Dynamic Disassembly Planning of End-of-Life Products for Human–Robot Collaboration Enabled by Multi-Agent Deep Reinforcement Learning

Yiqun Peng, Weidong Li, *Senior Member, IEEE*, Yong Zhou, and Duc Truong Pham

***Abstract*—Disassembly is a critical step in the remanufacturing of end-of-life products. High labor costs and the limited ability of robots to perform intricate disassembly tasks have led to the increasing use of human-robot collaboration (HRC) for disassembly. This paper addresses a challenge in HRC-based disassembly, i.e., the inherent human uncertainty during disassembly. The uncertainty is that the disassembly time for a task and the task sequence selection by a human during execution might differ from the pre-defined disassembly plan so that dynamic disassembly planning for subsequent tasks is necessary. Stackelberg equilibrium-enabled disassembly task assignment policies are designed to meet the above purpose efficiently and safely. The human leader's policy is to choose tasks that maximize the efficiency-related return value based on the robot's optimal response to the human's choice. As the follower, the robot selects tasks that maximize the safety-related return value for each human task choice. To identify the optimal values of the policies to ensure the safety and efficiency of the entire HRC-based disassembly process, an improved multi-agent proximal policy optimization (i-MAPPO) algorithm is designed. Finally, a case study for disassembling an electric vehicle battery is used to verify that the proposed approach can adapt to human uncertainty with a high success rate while ensuring that the disassembly time remains short and the human-robot distance remains within the safety threshold throughout the disassembly process.**

***Note to Practitioners*—HRC-based disassembly approaches can enhance the flexibility of disassembly lines. However, a significant challenge in HRC-based disassembly planning is the inherent uncertainty of human behavior, including instability in task completion times and unpredictability in task selection. To address this issue, this study combines the Stackelberg game theory with a multi-agent deep reinforcement learning optimization algorithm to design a dynamic disassembly task assignment approach. This approach can adapt to human uncertainties and quickly provide task assignment solutions based on changes in the disassembly scenario. Using the disassembly of retired electric vehicle batteries as a representative case, this approach offers a feasible solution for safe and efficient disassembly processes.**

***Index Terms*—Human-robot collaboration (HRC), disassembly planning, Stackelberg model.**

I. Introduction

TO MINIMIZE negative environmental impacts, it is critical to develop effective remanufacturing strategies to process end-of-life (EoL) products after their service life [1], [2]. Disassembly is essential to facilitate remanufacturing tasks. Manual and robotic disassembly methods have been developed for remanufacturing [2]. However, neither manual nor robotic disassembly can meet practical requirements. The former faces the challenges of high labor costs and human safety issues when separating hazardous parts from EoL products. Although it can overcome the above difficulties, robotic disassembly struggles with complicated tasks, such as dismantling welded, glued, rusted, or broken parts. Instead, the recently developed human-robot collaboration (HRC)-based disassembly approach combines human flexibility and robot strength/repeatability. As a result, it has emerged as a more practical solution for disassembly [1].

Disassembly planning is a critical step in the HRC-based disassembly process. In an ideal scenario, the human and the robot in HRC carry out the assigned disassembly tasks according to the pre-planned timing and sequence to achieve the overall goals in terms of safety and efficiency. Safety means that the human drives HRC (i.e., human centricity in HRC) and that the robot maintains a sufficient safe distance from the human during disassembly. Efficiency means that the disassembly process can be carried out in the shortest amount of time. However, the uncertain nature of humans poses a challenge to the safety and efficiency criteria during the disassembly process [3]. A human operator may randomly or wrongly select a disassembly task and exhibit erratic disassembly time, thereby introducing uncertainty and issues that can disrupt and invalidate pre-planned disassembly tasks [4], [5]. Human uncertainty has received less attention in existing studies on HRC-based disassembly planning.

Received 26 August 2024; revised 3 January 2025 and 13 March 2025; accepted 30 March 2025. Date of publication 2 April 2025; date of current version 24 April 2025. This article was recommended for publication by Associate Editor B. Lacevic and Editor P. Rocco upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Project 51975444, in part by the Ministry of Science and Technology of China under Project G2022013009, in part by the Science and Technology Commission of Shanghai Municipality under Project 23010503700, and in part by China Scholarships Council under Project 202206950015. *(Corresponding author: Weidong Li.)*

Yiqun Peng was with the School of Transportation and Logistics Engineering, Wuhan University of Technology, Wuhan 430070, China. He is now with the Jianghuai Advance Technology Center, Hefei 230000, China (e-mail: yqpeng@whut.edu.cn).

Weidong Li is with the School of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China (e-mail: weidongli@usst.edu.cn).

Yong Zhou is with the School of Transportation and Logistics Engineering, Wuhan University of Technology, Wuhan 430070, China (e-mail: zhouyong@whut.edu.cn).

Duc Truong Pham is with the Department of Mechanical Engineering, University of Birmingham, B15 2TT Birmingham, U.K. (e-mail: d.t.pham@bham.ac.uk).

Digital Object Identifier 10.1109/TASE.2025.3557155

The Stackelberg model is a game theoretic model that includes a leader-follower role [6]. In such a model, the leader chooses its initial actions according to the goal of tasks; the follower decides its actions in response to the leader's actions. Finally, both parties adjust their actions based on the best targets of both parties. The Stackelberg model can be used to support HRC where the human leads collaborations and the robot is the follower. However, the determination of return values in the model is influenced by the nature of ever-changing HRC-based disassembly scenarios and the uncertainty in human disassembly time use/disassembly task selection, complicating the identification of the model's optimal solution.

This paper presents a Stackelberg model-based dynamic disassembly task assignment approach optimized using an improved multi-agent proximal policy optimization (i-MAPPO) algorithm. The innovations and procedures of the approach are described below:

(i) HRC-based dynamic disassembly task assignment is innovatively modeled using the Stackelberg model to facilitate human-centric HRC. The uncertainties in human disassembly time and human disassembly task selection arising from ever-changing HRC scenarios are modeled as a normally distributed variable and an ε-greedy strategy to represent the nature of HRC-based dynamic disassembly processes;

(ii) The i-MAPPO algorithm is novel because it integrates the Stackelberg equilibrium into the MAPPO advantage function to address Stackelberg policies in dynamic HRC scenarios effectively. The human leader's policy in HRC is to choose tasks that maximize the efficiency-related return value based on the robot's optimal response to the human's choice. The robot's policy as the follower is to select tasks that maximize the safety-related return value for each human task choice. The i-MAPPO algorithm can identify the optimal policies under dynamic scenarios to ensure safe and efficient HRC-based disassembly processes. Extensive experiments demonstrate that i-MAPPO outperforms other advanced heuristic and reinforcement learning algorithms;

(iii) The approach is validated by using the disassembly of electric vehicle batteries as a case study. The experiments demonstrate that the optimal policies achieved by the proposed approach can minimize the disassembly time and maximize the human-robot distance to ensure HRC safety throughout HRC-based dynamic disassembly processes.

The remainder of this paper is organized as follows: Section II introduces the relevant research. Section III elucidates the research methodology. In Section IV, the Stackelberg model is detailed. Section V describes the i-MAPPO algorithm. In Section VI, the experimental results are presented, accompanied by discussions. Finally, in Section VII, key findings are summarized, and conclusions are drawn.

II. Review of Related Research

The related literature includes HRC-based disassembly pre-planning and dynamic disassembly planning with uncertainty factors [7]. The technical details of the reviewed approaches are summarized below (they are also compared in TABLE I).

TABLE I
THE SUMMARY OF RELATED WORKS

Ref.	Planning	Method	Disassembly uncertainty
[8]	Pre-planning	Discrete Bees	-
[9]	Pre-planning	Gurobi	-
[13]	Dynamic planning	Multi-agent Q-learning	-
[14]	Dynamic planning	QMIX	-
[10]	Dynamic planning	Frog leaping algorithm	Time
[11]	Dynamic planning	Deep reinforcement learning	Failure rates, operation skipping rates, residual value of parts
[12]	Dynamic planning	LSTM and Bayesian approximation	Operator's actions

A. HRC-Based Disassembly Pre-Planning

The pre-planning approaches can determine the disassembly sequence of the human and the robot in HRC before task execution. Heuristic or evolutionary algorithms have been applied to facilitate the research. Xu et al. [8] developed a Pareto-based discrete Bees algorithm for HRC-based disassembly planning optimization. Lee et al. [9] designed a Gurobi solver for HRC disassembly planning by considering resource constraints and human safety. However, the adaptability of the pre-planning approaches to dynamic disassembly conditions is limited as the optimization algorithms are inefficient in generating disassembly re-plans.

B. HRC-Based Dynamic Disassembly Planning

Various uncertainties lead to dynamic changes in disassembly planning [10], [11], [12]. Guo et al. [10] modeled the human disassembly time as a variable following a Gaussian distribution. The Monte Carlo method was developed to estimate the human disassembly time when optimizing the overall profit and energy consumption. Han et al. [11] modeled the uncertainties of disassembly failure rates, operation rates, and the residual value of disassembled parts. Liu et al. [12] employed the LSTM network and Bayesian approximation to model and quantify the uncertainty of human actions during disassembly processes.

With the uncertainty modeling, HRC-based dynamic disassembly planning approaches have been developed. Multi-agent reinforcement learning (MARL) algorithms, such as multi-agent Q-learning (MAQ-learning), QMIX, and MAPPO, enable real-time task adjustments for humans and robots to facilitate HRC-based dynamic disassembly planning. Based on an MAQ-learning algorithm, a dynamic disassembly planning approach was designed to optimize the task allocation and operational efficiency of HRC-based EoL battery disassembly [13]. Gao et al. [14] developed a multi-agent optimization approach based on partially observable QMIX to facilitate HRC-based disassembly. Compared with Q-learning and

QMIX, MAPPO can model complex nonlinear disassembly conditions more efficiently by adjusting the action policies of the human and robot in real time. On the other hand, the action policies of MAPPO are independently assigned to the human and the robot without considering the influence of the human policy on robotic behaviors, thereby failing to ensure human safety. The Stackelberg model ensures the efficiency and safety of HRC, where the human leads decision-making and the robot acts based on the human's behaviors [6]. Ramachandruni et al. [15] developed a user-aware hierarchical task planning framework to implement leader-follower-based HRC disassembly, and the optimization objectives were the minimization of task completion time and the human's cognitive workload. Zhou et al. [6] modeled the leader-follower HRC disassembly paradigm using the Stackelberg model, and the optimization objectives were the minimization of disassembly time and the maximization of the average distance between the human and the robot for safety assurance. Although the Stackelberg model addresses the action-independent outputs of MAPPO for multi-agent optimization, the lack of learning capabilities limits its adaptability to dynamic HRC disassembly. It can be observed that it is helpful to integrate the strengths of the Stackelberg model and MAPPO to achieve HRC-based dynamic disassembly planning. The Stackelberg model ensures human-centric HRC, MAPPO handles the uncertainties of HRC disassembly, and the integrated solution achieves safety and operation efficiency optimization during dynamic disassembly processes.

III. Overall Research Methodology

This study considers a scenario where a human and a robot work together to disassemble EoL products in a disassembly cell. The human is the uncertain factor in the disassembly process. This manifests in the fact that humans may not choose to perform tasks as planned. Uncertainty in the human's disassembly time will also lead to time deviations from fixed planning schemes during implementation. The uncertainty persists throughout the disassembly process, making replanning new schemes for subsequent tasks costly and inefficient whenever deviations have occurred.

This paper proposes an innovative HRC-based dynamic disassembly planning approach to address this problem. The approach can continuously assign disassembly tasks to both humans and robots as the disassembly task scenario changes. The implementation details are given below.

Product disassembly includes typical tasks such as removing fasteners (screws, snap connectors and pins). Disassembly planning requires considering the product's disassembly priority. In addition, disassembly tasks possess different characteristics that determine their suitability for the human or the robot during HRC. In this study, disassembly tasks are classified into the following four categories: Disassembly tasks that can be performed only by a human are denoted as H; disassembly tasks that can be performed only by the robot are denoted as R; disassembly tasks that the human or the robot can perform are denoted as H/R; disassembly tasks in which the human and the robot need to work together are denoted as H+R.

Disassembly feasibility for humans (*DFH*), disassembly feasibility for robots (*DFR*), and disassembly feasibility for HRC (*DFHRC*) are established to constrain disassembly task assignment. *DFH* considers two attributes of each disassembly task, i.e., workload Att_1(weight and volume of the disassembled product) and disassembly hazard Att_2 (toxic liquid/gas leakage risks during the disassembly process). *DFR* considers three attributes, i.e., disassembly accessibility Att_3 (extent and size of the disassembly area), disassembly positioning accuracy Att_4, and disassembly complexity Att_5 (number of actions required for robotic disassembly). *DFHRC* considers one attribute, i.e., collaborative necessity Att_6 Each attribute is evaluated using an expert scoring method. The evaluation is divided into five levels: 0, 0.25, 0.5, 0.75, and 1. The values of DFH, DFR, and *DFHRC* are calculated below.

$$\begin{cases} DFH = min\,(Att_1, Att_2) \\ DFR = min\,(Att_3, Att_4, Att_5) \\ DFHRC = Att_6 \end{cases} \tag{1}$$

Disassembly task assignment considers the following conditions: (i) If *DFH* and *DFR* are both greater than 0, the task belongs to H/R. (ii) If *DFH* is 0 and *DFR* is greater than 0, the task belongs to R. (iii) If *DFR* is equal to 0 and *DFH* is greater than 0, the task belongs to H. (iv) If *DFR* and *DFH* are 0, then for human safety, the task belongs to R. (v) If *DFH* and *DFHRC* are greater than 0, the task belongs to H+R.

During the disassembly process, dynamic task assignment effectively manages unpredictability due to human uncertainty. This study needs to address how to assign tasks to both the human and the robot at each time, ensuring that the entire disassembly process is both safe and efficient.

To address this, in this paper, disassembly task assignment policies for the human and the robot are designed based on the Stackelberg model. The policies satisfy the Stackelberg equilibrium with the human as the leader and the robot as the follower. This is depicted as follows: at time point t, under each task chosen by the human, the robot is assigned the task with the highest safety-related return value; based on the robot's response to the human's task choice, the human is assigned the task with the highest efficiency-related return value. In this way, the optimal policies pursue both safety and efficiency in disassembly while adapting to uncertain human task selection and execution through the robot's response to human task selection. The policies at different time points influence the efficiency-related or safety-related return values for the human and the robot at those time points, making it challenging to solve the optimal policies directly. Therefore, in this study, the i-MAPPO algorithm is designed to solve for the optimal policies iteratively. The entire process is shown in Fig. 1. The details of each part are described in the following sections.

IV. Disassembly Task Assignment Policies Based on the Stackelberg Model

A. Form of the Policies

Since disassembly scenarios differ at different time points t, the description of disassembly scenarios is used to represent the disassembly state, denoted as s_t. s_t needs to describe not

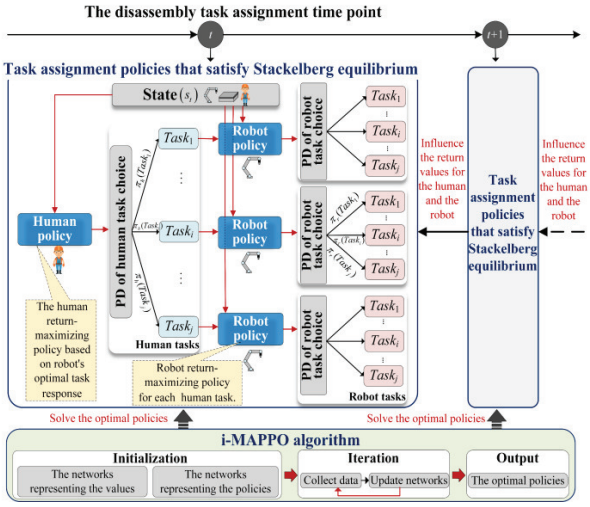


Fig. 1. The overall process of the proposed approach, where PD stands for probability distribution.

only the relationships between the human, the robot, and the EoL product at time point t but also the disassembly tasks that the human and the robot can perform at time t.

Therefore, s_t includes the description of the priority relationships among the disassembly tasks of the product at time point t (identifying which disassembly tasks are not interfered with by other tasks) and the execution status of the disassembly tasks by the human and the robot (identifying which disassembly tasks have been executed and refining the disassembly tasks that can be executed), as described below.

The priority relationship of the disassembly tasks is represented by the disassembly task priority matrix $\boldsymbol{M_t}$ with n rows and n columns, where n is the number of all the disassembly tasks in the product. The element of the i^{th} row and j^{th} column in $\boldsymbol{M_t}$ is calculated as follows:

$$\boldsymbol{M_t}(i,j) = \begin{cases} 1, & Task_j \text{ can be executed after } Task_i \text{ at time } t \\ 0, & \text{else} \end{cases} \tag{2}$$

where $Task_i$ and $Task_j$ represent the i^{th} and j^{th} disassembly tasks, respectively. However, analyzing $\boldsymbol{M_t}$ alone does not indicate which disassembly tasks have been completed.

To address this, the description of the execution status of disassembly tasks by the human and the robot involves whether the disassembly tasks have been performed by the human or the robot and the degree of completion. This allows the determination of which disassembly tasks have been executed and whether the human and the robot need to be assigned new disassembly tasks. It is represented by two vectors: $\boldsymbol{S_t^h}$, which is used to represent the relationship between the human and the disassembly tasks at time t, and $\boldsymbol{S_t^r}$, which is used to represent the relationship between the robot and the disassembly tasks at time t. $\boldsymbol{S_t^h} = [h_1, h_2, \ldots, h_i, \ldots, h_n]$, where n is the number of all the disassembly tasks in the product, and $0 \le h_i \le 1$. If $h_i = 0$, the human has not yet performed $Task_i$. If $h_i = 1$, it means that the human has performed $Task_i$. If $0 < h_i < 1$, it represents that the human is performing $Task_i$, and the value of h_i represents the completion degree of the task. Similarly, $\boldsymbol{S_t^r} = [c_1, c_2, \ldots, c_j, \ldots, c_n]$, where the value of c_j is defined as that for h_i. In summary, s_t can be represented as a matrix with $n+2$ rows and n columns, comprising $\boldsymbol{M_t}$ in n rows and n columns, followed by $\boldsymbol{S_t^h}$ in 1 row and n columns and $\boldsymbol{S_t^r}$ in 1 row and n columns. It is represented below:

$$s_t = \left\{\boldsymbol{M_t}, \boldsymbol{S_t^h}, \boldsymbol{S_t^r}\right\}_{(n+2)*n} \tag{3}$$

The above s_t is represented in the form of a matrix, which facilitates subsequent reinforcement learning processing.

Based on the description of s_t, the human's policy uses the state s_t as the input and outputs the human task selection. In contrast, the robot's policy, which is designed to balance the uncertain choices of the human, takes the state s_t and the human's task choices as the inputs and outputs the robot task selection. Notably, the range of task selections for both the human and the robot includes not only executable disassembly tasks but also the option to not perform any task, denoted as *wait* (due to disassembly constraints, in some states s_t, there may be no tasks available for the human or the robot to execute).

Let the human task selection be denoted as a_t^h, the robot task selection as a_t^r, the human's policy as π_h, and the robot's policy as π_r. The specific forms of π_h and π_r are shown in (4) and (5):

$$a_t^h \sim \pi_h(\cdot \mid s_t) \tag{4}$$

$$a_t^r \sim \pi_r\left(\cdot \mid s_t, a_t^h\right) \tag{5}$$

$\pi_h(\cdot \mid s_t)$ represents the human's policy, which takes s_t as input and outputs the probability distribution of a_t^h. For example, if the human has three choices of tasks at time point t and each choice has an equal probability of being selected, then $\pi_h(\cdot \mid s_t) = \left[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right]$. The purpose of this arrangement is twofold: on the one hand, it provides more solutions for disassembly task assignment to address human uncertainty; on the other hand, it offers a larger solution set range for the subsequent process to identify the optimal value of the policy (which outputs a unique task selection). Similarly, $\pi_r\left(\cdot \mid s_t, a_t^h\right)$ takes s_t and a_t^h as inputs and outputs the probability distribution of a_t^r.

B. Return Values of the Human and the Robot

To optimize the policies of the human and the robot, it is necessary to evaluate the quality of task selection by both the human and the robot at time point t. Specifically, if a task is performed with higher quality, the policies should increase the probability of that task being selected. After the human and the robot perform tasks a_t^h and a_t^r at time point t, no new tasks are selected until the next time point $t+1$. During the interval from t to $t+1$, both the human and the robot receive a value to evaluate their choices, called the reward value. The reward value received by the human is denoted as r_t^h, whereas the reward value received by the robot is denoted as r_t^r. The following sections specifically introduce r_t^h and r_t^r. In the disassembly process based on HRC, safety is the priority. According to the standards of ISO-15066 [16], maintaining a safe distance between the human and the robot during the HRC-based disassembly process is inevitable. Since the

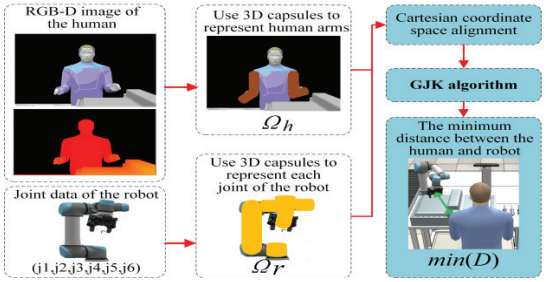


Fig. 2. Minimum distances between the human and the robot.

robot makes decisions based on the human's choice, the robot should assume the role of actively maintaining the human-robot distance. Therefore, at time point t, after the human and the robot execute a_t^h and a_t^r, the reward value r_t^r obtained by the robot by time point $t+1$ is determined by the distance between the human and the robot during this period, as shown below:

$$r_t^r(s_t, a_t^h, a_t^r) = \begin{cases} 0, & if\, a_t^h = wait or a_t^r = wait \\ \dfrac{D\left(s_t, a_t^h, a_t^r\right)}{w_r}, & else \end{cases} \tag{6}$$

where $D\left(s_t, a_t^h, a_t^r\right)$ represents the minimum distance between the human and the robot while they perform tasks $\left(a_t^h, a_t^r\right)$ in s_t and where w_r is a coefficient that is used for normalization. The greater the distance is, the higher r_t^r. When calculating the distance between the human and the robot, this study assumes that the human does not arbitrarily change his/her position. This study utilizes the Gilbert-Johnson-Keerthi (GJK) algorithm [17] to compute the minimum distance between the human and the robot. As shown in Fig. 2, the human arm skeletons are captured via an RGB-D camera and modeled as two 3D capsule-based convex hulls $\Omega_h = \{\omega_{h-1}, \omega_{h-2}\}$, where ω_{h-1} represents the convex hull of a single arm. Concurrently, the joint angle data of the robot are acquired, and each robot link is represented by a convex hull based on 3D capsules, denoted as $\Omega_r = \left\{\omega_{r-j} \,|\, 1 \le j \le j_n\right\}$, where j_n is the total number of robotic links, and ω_{r-j} represents the convex hull of a single link. After coordinate system alignment, the GJK algorithm iteratively computes minimum distances between each convex hull in Ω_h and Ω_r to form a set $D = \left\{d_{ij}\right\}$, where d_{ij} represents the minimum distance between ω_{h-i} and ω_{r-j}. The core principle is to construct Minkowski difference sets as follows:

$$S_{ij} = \omega_{h-i} \ominus \omega_{r-j} = \{a - b | a \in \omega_{h-i}, b \in \omega_{r-j} \tag{7}$$

Based on the above, the problem is transformed into finding the point in S_{ij} that is the closest to the origin. If such a point exists, the convex hulls intersect, and d_{ij} is 0; otherwise, d_{ij} corresponds to the distance from this point to the origin. The GJK algorithm approximates the closest point by iteratively updating a simplex. The detailed implementation of the algorithm is available in [17]. Finally, the minimum value in D, i.e., $min\,(D)$, represents the minimum distance between the human and the robot during HRC.

wait indicates that no task is to be performed. r_t^r is set to 0 when either the human or the robot chooses *wait* to promote more collaborative disassembly between the human and the robot during HRC.

In the disassembly process based on HRC, improving the overall efficiency of disassembly is also crucial. Since the human acts as the leader, it assumes that the human plays the role of regulating the pace of the entire disassembly process, striving for rapid completion of the process. Therefore, at time point t, after the human and the robot execute a_t^h and a_t^r, the reward value r_t^h obtained by the human by time point $t+1$ is determined by the disassembly time, as shown below.

$$r_t^h(s_t, a_t^h, a_t^r) = -t\left(s_t, a_t^h, a_t^r\right)/w_h \tag{8}$$

where $t\left(s_t, a_t^h, a_t^r\right)$ represents the time spent from time point t to the next time point $t+1$ after performing task $\left(a_t^h, a_t^r\right)$ in s_t and where w_h is a coefficient that is used for normalisation. The shorter the disassembly time is, the larger r_t^h is.

To solve $t\left(s_t, a_t^h, a_t^r\right)$, the transition relationship from state s_t to state s_{t+1} needs to be analysed. It is assumed that in state s_t, both the human and the robot need to be assigned new tasks. If $(a_t^h, a_t^r) = (Task_i, Task_j)$, s_{t+1} can be deduced by comparing the disassembly time required for the human and the robot to perform the corresponding tasks. The disassembly time for the human to perform $Task_i$ is assumed to be t_i^h, and the disassembly time for the robot to perform $Task_j$ is denoted as t_j^r. Considering the uncertainty of the disassembly time of the human, t_i^h is set to follow the normal distribution:

$$t_i^h \sim N\left(\mu_i, \sigma_i^2\right) \tag{9}$$

where μ_i is the average time for the human to complete $Task_i$ and where σ_i^2 is the corresponding variance.

If $t_i^h < t_j^r$, it signifies that after time t_i^h from s_t, the human will complete the task $Task_i$ and require the assignment of a new task, while the robot is still engaged in $Task_j$. This also means that after t_i^h from s_t, the process transits to the next task assignment time point $t+1$, and the state transits to s_{t+1}. In s_{t+1}, the i^{th} row of $\boldsymbol{M_{t+1}}$ is updated to 0, and the i^{th} element of $\boldsymbol{S_{t+1}^h}$ is modified to 1. Furthermore, the j^{th} element of $\boldsymbol{S_{t+1}^r}$ is updated to t_i^h/t_j^r, indicating the completion degree of $Task_j$. Specifically, Fig. 3 also demonstrates all the transition rules under the different types of states.

However, after the human and the robot select tasks a_t^h and a_t^r at time point t, they make new task choices at time point $t+1$ according to their respective policies π_h and π_r. They also receive new rewards, and this process continues until the disassembly ends. Therefore, evaluating the quality of a_t^h and a_t^r solely based on r_t^h and r_t^r is inappropriate. Additionally, according to the transition rules between the states, all the reward data after time point t can be obtained. Based on this, the return values for the human and the robot are defined in this paper to holistically evaluate the qualities of the task choices made by the human and the robot. The return value refers to the sum of the human's or the robot's reward (r_t^h or r_t^r) and the discounted expected cumulative rewards, which are shown in (10) and (11):

$$Q_h\left(s_t, a_t^h, a_t^r\right) = r_t^h + \gamma \times E_{(\pi_h, \pi_r, t_i^h)}\left[\sum_{l=0}^{T-t-1}\left(\gamma^l \times r_{t+1+l}^h\right)\right] \tag{10}$$

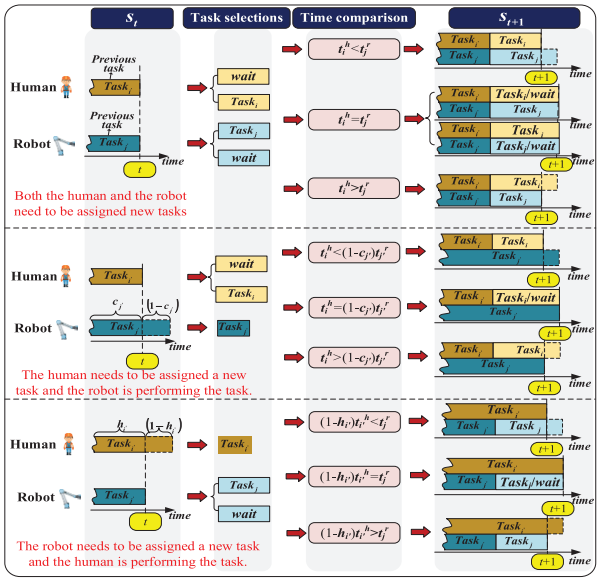


Fig. 3. All the transition rules under different types of states.

$$Q_r\left(s_t, a_t^h, a_t^r\right) = r_t^r + \gamma \times E_{(\pi_h, \pi_r, t_i^h)}\left[\sum_{l=0}^{T-t-1}\left(\gamma^l \times r_{t+1+l}^r\right)\right] \tag{11}$$

where $Q_h\left(s_t, a_t^h, a_t^r\right)$ and $Q_r\left(s_t, a_t^h, a_t^r\right)$ represent the return values for the human and the robot, respectively. γ is the discount factor to balance future rewards (the larger γ is, the greater the impact of future rewards on quality evaluations of a_t^h and a_t^r). E represents the expectation operation used to calculate the expected cumulative rewards (since π_h and π_r are stochastic policies, the human disassembly time t_i^h satisfies the normal probability distribution). T denotes the total number of assignment time points in the complete disassembly process.

C. Policies That Satisfy Stackelberg Equilibrium

The ranges of disassembly tasks that the human and the robot can perform under s_t are defined as $\boldsymbol{A_t^h}$ and $\boldsymbol{A_t^r}$, respectively. Based on the aforementioned definitions, in this paper, the policies that satisfy the Stackelberg equilibrium are designed to ensure that both the human and robot obtain balanced optimal expected return values, as shown in equations (12) and (13):

$$\begin{aligned}&\sum\nolimits_{a_t^h \in \boldsymbol{A_t^h}} \left\{\pi_h^*\left(a_t^h \mid s_t\right) \times \left[\sum\nolimits_{a_t^r \in \boldsymbol{A_t^r}} \left(\pi_r^*\left(a_t^r \mid s_t, a_t^h\right) \times Q_h^*\left(s_t, a_t^h, a_t^r\right)\right)\right]\right\} \\ &\geq \sum\nolimits_{a_t^h \in \boldsymbol{A_t^h}} \left\{\pi_h\left(a_t^h \mid s_t\right) \times \left[\sum\nolimits_{a_t^r \in \boldsymbol{A_t^r}} \left(\pi_r^*\left(a_t^r \mid s_t, a_t^h\right) \times Q_h^*\left(s_t, a_t^h, a_t^r\right)\right)\right]\right\}\end{aligned} \tag{12}$$

$$\begin{aligned}&\sum\nolimits_{a_t^r \in \boldsymbol{A_t^r}} \left[\pi_r^*\left(a_t^r \mid s_t, a_t^h\right) \times Q_r^*\left(s_t, a_t^h, a_t^r\right)\right] \\ &\geq \sum\nolimits_{a_t^r \in \boldsymbol{A_t^r}} \left[\pi_r\left(a_t^r \mid s_t, a_t^h\right) \times Q_r^*\left(s_t, a_t^h, a_t^r\right)\right]\end{aligned} \tag{13}$$

where $\pi_h^*\left(\cdot \mid s_t\right)$ and $\pi_r^*\left(\cdot \mid s_t, a_t^h\right)$ represent the human and robot policies, respectively, that satisfy Stackelberg equilibrium. $\pi_h^*\left(a_t^h \mid s_t\right)$ represents the probability that the human chooses a_t^h under s_t; $\pi_r^*\left(a_t^r \mid s_t, a_t^h\right)$ represents the probability that the robot chooses a_t^r under s_t and a_t^h; and $Q_h^*\left(s_t, a_t^h, a_t^r\right)$ and $Q_r^*\left(s_t, a_t^h, a_t^r\right)$ represent the return values obtained by the human and the robot when they perform tasks a_t^h and a_t^r in s_t and execute the policies that satisfy Stackelberg equilibrium after time point t, respectively. The expected return value refers to the sum obtained by multiplying the probability of each human-robot task choice by the corresponding return value and then adding them together.

Equation (12) indicates that the human's policy is to select the task that maximizes the human's expected return value based on the robot's response to the human's choice (ensuring efficiency). Similarly, (13) illustrates that the robot's policy is to select the task that maximizes the robot's expected return value for each human task (ensuring safety). In this way, through the robot's optimal response to the human choice, if the human's task selection becomes unreasonable (e.g., affecting safety), the robot can promptly adjust its decisions to adapt to the human's uncertain task choices. Additionally, when calculating the return values for the human and the robot, the expectation operation considers the human disassembly time that follows a normal probability distribution, ensuring that the task selections of both the human and the robot can adapt to the uncertain disassembly time of the human.

However, $Q_h^*\left(s_t, a_t^h, a_t^r\right)$ and $Q_r^*\left(s_t, a_t^h, a_t^r\right)$ in (12) and (13) are affected by the policies adopted after time point t. Once the subsequent policies are updated, $\pi_h^*\left(\cdot \mid s_t\right)$ and $\pi_r^*\left(\cdot \mid s_t, a_t^h\right)$ should also be updated. Therefore, the process of solving for the optimal policies that satisfy the Stackelberg equilibrium at time point t is dynamic. This paper designs the i-MAPPO algorithm to iteratively solve the optimal policies, which will be specifically described in Section V.

V. Solving Optimal Policies With the i-MAPPO Algorithm

The MAPPO algorithm is an effective algorithm that can solve optimal task decision-making for multiple agents [18]. Its continuous interaction with and ability to learn from the environment enables the agents to adapt to different scenarios. However, the optimal policies derived from the MAPPO algorithm conform to the Nash equilibrium, fail to distinguish roles among agents and do not align with the policies that satisfy the Stackelberg equilibrium proposed in this study. Therefore, the algorithm is improved as the i-MAPPO algorithm to solve for the optimal policies, as shown below.

A. Network Building

To enhance generalization, four deep networks are built to represent the policy and the return values of the human and the robot. The human's policy $\pi_h\left(\cdot \mid s_t; \beta_h\right)$ is known as the actor network of the human, where β_h represents the parameters of the network. When the number of disassembly tasks is n, the input is s_t with a size of $(n+2) * n$. Since *wait* needs to be considered in the task assignment process, the output is the human task choice probability distribution with a size of $n+1$. To prevent the occurrence of choosing unavailable

tasks, an unavailable task mask layer is added to the output of the hidden layer. It can assign an extremely small value to the output neuron corresponding to the nonexecutable task to minimize the probability value of that task [19]. The robot's policy $\pi_r\left(\cdot \mid s_t, a_t^h; \beta_r\right)$ is known as the actor network of the robot, where β_r represents the parameters of the network. The design of the network is mostly the same as that of the actor network of the human, except that the input includes human tasks a_t^h and s_t and the output is the robot task choice probability distribution. The return values $Q_h\left(s_t, a_t^h, a_t^r; \varphi_h\right)$ and $Q_r\left(s_t, a_t^h, a_t^r; \varphi_r\right)$ of the human and the robot are known as the critic networks of the human and the robot respectively, where φ_h and φ_r are the parameters of the corresponding networks. Their inputs include s_t, the human task choice a_t^h, and the robot task choice a_t^r. After passing through the hidden layers, the human's critic network outputs the human's return value, and the robot's critic network outputs the robot's return value, both with a size of 1. The above network will be continuously updated until convergence to obtain the optimal policies. The key working principles are as follows.

B. Network Update Process

In each iteration, the update direction of the actor network aims to achieve a policy that satisfies the Stackelberg equilibrium. The pre-update actor networks of the human and the robot are used to collect complete task assignment process data to evaluate the performance of the pre-update policies, and the network is subsequently updated based on this evaluation.

The collected data are denoted as $\left\{s_t, a_t^h, a_t^r, r_t^h, r_t^r, s_{t+1}\right\}_{t \le T_i}^{i}$ (where T_i is the number of task assignment time points in iteration i). It is important to note that, to improve adaptability to human uncertainty in task selection, an ε-greedy strategy [20] is used by the human to choose tasks. The detail of the process is below:

$$a_t^h \sim \begin{cases} \pi_h\left(\cdot \mid s_t; \beta_h\right), & if\ rand() < \varepsilon \\ U\left(\cdot \mid s_t\right), & otherwise \end{cases} \tag{14}$$

where ε is a constant within the range [0, 1], and $rand()$ represents a random number uniformly distributed within the same range. When $rand() < \varepsilon$, the human selects a task based on the policy determined by the human's actor network; otherwise, the human selects a task randomly, and the selection follows a uniform probability distribution.

Based on the above data, the advantage of human/robot task selection a_t^h/a_t^r, denoted as Ad_t^h/Ad_t^r, is defined to evaluate the performance of the task selections.

Ad_t^r is defined as the difference between the robot's return value for executing a_t^r and the expected return value for executing the pre-update robot policy with the human task selection a_t^h fixed. If Ad_t^r is greater than 0, selecting a_t^r can yield a higher expected return for the robot. Therefore, during the update of the robot's actor network, the probability of a_t^r being selected should be increased. Conversely, if Ad_t^r is less than 0, the probability should decrease. The calculation formula for Ad_t^r is as follows:

$$\begin{aligned} Ad_t^r = & \sum_{l=0}^{T_i - t}\left(\gamma^l \times r_{t+l}^r\right) \\ & - \sum_{a_t^{-r} \in \boldsymbol{A}_t^r} \pi_r\left(a_t^{-r} \mid s_t, a_t^h; \beta_{r,i}\right) Q_r\left(s_t, a_t^h, a_t^{-r}; \varphi_{r,i}\right) \end{aligned} \tag{15}$$

where $\sum_{l=0}^{T_i - t}\left(\gamma^l \times r_{t+l}^r\right)$ is the estimated robot's return value according to (11). and a_t^{-r} represents an arbitrary robot task choice. $\sum_{a_t^{-r} \in \boldsymbol{A}_t^r} \pi_r\left(a_t^{-r} \mid s_t, a_t^h; \beta_{r,i}\right) Q_r\left(s_t, a_t^h, a_t^{-r}; \varphi_{r,i}\right)$ is the expected return value for executing the pre-update robot policy in s_t. $\beta_{r,i}$ is the network parameter of the pre-update robot's actor network, and $\varphi_{r,i}$ is the network parameter of the pre-update robot's critic network.

Ad_t^h is defined as the difference between the human's return value for executing the task combination $\left(a_t^h, a_t^r\right)$ and the expected return value for executing the pre-update human policy. The calculation formula for Ad_t^h is as follows:

$$\begin{aligned} Ad_t^h = & \sum_{l=0}^{T_i - t}\left(\gamma^l \times r_{t+l}^h\right) \\ & - \sum_{a_t^{-h} \in \boldsymbol{A}_t^h} \pi_h\left(a_t^{-h} \mid s_t; \beta_{h,i}\right) Q_h\left(s_t, a_t^{-h}, a_t^{-r'}; \varphi_{h,i}\right) \end{aligned} \tag{16}$$

where $\sum_{l=0}^{T_i - t}\left(\gamma^l \times r_{t+l}^h\right)$ is the estimated return value of the human in (10). a_t^{-h} represents an arbitrary human task choice. To facilitate the calculation of the expected return value for executing the pre-update human policy, the robot task selection in response to the human task selection a_t^{-h} is obtained by sampling through the robot's actor network, denoted as $a_t^{-r'}$. $\sum_{a_t^{-h} \in \boldsymbol{A}_t^h} \pi_h\left(a_t^{-h} \mid s_t; \beta_{h,i}\right) Q_h\left(s_t, a_t^{-h}, a_t^{-r'}; \varphi_{h,i}\right)$ is the expected return value for executing the pre-update human policy in s_t. $\beta_{h,i}$ is the network parameter of the pre-update human's actor network, and $\varphi_{h,i}$ is the network parameter of the pre-update human's critic network.

Based on Ad_t^h and Ad_t^r, the loss functions of the human's and robot's actor networks are constructed to improve the policies of both the human and the robot. They are represented as follows:

$$\begin{aligned} L_h^p = & -\sum_{t=0}^{T_i} \min \\ & \left[r_i\left(\beta_{h,i}'\right) \times Ad_t^h, \operatorname{clip}\left(r_i\left(\beta_{h,i}'\right), 1-\tau, 1+\tau\right) \times Ad_t^h\right] \end{aligned} \tag{17}$$

$$\begin{aligned} L_r^p = & -\sum_{t=0}^{T_i} \min \\ & \left[r_i\left(\beta_{r,i}'\right) \times Ad_t^r, \operatorname{clip}\left(r_i\left(\beta_{r,i}'\right), 1-\tau, 1+\tau\right) \times Ad_t^r\right] \end{aligned} \tag{18}$$

where L_h^p represents the loss function of the actor network of the human, and L_r^p represents the loss function of the actor network of the robot. For L_h^p, $r_i\left(\beta_{h,i}'\right) = \pi_h'\left(a_t^h \mid s_t; \beta_{h,i}'\right) / \pi_h\left(a_t^h \mid s_t; \beta_{h,i}\right)$, which is the probability ratio of the update policy π_h' and the pre-update policy π_h to ensure that multiple updates can be made [21]. The values of $r_i\left(\beta_{h,i}'\right)$ are then clipped to the range $[1-\tau, 1+\tau]$ to prevent the update to the policy loss from being too large. Similarly, for L_r^p, $r_i\left(\beta_{r,i}'\right) = \pi_r'\left(a_t^r \mid s_t, a_t^h; \beta_{r,i}'\right) / \pi_r\left(a_t^r \mid s_t, a_t^h; \beta_{r,i}\right)$.

The loss functions of the critic networks of the human and robot are used to decrease the difference between the estimated return value and the predicted return value in each s_t. They are represented as follows:

$$L_h^v = \sum_{t=0}^{T_i}\left(Q_h\left(s_t, a_t^h, a_t^r; \varphi_{h,i}\right) - \sum_{l=0}^{T_i - t}\left(\gamma^l \times r_{t+l}^h\right)\right)^2 \tag{19}$$

$$L_r^v = \sum_{t=0}^{T_i}\left(Q_r\left(s_t, a_t^h, a_t^r; \varphi_{r,i}\right) - \sum_{l=0}^{T_i - t}\left(\gamma^l \times r_{t+l}^r\right)\right)^2 \tag{20}$$

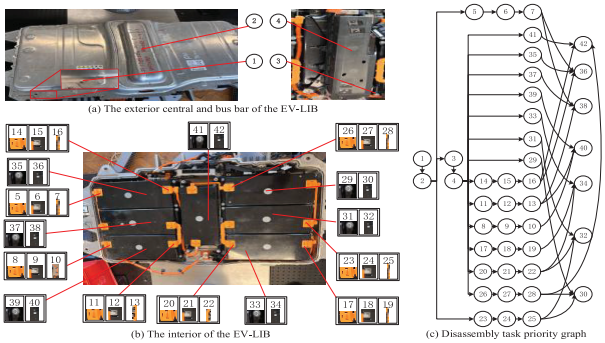


Fig. 4. The main parts of the EV-LIB.

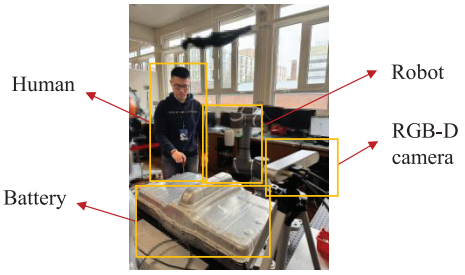


Fig. 5. HRC disassembly scene.

where L_h^v represents the loss function of the critic network of the human, and L_r^v represents the loss function of the critic network of the robot.

Algorithm 1 i-MAPPO Algorithm

Initialization: The actor networks $\pi_{h,1}\left(\cdot \mid s_t;\beta_{h,1}\right)$ and $\pi_{r,1}\left(\cdot \mid s_t, a_t^h;\beta_{r,1}\right)$, the critic networks $Q_h\left(s_t, a_t^h, a_t^r;\varphi_{h,1}\right)$ and $Q_r\left(s_t, a_t^h, a_t^r;\varphi_{r,1}\right)$, the storage container D.

Parameters Setting: Learning rates $\alpha 1$ and $\alpha 2$, discount factor γ, number of epochs K, total number of iterations N, ε, clipping parameter τ.

Training process:

for $i = 1$ to N **do**
Execute policies $\pi_{h,i}$ and $\pi_{r,i}$ to obtain the disassembly task assignment data $\left\{s_t, a_t^h, a_t^r, r_t^h, r_t^r, s_{t+1}\right\}_{t\le T_i}^{i}$, and store in D.
Calculate $\left\{Ad_t^h, Ad_t^r\right\}_{t\le T_i}^{i}$ according to (15) and (16) and store in D.
Construct update networks for the human and robot actor networks with parameters $\beta'_{h,i}$ and $\beta'_{r,i}$, respectively.
$\beta'_{h,i} \leftarrow \beta_{h,i}, \beta'_{r,i} \leftarrow \beta_{r,i}$
for $k = 1$ to K **do**
Apply gradient update on the human's and robot's actor networks based on (17) and (18) at learning rate $\alpha 1$.
Apply gradient update on the human's and robot's critic networks based on (19) and (20) at learning rate $\alpha 2$.
end for
Empty D
end for

After the loss function is determined, the training data are used multiple times to update the parameters of the networks. The number of times the entire data from an iteration are used to update the parameters of the networks is defined as the number of epochs. The i-MAPPO algorithm is below.

VI. Experiment Analysis

A. Case Study

In Fig. 4, an EoL electric vehicle lithium-ion battery (EV-LIB) is used to validate the approach presented in this study. Fig. 4(a) shows the exterior of the EV-LIB. It also shows the bus bars that are used to connect battery modules. Fig. 4(b) shows the interior of the EV-LIB. In this study, the EV-LIB is disassembled to the battery module level to support subsequent remanufacturing activities. For the EV-LIB, there are 42 parts in total to be disassembled, thus resulting in a total of 42 disassembly tasks. The disassembly task priority graph is shown in Fig. 4(c). The disassembly position in Table II refers to the position where the disassembly task is performed in the product coordinate system. The variance σ_i^2 of the human disassembly time in (9) is set to a constant value of 1.5 s.

B. Performance Analysis

In this case, the i-MAPPO algorithm is utilized to find the optimal disassembly task assignment policies to adapt to the uncertainty of the human. The computational environment for the experiments includes an AMD Ryzen5 5600H CPU and 16 GB of memory. Python tools are used to create a simulation environment. Since the number of tasks is 42, the size of s_t is (44×42), and the number of task choices is 43. In each network, the processing of s_t involves the following sequential operations to enhance learning performance: three convolutional networks with a kernel size of 3×3 and 32 output channels, followed by an average pooling layer; three convolutional networks with a kernel size of 3×3 and 64 output channels; another average pooling layer; and a fully connected layer consisting of 500 nodes. Additionally, when processing tasks a_t^h and a_t^r, they undergo one-hot encoding, initially converting to a size of 43, and then are further processed through a fully connected layer consisting of 500 nodes. The parameters of the algorithm are set as follows: the iteration number $N = 600$, the discount factor $\gamma = 0.9$, and $\varepsilon = 0.95$.

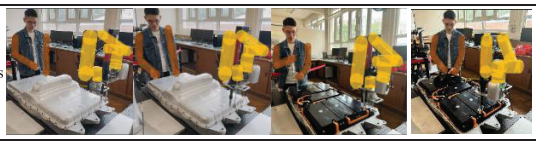
The learning rates of the actor network and the critic network are set as 0.0001 and 0.0002, respectively. The training numbers of epochs of both the actor network and the critic network are set as 10. The clipping parameter in (17) and (18) is set as $\tau = 0.2$. w_r in (6) is set as 1000, and w_h in (8) is set as 200. In the following parts of this section, the performance evaluation indices for the algorithm are first introduced, followed by a performance analysis of the proposed approach and a comparative analysis with other approaches.

Furthermore, determining the minimum distance between the human and the robot while performing different tasks is crucial. To achieve this, an HRC disassembly scenario is constructed, as shown in Fig. 5. The camera is positioned at an angle that provides an unobstructed view of the human to prevent occlusion by the robot. Table III presents the human-robot distances in several typical scenarios and the

TABLE II
DETAILED DESCRIPTION OF THE DISASSEMBLY TASKS

No.	Name	Quantity	Class	Disassembly position (mm, mm, mm)	Human mean disassembly time (s)	Robot disassembly time (s)
1	battery top screws	35	H+R	/	332	1470
2	battery top cover	1	R	(500,250,150)	/	11
3	central screws	4	H/R	/	64	108
4	central bus bar	1	H	(450,200,120)	3.6	/
5	bus bar cover 1	2	H	(100,300,100), (100,200,100)	6	/
6	bus bar screws 1	2	H/R	(100,300,100), (100,200,100)	13.5	40
7	bus bar #1	1	H/R	(100,250,100)	2.3	30
8	bus bar cover #2	2	H	(100,100,100), (400,60,50)	6	/
9	bus bar screws #2	2	H/R	(100,100,100), (400,60,50)	13.5	40
10	bus bar #2	1	H	(200,100,100)	2.3	/
11	bus bar cover #3	2	H	(400,100,100), (400,200,100)	6	/
12	bus bar screws #3	2	H/R	(400,100,100), (400,200,100)	13.5	40
13	bus bar #3	1	H/R	(350,60,100)	2.3	30
14	bus bar cover #4	2	H	(400,400,100), (500,100,50)	6	/
15	bus bar screws #4	2	H/R	(400,400,100), (500,100,50)	13.5	40
16	bus bar #4	1	H	(400,200,100)	2.3	/
17	bus bar cover #5	2	H	(550,100,50), (900,200,100)	6	/
18	bus bar screws #5	2	H/R	(550,100,50), (900,200,100)	13.5	40
19	bus bar #5	1	H	(700,100,60)	2.3	/
20	bus bar cover #6	2	H	(550,250,100), (900,350,100)	6	/
21	bus bar screws #6	2	H/R	(550,250,100), (900,350,100)	13.5	40
22	bus bar #6	1	H/R	(550,300,100)	2.3	30
23	bus bar cover #7	2	H	(900,350,100), (900,450,100)	6	/
24	bus bar screws #7	2	H/R	(900,350,100), (900,450,100)	13.5	40
25	bus bar #7	1	H/R	(900,400,100)	2.3	30
26	bus bar cover #8	2	H	(450,400,100), (500,400,100)	6	/
27	bus bar screws #8	2	H/R	(450,400,100),(550,400,100)	13.5	40
28	bus bar #8	1	H/R	(500,400,100)	2.3	30
29	Battery module screws #1	4	H/R	/	40.4	128
30	Battery module #1	1	R	(700,400,100)	/	54
31	Battery module screws #2	4	H/R	/	40.4	128
32	Battery module #2	1	R	(700,300,100)	/	54
33	Battery module screws #3	4	H/R	/	40.4	128
34	Battery module #3	1	R	(700,200,100)	/	54
35	Battery module screws #4	4	H/R	/	40.4	128
36	Battery module #4	1	R	(300,400,100)	/	54
37	Battery module screws #5	4	H/R	/	40.4	128
38	Battery module #5	1	R	(300,300,100)	/	54
39	Battery module screws #6	4	H/R	/	40.4	128
40	Battery module #6	1	R	(300,200,100)	/	54
41	Battery module screws #7	4	H/R	/	40.4	128
42	Battery module #7	1	R	(450,300,100)	/	54

TABLE III
THE MEASURED HUMAN-ROBOT MINIMUM DISTANCES AND CORRESPONDING TIME AT SEVERAL TYPICAL SCENARIOS

Scenarios				
Measuring time (ms)	56	57	53	59
Distance (m)	0.413	0.447	0.574	0.395

corresponding measurement times. As shown in Table III, in different scenarios, the human position is obtained using an RGB-D camera, and the minimum human-robot distance can be quickly calculated using the GJK algorithm, with an average time of approximately 55 ms. It meets the requirements for real-time human-robot distance computation.

1) Evaluation Indices: The optimal disassembly task assignment policies proposed in this study ensure the safety and efficiency of the entire HRC-based disassembly process. To demonstrate the effectiveness of the proposed approach, this paper designs evaluation indices for both efficiency and safety, as detailed below.

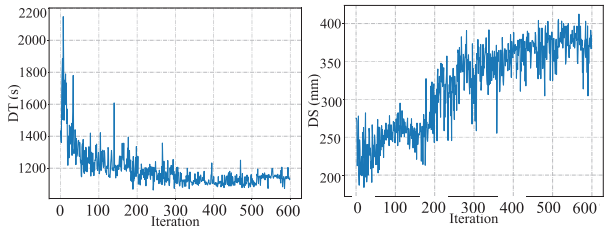
Disassembly time (DT): The disassembly time refers to the time from the start of disassembly to the end of disassembly. The smaller the *DT* is, the higher the efficiency is.

Disassembly safety (DS): Disassembly safety refers to the average distance value between the human and the robot, excluding cases where either the human or the robot chooses *wait*. The larger the *DS* is, the greater the safety is.

2) Performance Analysis of Optimal Task Assignment Policies:

a) Learning the Optimal Policies: Fig. 6 shows *the DTs* and *DSs* obtained in each iteration in the process of learning the optimal policies via the i-MAPPO algorithm. Fig. 6 shows that *DT* decreases and *DS* increases as the number of iterations increases. The algorithm converges when the number of iterations reaches 500. Intuitively, from Fig. 6, the i-MAPPO algorithm has strong convergence. After a number of iterations, the algorithm eventually converges (*DT* = 1133.5 s, *DS* = 377.08 mm).

b) Human Uncertainty Adaptability Performance: Following the iterations of i-MAPPO, the learned optimal policies can effectively adapt to human uncertainties during the dis-



(a) The change diagram of *DT* (b) The change diagram of *DS*

Fig. 6. Performance of the i-MAPPO algorithm.

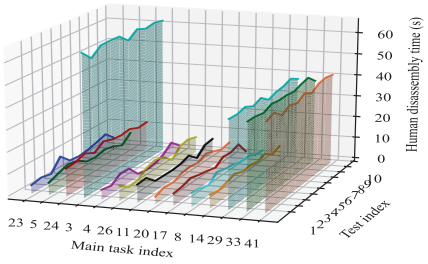


Fig. 7. Comparison of the time taken by the human in each test.

assembly process, including uncertain disassembly time and task selection. The following experimental analyses will be conducted for these two uncertainties.

To evaluate the adaptability of the proposed approach to uncertain human disassembly time, task assignment tests are conducted via the learned optimal policies. A total of ten tests are conducted, with the human disassembly time in each test following a normal distribution. Fig. 7 presents a 3D plot to compare the disassembly times required by the human to execute the tasks in each test. To demonstrate the performance relative to the results after iterative learning, the mean values (to compare the quality of the policies) and standard deviations (SD) (to compare the stability of the policies) are calculated.

As shown in Fig. 7, there are fluctuations in the time required for the human to complete the same task in each test. The mean *DT* is 1447.9 s, and the mean *DS* is 386.7 mm, which is close to the results obtained after the iterations of the i-MAPPO algorithm. The standard deviations of *DT* and *DS* are 6.3 s and 11.4 mm, respectively, which are relatively small compared with the overall values. It demonstrates that although the human disassembly times are variable, they do not significantly impact the overall HRC-based task assignment process. This finding indicates that the proposed approach not only is stable in adapting to the uncertain human disassembly time but also maintains the safety and efficiency of the disassembly process.

The uncertainty in human task selection is reflected primarily in the following: during a disassembly task assignment process, in a certain state s_t, the human could not execute tasks according to the optimally learned policies but randomly selects disassembly tasks (i.e., following a uniform distribution where each selection has an equal probability). In response, the proposed approach adaptively adjusts the robot's task selection based on the human's choices, thereby ensuring that the performance of the disassembly process is maintained while accommodating the uncertainty in human task selection.

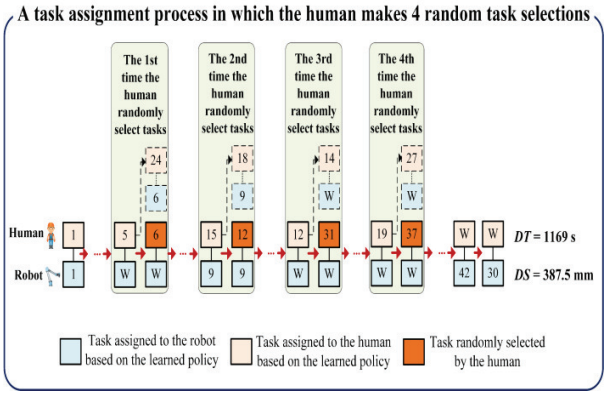


Fig. 8. The disassembly task assignment process.

TABLE IV
TEN SETS OF (DT, DS, GD, HV) VALUES FOR DIFFERENT ε

ε	0.92	0.94	0.96	0.98	1.00
Mean m	4	3	2	1	0
Mean *DT* (s)	1159.2	1168.9	1164.4	1149.4	1147.9
SD of *DT* (s)	22.9	23.3	21.7	11.7	6.3
Mean *DS* (mm)	373.4	382.4	380.7	380.0	386.6
SD of *DS* (mm)	14.6	14.7	17.6	12.4	11.4
GD	0.016	0.015	0.014	0.012	0.010
HV	0.543	0.543	0.551	0.553	0.551

Tests are conducted to validate the adaptability of the proposed approach to uncertain human task selection, where the human follows an ε-greedy strategy to select disassembly tasks. That is, the smaller the value of ε, the greater the probability that the human randomly selects disassembly tasks. It is assumed that in each test, the number of times that the human randomly selects disassembly tasks is denoted as m. Fig. 8 shows a task assignment process in which the human makes $m = 4$ random task selections. The smaller ε, the larger m is likely to be. The *DT* and *DS* values for task assignment under different ε values are evaluated. The values of ε are selected from {0.92, 0.94, 0.96, 0.98, 1.00}, with ten tests conducted for each value. When $\varepsilon = 1.0$, it indicates that the human executes the assigned tasks throughout the entire task assignment process, and its results are used as a baseline for comparison. The mean m values under different ε values and the corresponding results are shown in Table IV. As shown in Table IV, as the value of ε decreases, the mean m value increases, and the mean *DT* also increases to some extent, while the mean *DS* decreases slightly, but all changes are within a small range. This indicates that the policies proposed in this study exhibit adaptability to the uncertainty in human task selection.

Furthermore, since the problem addressed in this study is essentially a multi-objective optimization problem, to compare the performance of the solution sets for different m values more comprehensively, the hypervolume (HV) and generational distance (GD) indices are introduced. The HV is a metric that quantifies the volume enclosed by the solution set and a reference point, with more details available in [22]. A larger HV value indicates better diversity and quality of the obtained solution set. GD refers to the average shortest

distance between the solution set and the optimal Pareto solution set, with more details available in [23]. A smaller GD value indicates a higher quality of the obtained solution set. The Pareto solution set can be obtained from a number of simulation experiments. To reduce the mutual scale influence of *DTs* and *DSs*, before the GD value is calculated, the *DT* and *DS* values in the solution set need to be normalized, as shown in the following formula.

$$DT_{norm} = \frac{DT_{max} - DT}{DT_{max} - DT_{min}} \tag{21}$$

$$DS_{norm} = \frac{DS_{max} - DS}{DS_{max} - DS_{min}} \tag{22}$$

where DT_{norm} and DS_{norm} are the normalized DT value and DS value, respectively. DT_{max} and DT_{min} are the maximum value and minimum value of the DT, respectively. DS_{max} and DS_{min} are the maximum value and minimum value of the DS, respectively.

The reference point for HV calculation is established at (1.0, 0.0). Table IV shows the GD and HV values for different ε values. the GD value gradually decreases, whereas the HV value gradually increases with increasing ε. This observation suggests that an increase in the frequency of random task selection by the human leads to a decline in the overall performance of the task assignment process. However, the differences in the GD and HV values between $\varepsilon < 1.00$ and $\varepsilon = 1.00$ are minimal. This finding indicates that the proposed approach maintains low GD values and high HV values under varying frequencies of random task selection by the human, demonstrating its strong adaptive capability in the face of uncertain human task choices.

3) Comparative Analysis With Other Approaches: To validate the advantages of the proposed i-MAPPO algorithm, first, it is compared with the NSGA-II algorithm, which is a widely used heuristic approach for multi-objective optimization. The NSGA-II algorithm, which is a variant of the genetic algorithm, is implemented with a crossover rate of 0.9 and a mutation rate of 0.1, following the settings proposed by Xu et al. [8]. Second, the i-MAPPO algorithm is compared with other multi-agent reinforcement learning algorithms, including MAPPO, multi-agent deep Q-network (MADQN), and i-MADQN algorithms. i-MADQN also solves the Stackelberg model-based equilibrium policy, and MAPPO and MADQN are designed to solve the Nash-based equilibrium policy, providing a basis for comparison between the two equilibrium policies. Each algorithm is iterated 600 times, and the performance is evaluated based on the quality of the Pareto solution sets, adaptability to human uncertainty, and algorithm training time.

a) The Performance Comparison of the Pareto Solution Sets: Fig. 9(a) depicts the Pareto solution set derived from each of the five algorithms. To evaluate the overall performance of these algorithms, HV and GD values are used for assessment. Fig. 9(b) shows the comparison results.

As depicted in Fig. 9(a), the solutions within the Pareto solution set, derived through the i-MAPPO algorithm, exhibit smaller *DT* values and larger *DS* values. Furthermore, Fig. 9(b) indicates that i-MAPPO outperforms the NSGA-II, MAPPO,

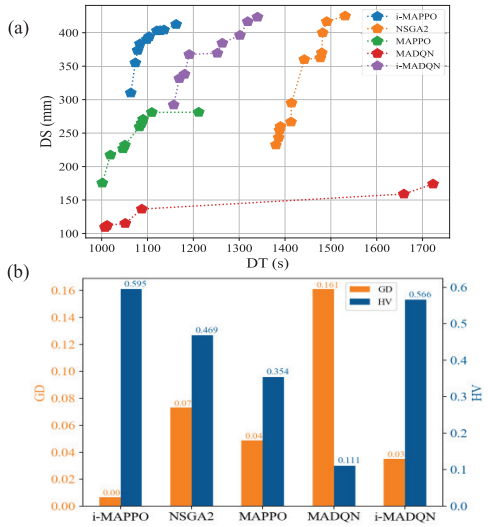


Fig. 9. The comparison with different algorithms.

TABLE V

SUCCESS RATE OF TASK ASSIGNMENT FOR EACH APPROACH AT DIFFERENT VALUES OF ε AND THE GD AND HV VALUES OF THE VALID SOLUTIONS OBTAINED BY EACH APPROACH

Algorithm	NoS ε=0.92	NoS ε=0.94	NoS ε=0.96	NoS ε=0.98	GD	HV
i-MAPPO	**10**	**10**	**10**	**10**	**0.009**	**0.538**
NSGAII	1	1	5	7	0.085	0.296
MAPPO	7	6	7	8	0.474	0.122
i-MADQN	**10**	**10**	**10**	**10**	0.047	0.485
MADQN	4	4	6	7	0.372	0.095

MADQN, and i-MADQN algorithms. In particular, the results of the i-MADQN algorithm are close to those of the i-MAPPO algorithm, demonstrating the excellent performance of the Stackelberg-based equilibrium policy.

b) Adaptability Comparison to Human Uncertainty: To highlight the advantages of the proposed algorithm in adapting to human uncertainty, in this section, a series of tests are conducted on i-MAPPO, NSGA-II, MAPPO, MADQN, and i-MADQN. The experimental procedures are as follows:

Before testing, the task assignment models for i-MAPPO, NSGA-II, MAPPO, MADQN, and i-MADQN are pretrained. During each task assignment test, the human selects tasks using the ε-greedy strategy, with human disassembly time modeled as a normal distribution. The ε values considered are {0.92, 0.94, 0.96, 0.98}, and for each ε value, each algorithm performs 10 task assignment trials. Table V records the number of successful task assignment (NoS), where task assignment failure refers to situations in which the tasks assigned to either the human or the robot cannot satisfy the task priority relationship constraints or the human-robot task classification constraints. Additionally, for all successful task assignment results (DTs and DSs), the GD and HV values obtained by each algorithm is shown Table V.

Table V highlights that regardless of how ε varies, the i-MAPPO algorithm consistently ensures the successful completion of the entire task assignment process. This phenomenon is also observed in the i-MADQN algorithm. This is because, when using the Stackelberg model-based policy, if the

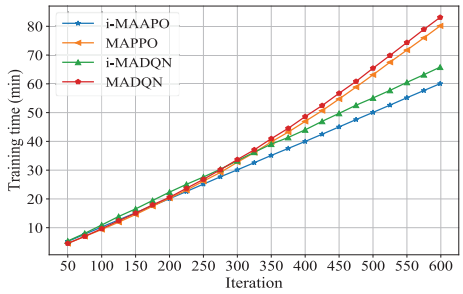


Fig. 10. The comparison of algorithm training time.

TABLE VI
THE AVERAGE INFERENCE TIME DURING ACTUAL EXECUTION

Algorithm	i-MAPPO	MAPPO	i-MADQN	MADQN
Inference time	48 ms	27 ms	79 ms	58 ms

human makes random task selections, the robot can quickly adjust its selected tasks based on the learned optimal policy.

In contrast, the NSGAII algorithm experiences a gradual decline in the success number of the task assignment process as the value of ε decreases. This decline is attributed to the fixed nature of the solutions obtained via the NSGAII algorithm. The deviation of the tasks performed by the human from the pre-planned tasks causes the subsequent task assignment sequence to fail to meet the task priority relationship constraints and the human-robot disassembly task classification constraints.

The success number of MAPPO stabilized at around 7, while the success number of MADQN is stabilized at around 5, both showing a certain level of adaptability. Failures in its task assignment process arise primarily from the robot's inability to adapt to human tasks. For example, when both the human and the robot select *wait*, the task assignment process could not proceed. Moreover, the data indicates that the i-MAPPO algorithm achieves the smallest GD value and the largest HV value. In summary, the i-MAPPO algorithm demonstrates excellent adaptability to human uncertainty.

c) Comparison of Training Time and Inference Time: To demonstrate the applicability of the i-MAPPO algorithm in real-world scenarios, both the training time and inference time of different algorithms are compared and analyzed. Inference time refers to the average decision-making time per step when applying the trained algorithm in practice. To highlight the advantages of the proposed algorithm, only similar reinforcement learning algorithms are considered for comparison. The experimental process involves recording the time consumed during the iteration process and the average inference time during execution. The results are shown in Fig. 10 and Table VI, respectively.

As shown in Fig. 10, the training time for each algorithm gradually increases with the number of training iterations. The i-MAPPO algorithm achieves the shortest training time of approximately 60 mins. This is primarily because the algorithm incorporates an unavailable task mask layer, significantly improving training efficiency. As shown in Table VI, i-MAPPO's average inference time is 48 ms, which is slightly higher than MAPPO's but lower than that for the MADQN series, demonstrating higher decision-making efficiency. This is because the i-MAPPO algorithm generates task instructions directly through the actor network.

VII. CONCLUSION

In this paper, the issue of human uncertainty in HRC during the disassembly of EoL products is addressed. By proposing optimal disassembly task assignment policies that adhere to the Stackelberg model, this paper offers a robust solution for dynamic disassembly planning that balances efficiency and safety in HRC. The human, acting as the leader, selects tasks that maximize efficiency, whereas the robot, as the follower, optimizes for safety in response to the human's choices. Furthermore, the implementation of the i-MAPPO algorithm enables the iterative determination of optimal policies, reinforcing the adaptability and success of the approach. A case study involving the disassembly of an electric vehicle battery demonstrates the practical application and efficacy of the proposed approach, achieving a 100% success rate in adapting to human uncertainty while maintaining short disassembly times and safe human–robot distances. This research thus makes significant strides in advancing the efficiency and safety of the HRC-based disassembly process.

In future work, the factors that influence human uncertainty in HRC and their impact on disassembly task assignments can be explored in greater depth. The case study presented in this paper focuses on complete disassembly planning problems. The approach is reconfigurable to selective disassembly planning problems by adding a mask layer in the i-MAPPO algorithm to set the probability of non-disassemblable tasks to 0 in the policy output. In addition, this study assumes that human working time follows a normal distribution with fixed variance. Still, individual differences exist in working time distributions, influenced by human factors such as physical strength and fatigue. Future research could incorporate individual human characteristics into the state observation space to develop personalized task assignment strategies while establishing a quantitative relationship model between variance and physiological load parameters. Furthermore, occlusion may still occur when obtaining the human convex hull through vision. To address this, research could utilize wearable IMU devices for more precise convex hull modeling of the human arms. Finally, it needs to explore how to integrate the pre-planned information into the reward mechanism to improve the training speed and performance of the i-MAPPO algorithm.

REFERENCES

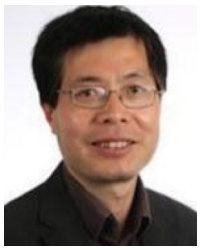
[1] W. Li, Y. Peng, Y. Zhu, D. T. Pham, A. Y. C. Nee, and S. K. Ong, "End-of-life electric vehicle battery disassembly enabled by intelligent and human–robot collaboration technologies: A review," *Robot. Comput.-Integr. Manuf.*, vol. 89, Oct. 2024, Art. no. 102758, doi: 10.1016/j.rcim.2024.102758.

[2] Y. Peng, W. Li, Y. Liang, and D. T. Pham, "Robotic disassembly of screws for end-of-life product remanufacturing enabled by deep reinforcement learning," *J. Cleaner Prod.*, vol. 439, Feb. 2024, Art. no. 140863, doi: 10.1016/j.jclepro.2024.140863.

[3] S. Lou, R. Tan, Y. Zhang, M. Zhou, and C. Lv, "Personalized disassembly sequence planning for a human–robot hybrid disassembly cell," *IEEE Trans. Ind. Informat.*, vol. 20, no. 9, pp. 11372–11383, Sep. 2024, doi: 10.1109/TII.2024.3403254.

[4] Z. Liu, Q. Liu, L. Wang, W. Xu, and Z. Zhou, "Task-level decision-making for dynamic and stochastic human–robot collaboration based on dual agents deep reinforcement learning," *Int. J. Adv. Manuf. Technol.*, vol. 115, nos. 11–12, pp. 3533–3552, Jun. 2021, doi: 10.1007/s00170-021-07265-2.
[5] W. Wang, R. Li, Z. M. Diekel, and Y. Jia, "Robot action planning by online optimization in human–robot collaborative tasks," *Int. J. Intell. Robot. Appl.*, vol. 2, no. 2, pp. 161–179, Jun. 2018, doi: 10.1007/s41315-018-0054-x.
[6] Y. Zhou, Y. Peng, W. Li, and D. T. Pham, "Stackelberg model-based human–robot collaboration in removing screws for product remanufacturing," *Robot. Comput.-Integr. Manuf.*, vol. 77, Oct. 2022, Art. no. 102370, doi: 10.1016/j.rcim.2022.102370.
[7] J. Xiao and K. Huang, "A comprehensive review on human–robot collaboration remanufacturing towards uncertain and dynamic disassembly," *Manuf. Rev.*, vol. 11, p. 20, Jul. 2024, doi: 10.1051/mfreview/2024015.
[8] W. Xu, Q. Tang, J. Liu, Z. Liu, Z. Zhou, and D. T. Pham, "Disassembly sequence planning using discrete bees algorithm for human–robot collaboration in remanufacturing," *Robot. Comput.-Integr. Manuf.*, vol. 62, Apr. 2020, Art. no. 101860, doi: 10.1016/j.rcim.2019.101860.
[9] M.-L. Lee, S. Behdad, X. Liang, and M. Zheng, "Task allocation and planning for product disassembly with human–robot collaboration," *Robot. Comput.-Integr. Manuf.*, vol. 76, Aug. 2022, Art. no. 102306, doi: 10.1016/j.rcim.2021.102306.
[10] X. Guo et al., "Human–robot collaborative disassembly line balancing problem with stochastic operation time and a solution via multi-objective shuffled frog leaping algorithm," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 3, pp. 4448–4459, Jul. 2024, doi: 10.1109/TASE.2023.3296733.
[11] M. Han, L. Yun, and L. Li, "Deep reinforcement learning-based approach for dynamic disassembly scheduling of end-of-life products with stimuli-activated self-disassembly," *J. Cleaner Prod.*, vol. 423, Oct. 2023, Art. no. 138758, doi: 10.1016/j.jclepro.2023.138758.
[12] W. Liu, X. Liang, and M. Zheng, "Task-constrained motion planning considering uncertainty-informed human motion prediction for human–robot collaborative disassembly," *IEEE/ASME Trans. Mechatronics*, vol. 28, no. 4, pp. 2056–2063, Aug. 2023, doi: 10.1109/TMECH.2023.3275316.
[13] J. Xiao, J. Gao, N. Anwer, and B. Eynard, "Multi-agent reinforcement learning method for disassembly sequential task optimization based on human–robot collaborative disassembly in electric vehicle battery recycling," *J. Manuf. Sci. Eng.*, vol. 145, no. 12, Dec. 2023, Art. no. 121001, doi: 10.1115/1.4062235.
[14] J. Gao, G. Wang, J. Xiao, P. Zheng, and E. Pei, "Partially observable deep reinforcement learning for multi-agent strategy optimization of human–robot collaborative disassembly: A case of retired electric vehicle battery," *Robot. Comput.-Integr. Manuf.*, vol. 89, Oct. 2024, Art. no. 102775, doi: 10.1016/j.rcim.2024.102775.
[15] K. Ramachandruni, C. Kent, and S. Chernova, "UHTP: A user-aware hierarchical task planning framework for communication-free, mutually-adaptive human–robot collaboration," *ACM Trans. Hum.-Robot Interact.*, vol. 13, no. 3, pp. 1–27, Sep. 2024, doi: 10.1145/3623387.
[16] Robots and Robotic Devices-Collaborative Robot, ISO/TS Standard 15066: 2016, 2016.
[17] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, "A fast procedure for computing the distance between complex objects in three-dimensional space," *IEEE J. Robot. Autom.*, vol. 4, no. 2, pp. 193–203, Apr. 1988, doi: 10.1109/56.2083.
[18] M. Mishra, P. Poddar, R. Agrawal, J. Chen, P. Tokekar, and P. B. Sujit, "Multi-agent deep reinforcement learning for persistent monitoring with sensing, communication, and localization constraints," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 2831–2843, Mar. 2025, doi: 10.1109/TASE.2024.3385412.
[19] C. Yu et al., "The surprising effectiveness of PPO in cooperative, multi-agent games," 2021, *arXiv:2103.01955*.
[20] P. Gautier, J. Laurent, and J.-P. Diguet, "Deep Q-learning-based dynamic management of a robotic cluster," *IEEE Trans. Autom. Sci. Eng.*, vol. 20, no. 4, pp. 2503–2515, Sep. 2023, doi: 10.1109/TASE.2022.3205651.
[21] D. Guo, L. Tang, X. Zhang, and Y. Liang, "Joint optimization of handover control and power allocation based on multi-agent deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13124–13138, Nov. 2020, doi: 10.1109/TVT.2020.3020400.
[22] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—A comparative case study," in *Proc. Int. Conf. Parallel Problem Solving Nature*, 1998, pp. 292–301.
[23] T. Okabe, Y. Jin, and B. Sendhoff, "A critical survey of performance indices for multi-objective optimisation," in *Proc. Congr. Evol. Comput.*, vol. 2, Dec. 2003, pp. 878–885.

Yiqun Peng received the B.E. and Ph.D. degrees in mechanical engineering from Wuhan University of Technology, China, in 2018 and 2024, respectively. He was with the Department of Mechanical Engineering, University of Birmingham, U.K., as a Visiting Ph.D. Student. He is currently with the Jianghuai Advance Technology Center, Hefei, as a Research Engineer. His research interests include human–robot collaboration for remanufacturing applications.



Weidong Li (Senior Member, IEEE) is currently with the University of Shanghai for Science and Technology, China, as the Chair Professor and the Dean of the School of Mechanical Engineering. He has been a Full Professor at the School of Mechanical and Automotive Engineering, Coventry University. U.K., since 2013. Before that, he was at Singapore Institute of Manufacturing Technology, University of Bath, Cranfield University, and Wuhan University of Technology. He has published 260 research papers in international journals and conferences and five books (Springer). His research interests include sustainable manufacturing and human–robot collaboration. His research has been sponsored by European Commission, Innovate U.K., EPSRC, U.K., and NSFC, China. He is a fellow of the Institution of Engineering and Technology (FIET) and the Institution of Mechanical Engineers (FIMechE).

Yong Zhou received the Ph.D. degree in mechanical manufacturing and automation from the Huazhong University of Science and Technology, China, in 2008. He is currently an Associate Professor at the School of Transportation and Logistics Engineering, Wuhan University of Technology. His research interests include robotics and 3D printing/scanning technologies.

Duc Truong Pham holds the Chance Chair of Engineering at the University of Birmingham. His academic output includes more than 600 technical articles and 17 books. He has supervised over 100 Ph.D. theses to completion. He received in excess of £40 M in external research grants and contracts. His research interests include intelligent systems, robotics and autonomous systems, and advanced manufacturing technology.

Dr. Pham is a fellow of the Royal Academy of Engineering, the Learned Society of Wales, the Society of Manufacturing Engineers, the Institution of Engineering and Technology, and the Institution of Mechanical Engineers.