative experiment with classical reinforcement learning algorithms

The detailed experimental configurations for the proposed MO-PPO algorithm are summarized in Table 5. Comparative evaluations against two representative multi-objective reinforcement learning algorithms – MO-DQN and MO-D3QN – demonstrate that MO-PPO consistently outperforms both baselines in terms of solution quality and convergence speed. As shown in Fig. 5, although MO-DQN and MO-D3QN exhibited moderate improvements in cumulative rewards during the early stages of training, they eventually plateaued within the range of –6.5 to –5, which is substantially lower than the reward values achieved by MO-PPO. Moreover, as shown in Figs. 6, 7 and 8 their objective values displayed significant instability: $f1$ fluctuated between 3 and 4, $f2$ between 38 and 44, and $f3$ between 400 and 550, indicating an inability to achieve satisfactory convergence. Consequently, these two methods were excluded from further comparative analysis in subsequent experiments.

For MO-PPO, during the initial training phase, reward values were low and exhibited substantial volatility. The objective values ($f1, f2, f3$) varied widely and occasionally approached near-optimal levels, yet failed to simultaneously attain the joint optimal region. As training progressed, the learned policy showed consistent improvement, with increasing and stabilizing reward values and reduced variance—indicative of successful parameter tuning. The three objective metrics demonstrated converging downward trends and eventually stabilized near their respective optimal values. The final disassembly scheme derived from the MO-PPO-trained policy is provided in Table 6.

As illustrated in Figs. 6, 7, and 8, the MO-PPO algorithm ultimately converges to a joint optimal solution of the objective function: (2.00, 32.64, 255.35), highlighting its effectiveness and robustness in addressing multi-objective disassembly line balancing problems.

5.2.2. Comparative study with classical metaheuristic algorithms

To systematically assess the performance of the proposed MO-PPO algorithm in solving multi-objective optimization problems, this section constructs a benchmark comparison with three representative classical metaheuristic algorithms: NSGA-II, NSGA-III, and SPEA2. Each algorithm is executed independently 30 times to mitigate the effects of randomness and ensure the robustness of the evaluation results. The best solutions obtained by each method are summarized in Table 7, where values in bold black indicate the current Pareto-optimal results across all algorithms. The notation “W2s” denotes a required 2-second delay between two tasks to ensure that the precedence constraint – completion of the predecessor task prior to the current one – is satisfied during the disassembly process.

The experimental results reveal that MO-PPO achieves superior solution quality across all three objective dimensions, highlighting its effectiveness and potential in solving complex optimization problems. Compared with traditional metaheuristic methods, MO-PPO leverages a pre-trained deep neural network, providing enhanced generalization and adaptability. This enables the algorithm to rapidly generate feasible and high-quality solutions in dynamically changing environments. Further validation under dynamic conditions will be presented in the next section.

5.3. Dynamic rebalancing

Compared to product manufacturing and assembly processes, disassembly of EOL is subject to various uncertainties, such as fastener corrosion, component deformation, and missing parts. Typically, missing

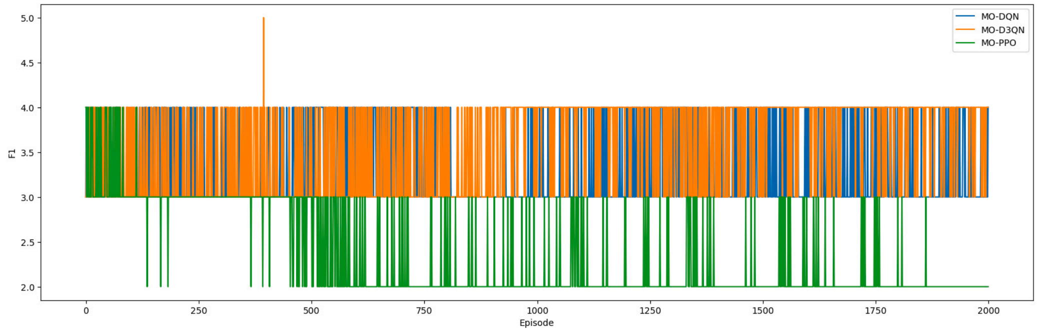


Fig. 6. The training process curves of f1 for MO-DQN, MO-D3QN, MO-PPO.

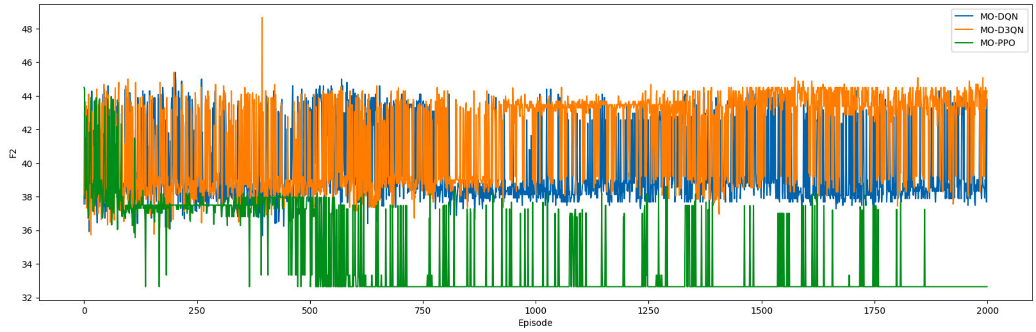


Fig. 7. The training process curves of f2 for MO-DQN, MO-D3QN, MO-PPO.

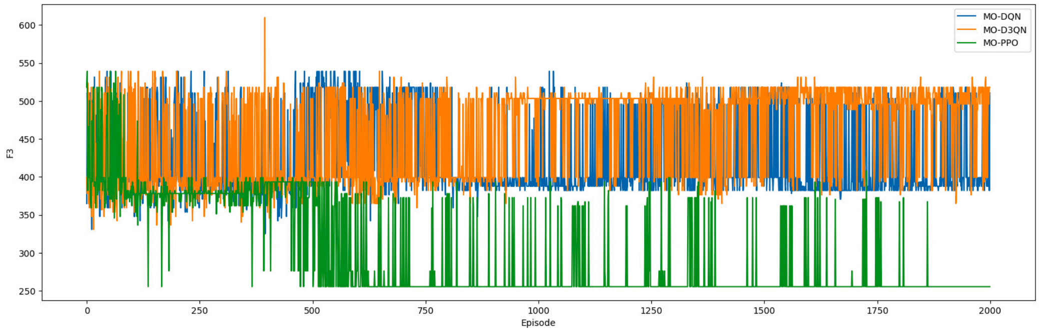


Fig. 8. The training process curves of f3 for MO-DQN, MO-D3QN, MO-PPO.

Table 7
The disassembly scheme.

Algorithm	WS	Side	Assignment sequence	T	T_P	F1	F2	F3
MO-PPO	1	H	1-2-4-17-20	69.00	69.00	**2.00**	**32.64**	**255.35**
	1	R	7-(W4s)-8-16-9-12	67.00				
	2	H	6-18-3-15	65.00				
	2	R	13-5-19-14-10-11	66.00				
NSGA-II	1	H	2-1-20-13	61.00	69.00	**2.00**	**32.64**	**255.35**
	1	R	7-(W2s)-12-(W6s)-8-9-10	69.00				
	2	H	16-4-6-14-18	69.00				
	2	R	3-5-(W1s)-19-17-11-15	69.00				
NSGA-III	1	H	1-2-20-(W7s)-13	68.00	69.00	**2.00**	**32.64**	**255.35**
	1	R	7-(W4s)-8-16-12-9	67.00				
	2	H	14-6-17-4-15	67.00				
	2	R	3-10-11-5-18-19	69.00				
SPEA2	1	H	1-2-16	46.00	55.00	3.00	34.45	297.05
	1	R	7-(W4s)-8-9-12	54.00				
	2	H	6-20-14-4	55.00				
	2	R	13-5-10	48.00				
	3	H	3-18-5	31.00				
	3	R	empty	0.00				

Table 8
The disassembly scheme.

Algorithm	WS	Side	Assignment sequence	T	T_P	F1	F2	F3	Running time
MO-PPO	1	H	2-1-13-20	63.00	69.00	**2.00**	**32.64**	**255.35**	**0.056 s**
	1	R	7-(W2s)-12-(W6s)-8-9-10	69.00					
	2	H	6-16-4-19-15	69.00					
	2	R	14-5-3-11-18-17	69.00					
NSGA-II	1	H	1-2-20-16	62.00	68.00	**2.00**	**32.34**	**255.35**	35 s
	1	R	7-(W4s)-8-9-12-13	68.00					
	2	H	6-4-18-17-11	70.00					
	2	R	5-14-3-10-19-15	70.00					
NSGA-III	1	H	2-1-8-3	66.00	70.00	**2.00**	32.80	259.05	38 s
	1	R	7-(W2s)-12-13-(W6s)-5-9	70.00					
	2	H	20-17-4-6-18	70.00					
	2	R	16-10-14-15-11-19	69.00					
SPEA2	1	H	2-1-20-(W2s)-15	66.00	70.00	**2.00**	33.50	280.05	33 s
	1	R	7-(W2s)-12-13-14-8	62.00					
	2	H	4-6-18-10-11	69.00					
	2	R	5-16-9-17-3-19	70.00					

Table 9
The disassembly scheme.

Algorithm	WS	Side	Assignment sequence	T	T_P	F1	F2	F3	Running time
MO-PPO	1	H	2-1-7-19	65.00	69.00	**2.00**	**31.50**	240.55	**0.048 s**
	1	R	6-(W2s)-11-12-13-8	64.00					
	2	H	4-15-14-3	59.00					
	2	R	5-9-17-18-10-16	65.00					
NSGA-II	1	H	1-2-19-9	59.00	64.00	**2.00**	31.54	**236.85**	32 s
	1	R	6-(W4s)-7-8-11-5	64.00					
	2	H	12-4-13-14-18	64.00					
	2	R	10-15-3-16-17	62.00					
NSGA-III	1	H	2-1-19-(W6s)-15	66.00	66.00	**2.00**	31.66	244.25	38 s
	1	R	6-(W2s)-11-12-7-8	66.00					
	2	H	4-17-16-14	51.00					
	2	R	13-9-3-10-5-18	66.00					
SPEA2	1	H	1-2-19-3	66.00	66.00	**2.00**	31.66	244.25	32 s
	1	R	6-(W3s)-7-(W3s)-11-8	57.00					
	2	H	17-15-4-13-14	66.00					
	2	R	12-9-10-16-5-18	66.00					

components do not require disassembly and are assigned a disassembly time of zero, whereas deformed parts may significantly prolong disassembly time. These uncertainties complicate the accurate estimation of task durations and may even lead to disassembly failure.

Such variability directly affects the applicability of the disassembly schedule presented in Table 7, potentially rendering it invalid under changed conditions. Therefore, real-time replanning of disassembly tasks is essential to ensure the adaptability and robustness of the disassembly line in dynamic environments.

5.3.1. Experiment on the impact of component deformation on disassembly time

This section investigates the algorithm's robustness in the presence of realistic operational disturbances, specifically simulating a scenario where component deformation leads to extended disassembly durations—a frequently encountered source of uncertainty in industrial disassembly lines. In this case study, Component 15 is assumed to be deformed, resulting in increased disassembly times of 14 s on the R side and 16 s on the H side. This disturbance is introduced to assess the resilience of the proposed method under time-based disruptions. To benchmark performance, the MO-PPO model – trained under nominal conditions – is evaluated alongside three widely used metaheuristic algorithms: NSGA-II, NSGA-III, and SPEA2. The comparative outcomes are presented in Table 8.

According to the results in Table 8, MO-PPO successfully generates a valid disassembly line balancing plan within just 0.056 s, illustrating its high computational efficiency and rapid adaptability to environmental changes. The resulting objective values are (2.00, 32.64, 255.35). While the cycle time is marginally longer – by 1 second – than that achieved by NSGA-II, both MO-PPO and NSGA-II attain equally optimal values across the three objectives $f1$, $f2$ and $f3$. Importantly, MO-PPO demonstrates superior runtime performance relative to all benchmarked metaheuristics. A comparison with the baseline configuration in Table 7 reveals partial reassignments of disassembly tasks in Table 8, highlighting MO-PPO's capacity for dynamic task adjustment and real-time replanning under uncertainty.

5.3.2. Experimental analysis of zero disassembly time caused by missing components

This section simulates a practical disassembly scenario involving the absence of a component, in order to evaluate task replanning capabilities in dynamic environments. Specifically, the absence of Component 6 is assumed, resulting in a disassembly time of zero for that part. This setting mimics real-world cases where missing components may disrupt the disassembly process. The trained MO-PPO model is compared against three classical metaheuristic methods—NSGA-II, NSGA-III, and SPEA2. The comparative results are summarized in Table 9.

As shown in Table 9, the trained MO-PPO model is capable of generating a feasible disassembly line balancing solution within 0.048 s. In the updated scenario, the optimized objective values achieved by MO-PPO are (2.00, 31.50, 240.55). Among all compared algorithms, MO-PPO produces the best results for the first two objectives ($f1$ and $f2$), while the third objective ($f3$) is slightly inferior to that of NSGA-II. This marginal reduction in $f3$ performance may be attributed to the altered precedence constraints caused by the missing component, which affects the generalization of the pre-trained MO-PPO model. In contrast, metaheuristic algorithms re-optimize from scratch under the

Table 10
The mean and variance of the experiments in Sections 5.2.2, 5.3.1, and 5.3.2.

Experiment	Algorithm	Objective function		
		f1	f2	f3
1	MO-PPO	2.1 ± 0.31	32.81 ± 0.51	259.16 ± 11.66
	NSGA-II	2.13 ± 0.35	32.86 ± 0.58	260.55 ± 13.52
	NSGA-III	2.13 ± 0.34	32.85 ± 0.56	260.37 ± 13.04
	SPEA2	3.0 ± 0.00	34.49 ± 0.16	301.19 ± 16.60
2	MO-PPO	2.0 ± 0.00	32.70 ± 0.70	257.09 ± 4.14
	NSGA-II	2.00 ± 0.00	32.50 ± 0.40	258.76 ± 8.51
	NSGA-III	2.00 ± 0.00	32.89 ± 0.24	261.94 ± 7.24
	SPEA2	2.14 ± 0.04	33.75 ± 0.63	284.62 ± 15.91
3	MO-PPO	2.0 ± 0.00	31.58 ± 0.15	242.61 ± 3.54
	NSGA-II	2.00 ± 0.00	31.56 ± 0.09	237.67 ± 3.07
	NSGA-III	2.00 ± 0.00	31.68 ± 0.08	244.79 ± 1.93
	SPEA2	2.00 ± 0.00	31.69 ± 0.10	245.07 ± 2.33

new conditions, potentially achieving better objective values at the cost of significantly higher computational time. A comparison between Tables 7 and 9 further reveals task replanning within the disassembly line, highlighting the proposed method's adaptability and operational viability in engineering applications.

While metaheuristic algorithms can yield competitive solutions after numerous iterations, their computational cost is substantial and they exhibit limited responsiveness to dynamic changes. The simulation results reinforce the advantages of MO-PPO in dynamic environments, demonstrating strong generalization capabilities and the ability to maintain effective disassembly line balance in the presence of structural disruptions.

5.4. Evaluation of the impact of theoretical cycle time on experimental results

We conducted experiments in Sections 5.2.2, 5.3.1, and 5.3.2, sequentially numbered as Experiment 1, Experiment 2, and Experiment 3, and calculated the mean and standard deviation of the objective functions, as recorded in Table 10. During the experiments, we observed that the dual-sided disassembly line in human–robot collaboration exhibits unique characteristics. It requires careful consideration of human–robot cooperation while adhering to task priority constraints. Specifically, a disassembly task cannot be executed unless its predecessor task has been completed. Even minor changes can lead to significant increases in idle time, necessitating the activation of additional workstations to complete the disassembly tasks. These traits are inherent to human–robot collaborative disassembly lines.

Experiment 1 used the original data, Experiment 2 explored changes in the disassembly time of a particular task, and Experiment 3 simulated a scenario where the absence of a component led to a reduction in disassembly tasks. Analysis of Table 10 revealed that the solution variation in Experiment 3 was smaller and more stable compared to Experiment 1. We attribute this to the reduction in disassembly tasks, which provided more flexibility in the original cycle time. To address this issue, we designed further experiments to investigate the impact of different cycle times on disassembly. These experiments were conducted for theoretical cycle times of 50, 60, 70, 80, 90, and 100, with the objective functions of various algorithms recorded in Table 11.

From the analysis of Table 11, it can be observed that when the cycle time is shorter, more workstations are required, which increases the demand for infrastructure and labor. As the cycle time increases, the number of workstations decreases from 3 to 2. However, when the cycle time becomes too long, the number of workstations remains the same, resulting in more idle time and a significant increase in energy consumption during disassembly. Therefore, selecting an appropriate disassembly cycle time is crucial for optimizing the disassembly process.

6. Conclusion

This study addresses the multi-objective optimization problem of human–robot collaborative two-sided disassembly line balancing. An optimization model is proposed with the objectives of minimizing the number of workstations, disassembly cost, and energy consumption. Based on the model's characteristics, a state space consisting of eight feature categories and an action space structure that adheres to precedence constraints are designed. To solve this problem, a preference-based multi-objective reinforcement learning algorithm, MO-PPO, is developed. This innovative approach not only enables real-time decision-making and the flexible balancing of conflicting objectives but also overcomes the limitations of traditional methods, significantly enhancing adaptability and decision-making efficiency.

To validate the effectiveness of the proposed method, the preference-based MO-PPO is applied to a case study involving refrigerator disassembly. The simulation results demonstrate superior optimization performance and the generation of high-quality disassembly schedules. Analysis of the case results indicates that the proposed model and algorithm substantially improve the rationality of task allocation and resource utilization efficiency, providing new insights and practical validation for balancing human–robot collaborative disassembly lines in multi-objective environments.

However, the current model has been trained within the specific context of refrigerator disassembly, and while it performs well for this task, applying it to other EOL products (such as washing machines, televisions, air conditioners, etc.) may present challenges. The disassembly processes, workflows, component structures, and resource utilization efficiencies of different products vary significantly, requiring the model to possess robust adaptability and generalization capabilities. If the new EOL product shares similarities with the refrigerator disassembly process (e.g., similar components and disassembly steps), the model can be adapted via transfer learning or fine-tuning without the need for retraining from scratch. However, for products with substantial

Table 11
Results of different algorithms under varying T_C values.

T_C	50			60			70		
Objective function	f1	f2	f3	f1	f2	f3	f1	f2	f3
MO-PPO	**3.00**	**32.40**	**256.55**	**3.00**	**33.11**	**273.65**	**2.00**	**32.64**	**255.35**
NSGA-II	**3.00**	**32.40**	**256.55**	**3.00**	33.12	**273.65**	**2.00**	**32.64**	**255.35**
NSGA-III	**3.00**	32.64	262.25	**3.00**	33.12	273.65	**2.00**	**32.64**	**255.35**
SPEA2	**3.00**	32.64	262.25	**3.00**	33.36	297.35	**3.00**	34.45	297.05
T_C	80			90			100		
Objective function	f1	f2	f3	f1	f2	f3	f1	f2	f3
MO-PPO	**2.00**	32.80	255.36	**2.00**	**32.96**	**262.75**	**2.00**	**34.16**	**299.75**
NSGA-II	**2.00**	**32.64**	255.36	**2.00**	**32.96**	**262.75**	**2.00**	**34.16**	**299.75**
NSGA-III	**2.00**	33.37	**255.35**	**2.00**	**32.96**	**262.75**	**2.00**	35.44	310.84
SPEA2	**2.00**	33.37	259.05	**2.00**	**32.96**	**262.75**	**2.00**	35.44	310.85

differences, the model would require retraining or large-scale transfer learning, although existing model parameters could be used as initializations.

To enhance the model's generalizability, future research will focus on the following directions: first, incorporating real-world uncertainties, such as equipment failures, operator rest periods, and workload balancing, to improve the robustness of the model; second, extending the framework to support various disassembly line configurations, such as U-shaped or hybrid layouts, to increase its applicability; and third, applying the preference-based MO-PPO approach to other multi-objective optimization problems, such as the Traveling Salesman Problem, Job Shop Scheduling, and Assembly Line Balancing, to demonstrate its scalability and versatility.

CRediT authorship contribution statement

Jinlong Wang: Writing – review & editing, Writing – original draft, Methodology, Funding acquisition, Conceptualization. **Min Li:** Writing – review & editing, Writing – original draft, Investigation. **Fanyun Meng:** Writing – review & editing, Writing – original draft, Conceptualization. **Haoran Zhao:** Writing – review & editing, Writing – original draft, Funding acquisition, Conceptualization. **Xin Sun:** Writing – review & editing, Methodology, Investigation, Conceptualization.

Acknowledgments

This research was funded by the Shandong Province Natural Science Foundation, China (No. ZR2023QF043), the Shandong Provincial Key Research and Development Program (Major Scientific and Technological Innovation Project), China (No. 2025CXGC010902), the National Key R&D Program of China (No. 2022YFB3305800) and the Key Technology Tackling and Industrialization Demonstration Projects of Qingdao (No. 24-1-2-qljh 25-gx).

Data availability

Data will be made available on request.

References

Agrawal, S., & Tiwari, M. K. (2008). A collaborative ant colony algorithm to stochastic mixed-model U-shaped disassembly line balancing and sequencing problem. *International Journal of Production Research*, *46*(6), 1405–1429.

Aydemir-Karadag, Ayyuce, & Turkbey, Orhan (2013). Multi-objective optimization of stochastic disassembly line balancing with station paralleling. *Computers & Industrial Engineering*, *65*(3), 413–425.

Bai, Chunguang, Zhou, Hua, & Sarkis, Joseph (2023). Evaluating industry 4.0 technology and sustainable development goals–a social perspective. *International Journal of Production Research*, *61*(23), 8094–8114.

Bentaha, Mohand Lounes, Battaïa, Olga, & Dolgui, Alexandre (2015). An exact solution approach for disassembly line balancing problem under uncertainty of the task processing times. *International Journal of Production Research*, *53*(6), 1807–1818.

Caterino, Mario, Iannone, Raffaele, Macchiaroli, Roberto, Riemma, Stefano, Pham, Duc Truong, & Fera, Marcello (2025). Enhancing remanufacturing operations: A review on decision-making models and their implementation challenges. *Computers & Industrial Engineering*, Article 111088.

Cella, Christian, Robin, Matteo Bruce, Faroni, Marco, Zanchettin, Andrea Maria, & Rocco, Paolo (2025). Digital model-driven genetic algorithm for optimizing layout and task allocation in human-robot collaborative assemblies. In *2025 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 3103–3109). IEEE, http://dx.doi.org/10.1109/ICRA55743.2025.11127401.

Cevikcan, Emre, Aslan, Dicle, & Yeni, Fatma Betul (2020). Disassembly line design with multi-manned workstations: a novel heuristic optimisation approach. *International Journal of Production Research*, *58*(3), 649–670.

Chand, Mirothali, & Ravi, Chandrasekar (2024). A novel reinforcement learning framework for disassembly sequence planning using Q-learning technique optimized using an enhanced simulated annealing algorithm. *AI EDAM*, *38*, Article e5.

Chu, Mengling, & Chen, Weida (2024). Multi-manned disassembly line balancing problems for retired power batteries based on hyper-heuristic reinforcement. *Computers & Industrial Engineering*, *194*, Article 110400.

Dalle Mura, Michela, Pistolesi, Francesco, Dini, Gino, & Lazzerini, Beatrice (2021). End-of-life product disassembly with priority-based extraction of dangerous parts. *Journal of Intelligent Manufacturing*, *32*(3), 837–854.

Dong, ChengShun, Liu, Peisheng, Guo, Xi Wang, Qi, Liang, Qin, Shujin, & Xu, Gongdan (2021). Multi-objective ant lion optimizer for stochastic robotic disassembly line balancing problem subject to resource constraints. In *Journal of physics: conference series*: *vol. 2024*, (1), IOP Publishing, Article 012014.

El Jaouhari, Asmae, Samadhiya, Ashutosh, Benbrahim, Fatima Zahra, Kumar, Anil, & Luthra, Sunil (2025). Forging a green future: Synergizing industry 4.0 technologies and circular economy tactics to achieve net-zero in sustainable supply chains. *Computers & Industrial Engineering*, *201*, Article 110691.

Gungor, Askiner, & Gupta, Surenda M. (2001). A solution approach to the disassembly line balancing problem in the presence of task failures. *International Journal of Production Research*, *39*(7), 1427–1467.

Guo, Xiwang, Wei, Tingting, Wang, Jiacun, Liu, Shixin, Qin, Shujin, & Qi, Liang (2022). Multiobjective U-shaped disassembly line balancing problem considering human fatigue index and an efficient solution. *IEEE Transactions on Computational Social Systems*, *10*(4), 2061–2073.

Guo, Lei, Zhang, Zeqiang, Wu, Tengfei, Zhang, Yu, Zeng, Yanqing, & Xie, Xinlan (2024). Green and efficient-oriented human-robot hybrid partial destructive disassembly line balancing problem from non-disassemblability of components and noise pollution. *Robotics and Computer-Integrated Manufacturing*, *90*, Article 102816.

Hezer, Seda, & Kara, Yakup (2015). A network-based shortest route model for parallel disassembly line balancing problem. *International Journal of Production Research*, *53*(6), 1849–1865.

Hu, Youxi, Liu, Chao, Zhang, Ming, Lu, Yuqian, Jia, Yu, & Xu, Yuchun (2024). An ontology and rule-based method for human–robot collaborative disassembly planning in smart remanufacturing. *Robotics and Computer-Integrated Manufacturing*, *89*, Article 102766.

Huang, Yufan, & Zhou, Binghai (2025). Deep-Q-network-enhanced aquila-equilibrium hyper-heuristic algorithm for preventive maintenance integrated disassembly line balancing involving worker redeployment. *Computers & Industrial Engineering*, *204*, Article 111113.

Ilgin, Mehmet Ali, Gupta, Surendra M., & Battaïa, Olga (2015). Use of MCDM techniques in environmentally conscious manufacturing and product recovery: State of the art. *Journal of Manufacturing Systems*, *37*, 746–758.

Kalaycilar, Eda Goksoy, Azizoğlu, Meral, & Batun, Sakine (2024). Disassembly line balancing with hazardous task failures–model based solution approaches. *Computers & Industrial Engineering*, *190*, Article 110089.

Kerin, Mairi, & Pham, Duc Truong (2020). Smart remanufacturing: a review and research framework. *Journal of Manufacturing Technology Management*, *31*(6), 1205–1235.

Ketzenberg, Michael E., Souza, Gilvan C., & Guide Jr, V. Daniel R. (2003). Mixed assembly and disassembly operations for remanufacturing. *Production and Operations Management*, *12*(3), 320–335.

Lee, Meng-Lun, Liang, Xiao, Hu, Boyi, Onel, Gulcan, Behdad, Sara, & Zheng, Minghui (2024). A review of prospects and opportunities in disassembly with human–robot collaboration. *Journal of Manufacturing Science and Engineering*, *146*(2), Article 020902.

Li, Jinlin, Chen, Xiaohong, Zhu, Zhanguo, Yang, Caijun, & Chu, Chengbin (2019). A branch, bound, and remember algorithm for the simple disassembly line balancing problem. *Computers & Operations Research*, *105*, 47–57.

Liang, Wei, Zhang, Zeqiang, Yin, Tao, Zhang, Yu, & Wu, Tengfei (2023). Modelling and optimisation of energy consumption and profit-oriented multi-parallel partial disassembly line balancing problem. *International Journal of Production Economics*, *262*, Article 108928.

Liu, Chunfeng, Wang, Deli, & Luo, Xinggang (2025). Optimization of new and remanufactured products in market segments via improved league championship approach. *Computers & Industrial Engineering*, Article 111263.

McGovern, Seamus M., & Gupta, Surendra M. (2007). A balancing method and genetic algorithm for disassembly line balancing. *European Journal of Operational Research*, *179*(3), 692–708.

Mete, Süleyman, Cil, Zeynel Abidin, Ağpak, Kürşad, Özceylan, Eren, & Dolgui, Alexandre (2016). A solution approach based on beam search algorithm for disassembly line balancing problem. *Journal of Manufacturing Systems*, *41*, 188–200.

Öksüz, Elif, Yılmaz, Ömer Faruk, Öksüz, Mehmet Kürşat, & Gürsoy Yılmaz, Beren (2025). Integrated multi-manned disassembly line balancing problem with reverse supply chain design strategies by considering lot sizing. *Journal of Industrial and Production Engineering*, *42*(2), 165–188.

Özceylan, Eren, Kalayci, Can B, Güngör, Aşkıner, & Gupta, Surendra M (2019). Disassembly line balancing problem: a review of the state of the art and future directions. *International Journal of Production Research*, *57*(15–16), 4805–4827.

Paksoy, Turan, Güngör, Aşkıner, Özceylan, Eren, & Hancilar, Arif (2013). Mixed model disassembly line balancing problem with fuzzy goals. *International Journal of Production Research*, *51*(20), 6082–6096.

Qin, Shujin, Li, Chong, Wang, Jiacun, Liu, Shixin, Zhao, Ziyan, Guo, Xiwang, et al. (2024). Multiobjective human–robot collaborative disassembly sequence planning: Considering the properties of components. *IEEE Systems, Man, and Cybernetics Magazine*, *10*(2), 15–23.

Qu, Weibin, Li, Jie, Zhang, Rong, Liu, Shimin, & Bao, Jinsong (2024). Adaptive planning of human–robot collaborative disassembly for end-of-life lithium-ion batteries based on digital twin. *Journal of Intelligent Manufacturing*, *35*(5), 2021–2043.

Riggs, Robert J., Battaïa, Olga, & Hu, S. Jack (2015). Disassembly line balancing under high variety of end of life states using a joint precedence graph approach. *Journal of Manufacturing Systems*, *37*, 638–648.

Rosenberg, Sonja, Huster, Sandra, Rudi, Andreas, & Schultmann, Frank (2025). Assessing economic uncertainty in dynamic reverse logistics networks–a stochastic modeling approach for planning circular battery treatment. *Computers & Industrial Engineering*, *201*, Article 110900.

Vahedi-Nouri, Behdin, Rohaninejad, Mohammad, Hanzálek, Zdeněk, & Foumani, Mehdi (2025). A batch production scheduling problem in a reconfigurable hybrid manufacturing-remanufacturing system. *Computers & Industrial Engineering*, *204*, Article 111099.

Wang, Kaipu, Guo, Jun, Du, Baigang, Li, Yibing, Tang, Hongtao, Li, Xinyu, et al. (2023). A novel MILP model and an improved genetic algorithm for disassembly line balancing and sequence planning with partial destructive mode. *Computers & Industrial Engineering*, *186*, Article 109704.

Wang, Kaipu, Li, Xinyu, & Gao, Liang (2019). A multi-objective discrete flower pollination algorithm for stochastic two-sided partial disassembly line balancing problem. *Computers & Industrial Engineering*, *130*, 634–649.

Wang, Kaipu, Li, Yibing, Guo, Jun, Gao, Liang, & Li, Xinyu (2024). Dynamic balancing of U-shaped robotic disassembly lines using an effective deep reinforcement learning approach. *IEEE Transactions on Industrial Informatics*, *20*(4), 6855–6865.

Wang, Dongyuan, Qiao, Fei, Guan, Liuen, Liu, Juan, Ding, Chen, & Shi, Jiaxuan (2024). Human–machine collaborative optimization method for dynamic worker allocation in aircraft final assembly lines. *Computers & Industrial Engineering*, *194*, Article 110370.

Wu, Tengfei, Zhang, Zeqiang, Guo, Lei, Song, Haoxuan, Xie, Xinlan, & Ren, Shiyi (2024). A hybrid evolutionary algorithm for the stochastic human–robot collaborative disassembly line balancing problem considering carbon emission optimization. *Engineering Applications of Artificial Intelligence*, *135*, Article 108703.

Wu, Tengfei, Zhang, Zeqiang, Zeng, Yanqing, & Zhang, Yu (2024). Mixed-integer programming model and hybrid local search genetic algorithm for human–robot collaborative disassembly line balancing problem. *International Journal of Production Research*, *62*(5), 1758–1782.

Wu, Tengfei, Zhang, Zeqiang, Zeng, Yanqing, Zhang, Yu, Guo, Lei, & Liu, Junqi (2024). Techno-economic and environmental benefits-oriented human–robot collaborative disassembly line balancing optimization in remanufacturing. *Robotics and Computer-Integrated Manufacturing*, *86*, Article 102650.

Xu, ZhenYu, & Han, Yong (2024). Two sided disassembly line balancing problem with rest time of works: A constraint programming model and an improved NSGA II algorithm. *Expert Systems with Applications*, *239*, Article 122323.

Yeni, Fatma Betul, Cevikcan, Emre, Yazici, Busra, & Yilmaz, Omer Faruk (2024). Aggregated planning to solve multi-product multi-period disassembly line balancing problem by considering multi-manned stations: A generic optimization model and solution algorithms. *Computers & Industrial Engineering*, *196*, Article 110464.

Yılmaz, Ömer Faruk, et al. (2022). Tactical level strategies for multi-objective disassembly line balancing problem with multi-manned stations: an optimization model and solution approaches. *Annals of Operations Research*, *319*(2), 1793–1843.

Yin, Tao, Zhang, Zeqiang, Liang, Wei, Zeng, Yanqing, & Zhang, Yu (2023). Multi-man–robot disassembly line balancing optimization by mixed-integer programming and problem-oriented group evolutionary algorithm. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *54*(3), 1363–1375.

Yin, Tao, Zhang, Zeqiang, Zhang, Yu, Wu, Tengfei, & Liang, Wei (2022). Mixed-integer programming model and hybrid driving algorithm for multi-product partial disassembly line balancing problem with multi-robot workstations. *Robotics and Computer-Integrated Manufacturing*, *73*, Article 102251.

Zheng, Feifeng, He, Junkai, Chu, Feng, & Liu, Ming (2018). A new distribution-free model for disassembly line balancing problem with stochastic task processing times. *International Journal of Production Research*, *56*(24), 7341–7353.

Zhi, Jixuan, & Lien, Jyh-Ming (2025). Improving human-robot collaboration via computational design. *IEEE Robotics and Automation Letters*, *10*(2), 1074–1081. http://dx.doi.org/10.1109/LRA.2024.3519863.

Zhou, Zhuo, Xu, Liyun, Chen, Yiyang, Liao, Liqiang, & Xu, Zhun (2025). Digital-twin-based AGV cluster dynamic scheduling for solar cell production workshop using deep reinforcement learning. *Neurocomputing*, Article 130772.

Zhu, Lixia, Chen, Yarong, & Mumtaz, Jabir (2024). Multi-objective human-robot collaborative disassembly line balancing considering components remanufacture demand and hazard characteristics. *Computers & Industrial Engineering*, *197*, Article 110621.

Zhu, Lixia, Zhang, Zeqiang, & Guan, Chao (2020). Multi-objective partial parallel disassembly line balancing problem using hybrid group neighbourhood search algorithm. *Journal of Manufacturing Systems*, *56*, 252–269.