Reinforcement Learning-Based Selective Disassembly Sequence Planning for the End-of-Life Products With Structure Uncertainty

Xikun Zhao, *Student Member, IEEE*, Congbo Li, *Senior Member, IEEE*, Ying Tang, *Senior Member, IEEE*, and Jiabin Cui

***Abstract*—Selective disassembly sequence planning (SDSP) is regarded as an efficient strategy to determine optimal disassembly sequences for extracting target parts (TP) from complex end-of-life (EOL) products. Previous research assumes that all EOL products have the same structure and the optimal selective disassembly sequences are given before the EOL products are removed. However, the products have different operation states during their use stage, which results in high structure uncertainty of EOL products. The structure uncertainty of EOL products often makes the predetermined selective disassembly sequences impractical for minimizing disassembly time and maximizing disassembly profit. This letter undertakes this challenge by integrated reinforcement learning (RL) to determine the optimal disassembly sequences adaptive to the structure uncertainty of the EOL products. Firstly, a multi-level selective disassembly hybrid graph model (MSDHGM) is developed to illustrate the contact, precedence, and level relationships among parts. Then, the SDSP is formulated as a finite Markov Decision Process and a deep Q-network based selective disassembly sequence planning (DQN-SDSP) is proposed. Finally, extensive comparative experiments are conducted to verify the proposed method compared with NSGA-II and ABC algorithms.**

***Index Terms*—Selective disassembly sequence planning, structure uncertainty, reinforcement learning, hybrid graph model.**

I. Introduction

With the fossil fuel depletion and environmental deterioration, remanufacturing and recycling of EOL products have gained a lot of research attention in recent years [1]. According to the Annual Energy Outlook 2019, the rising demand for industrial energy consumption will reach 39 quadrillions Btu in 2050 [2]. The industrial sectors are facing growing economic pressure as well as huge environmental challenges. Hence, both remanufacturing and recycling of EOL products are becoming critical elements of a circular economy.

Disassembly is the first step in remanufacturing and recycling because most EOL products must be removed into parts before they are treated [3]. Generally, disassembly includes complete disassembly and selective disassembly. The complete disassembly separates an entire EOL product into its constituent parts. Selective disassembly only extracts high-value or high-impact parts for the remanufacturing and recycling purpose [4]. In fact, the complete disassembly is often inefficient when only a few parts need to be removed [5]. On the contrary, selective disassembly only removes a few parts for obtaining the target part, which can save disassembly time and increase disassembly profit.

SDSP is regarded as an effective strategy to find the disassembly sequences for extracting target parts. Since products may experience different conditions during their use stage, the EOL products exhibit high uncertainties, such as disassembly time uncertainty and quality uncertainty [6], [7]. These uncertainties affect the decision-making of selective disassembly sequences. Some previous researches assumed that all EOL products have the same structure [8], [9]. In fact, the EOL products have different operation states during their recycling stage that may result in the missing of some parts. Besides, the reusable parts can be removed using destructive disassembly operations because of structural damage or rust. Thus, the EOL products will have different structure, called structure uncertainty, when they are treated. In turn, the structure uncertainty often makes the predetermined disassembly sequences impractical. Therefore, the influence of structure uncertainty on SDSP should be fully considered in SDSP.

Disassembly modeling methods, mainly include AND/OR graph, Petri Net, disassembly tree, undirected graph, and hybrid graph model (HGM) [10], are used to illustrate the assembly structure of EOL product. The AND/OR graph, Petri Net, disassembly tree and undirected graph assume that all parts are regarded as the same, and there is no consideration for the disassembly constraint relationship among components. Thus, they cannot clearly illustrate the differences in constraint relationship between parts [11]. The hybrid graph model can illustrate the direct and indirect constraint relationship of EOL products. Moreover, the hybrid graph model has better performance because of its simple structure and response to disassembly uncertainties. Unfortunately, the structure of the hybrid graph model becomes disordered for complex EOL products, which decreases the efficiency of SDSP. In this letter, the MSDHGM is proposed

Manuscript received February 25, 2021; accepted June 20, 2021. Date of publication July 20, 2021; date of current version August 20, 2021. This letter was recommended for publication by Associate Editor N. Kong and Editor J. Yi upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 51975075 and in part by the National Key R&D Program of China under Grant 2019YFB1706103. *(Corresponding author: Congbo Li.)*

Xikun Zhao, Congbo Li, and Jiabin Cui are with the State Key Laboratory of Mechanical Transmissions, Chongqing University, Chongqing 400044, China (e-mail: xikunzhao@163.com; congboli@cqu.edu.cn; cuijiabin@cqu.edu.cn).

Ying Tang is with the Department of Electrical and Computer Engineering, Rowan University, Glassboro, NJ 08028, USA, and with the Institute of Smart Education, Qingdao Academy of Intelligent Industries, Qingdao 266109, China.

Digital Object Identifier 10.1109/LRA.2021.3098248

by integrated hybrid graph model and hierarchical mechanism [12] to reduce the complexity of SDSP.

With the structure of EOL products continues to be complex, the search space of disassembly sequences will become wider. Thus, heuristic algorithms have been extensively searching for the optimal disassembly sequence. The disassembly sequences are obtained before the EOL products are removed. However, the structure uncertainty of EOL products often makes the predetermined disassembly sequences impractical. As for heuristic algorithms, a minor change to the structure of EOL products might restart the whole optimization process. Thus, the decision-making of heuristic algorithms cannot be adaptive to the changes in the structure of EOL products.

SDSP involves a great deal of disassembly knowledge (e.g., the disassembly tool and direction change, disassembly time and cost of each part, the length of disassembly sequence), which can be used to improve the decision-making process of SDSP. RL enables an agent to autonomously capture disassembly knowledge and learn good solutions by exploring the environment. Each selective disassembly sequence receives a feedback that guides the agent to learn the optimal strategy. So, we use RL to accumulate of disassembly knowledge, which can be used to respond to the structure uncertainty of EOL products. Some researchers have studied Q-learning-based disassembly methods to obtain disassembly sequences [13],[14]. However, the dynamic decision of SDSP results in large-scale state space for the Q-learning based disassembly methods. The Q-learning algorithm consumes enormous time to train the optimal strategy with large observation states. Hence, some researches used deep Q-network (DQN) to accelerate the learning speed [15]. Besides, the aforementioned research determined disassembly sequences without considering the structure uncertainty of EOL products.

Motivated by the aforementioned remarks, this letter attempts to fill this gap in this research and makes contributions in the following areas. 1) This letter develops an MSDHGM integrating the hybrid graph model and hierarchical mechanism to simplify constraint relationships among parts, which can be used to respond to the structure uncertainty of EOL products. 2) The decision problem of SDSP is formulated as a Markov decision process (MDP) and the DQN-SDSP is proposed for EOL products with structure uncertainty. The DQN-SDSP can be adaptive to obtain selective disassembly sequences for when the structure of EOL products changes.

The rest of the letter is organized as follows. Section II gives the multi-level selective disassembly hybrid graph model. Section III presents deep Q network-based selective disassembly sequence planning. The performance of the proposed method is demonstrated via a study in Section IV, followed by the conclusion in Section V.

II. Description of Selective Disassembly Model

A. Multi-Level Selective Disassembly Hybrid Graph Model

HGM is used to illustrate the structure of EOL products, which can be defined as a 4-tuple: $HGM = \{VF_n, VC_c, PR\}$, where $VF_n = \{VF_1, VF_2, \ldots, VF_n\}$ is the part that cannot be further disassembled. $VC_c = \{VC_{c1}, VC_{c2}, \ldots, VC_{cf}\}$is an undirected edge that represents the contact constraint between two parts. *PR* is precedence relationships among the parts of the EOL products, which consists of E_p and DE_u where $E_p = \{E_{p1}, E_{p2}, \ldots, E_{pe}\}$ illustrates the precedence relationships among the contact parts of the EOL products, which can be represented by a directed solid edge. $DE_u = \{DE_{u1}, DE_{u2}, \ldots, DE_{um}\}$ denotes the precedence relationships among two noncontact parts, which can be represented by directed dotted edge. The proposed MSDHGM is extending the HGM with hierarchical mechanisms, since HGM alone is unable to rapidly adjust its response to the (potential) dynamical state of EOL products. To exemplify our work, we have chosen a reducer to illustrate the MSDHGM. The assembly drawing of the reducer is shown in Fig. 1 and the appertaining MSDHGM is presented in Fig. 2.

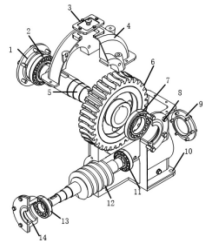

Fig. 1. Assembly drawing of a reducer.

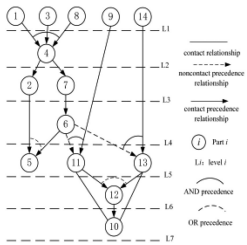

Fig. 2. MSDHGM of the reducer.

MSDHGM can be defined as follows:

$$\mathrm{MSDHGM} = \{\mathrm{VF}_n, \mathrm{VC}_c, \mathrm{PR}_h, \mathrm{L}_n\} \tag{1}$$

where $L_n = \{L_1, L_2, \cdots, L_n\}$ is the disassembly level of each part of EOL products. According to its composition characteristics, the $MSDHGM = \{VF_n, VC_c, PR, L_n\}$ can be categorized into $\boldsymbol{M_p} = \{VF_n, PR\}$, $\boldsymbol{M_c} = \{VF_n, VC_c\}$, and $\boldsymbol{M_l} = \{VF_n, L_n\}$. Then, the following defines precedence matrix $\boldsymbol{M_p}$, contact matrix $\boldsymbol{M_c}$, and level matrix $\boldsymbol{M_l}$, respectively.

1) Precedence Matrix $\mathbf{M_p}$*:* Generally, the precedence relationships include AND and OR relationships among parts [16]. The AND relationship indicates the part *i* can be disassembled after completing all its prior disassembly parts. For instance, in Fig. 2, part 4 can be obtained after part 1, part 3, and part 8 are released. The OR relationship means that the part *i* can be removed after one of its prior disassembly parts is disassembled. The precedence matrix $\boldsymbol{M_p}$ can be defined as follows.

$$M_p = \begin{bmatrix} a_{1,1}^{DE} & a_{1,2}^{DE} & \cdots & a_{1,n}^{DE} \\ a_{2,1}^{DE} & a_{2,2}^{DE} & \cdots & a_{2,n}^{DE} \\ \vdots & \vdots & & \vdots \\ a_{n,1}^{DE} & a_{n,2}^{DE} & \cdots & a_{n,n}^{DE} \end{bmatrix} \tag{2}$$

$$a_{i,j}^{DE} = \begin{cases} 1, \ if\ part\ i\ is \text{ the AND predecessor of part } j \\ -1, if\ part\ i\ is \text{the OR predecessor of part} j \\ 0, \ otherwise \end{cases} \tag{3}$$

where $a_{i,j}^{DE}$ indicates precedence relationship between the parts *i* and *j*.

2) Contact Matrix $\mathbf{M_c}$*:* The contact relationships of parts consist of two primary parts: direct contact constraints and indirect contact among parts. For instance, in Fig. 2, part 10 has a contact relationship with part 11, which is marked by the

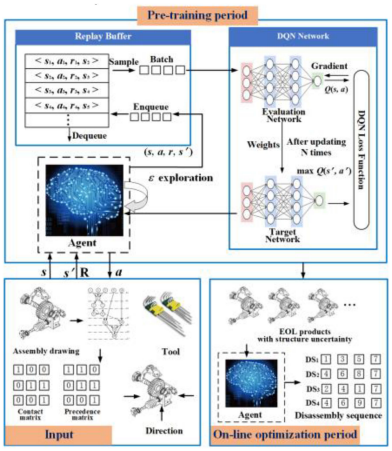

Fig. 3. DQN-SDSP diagram.

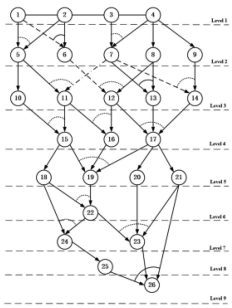

Fig. 4. The MSDHGM of the case study.

solid line. The contact matrix M_c is shown in Eq. 4 and Eq. 5 and the contact matrix $\boldsymbol{M_c}$ of a reducer is given in Fig. 4.

$$M_c = \begin{bmatrix} a^E_{1,1} & a^E_{1,2} & \cdots & a^E_{1,n} \\ a^E_{2,1} & a^E_{2,2} & \cdots & a^E_{2,n} \\ \vdots & \vdots & & \vdots \\ a^E_{n,1} & a^E_{n,2} & \cdots & a^E_{n,n} \end{bmatrix} \tag{4}$$

$$a^E_{i,j} = \begin{cases} 1, \ if\ there\ is\ a\ contact\ beteween\ part\ i\ and\ j \\ 0, \ otherwise \end{cases} \tag{5}$$

where $a^E_{i,j}$ indicates the contact relationship between part *i* and part *j*.

3) Level Matrix $\mathbf{M_l}$*:* The layered process is designed to obtain the disassembly level of each part, which is given in **Algorithm 1.** For instance, in Fig. 2, it is obvious that parts 13,89, and 14 are in level 1 at the beginning of the disassembly process. The level matrix $\boldsymbol{M_l}$. is shown in Eq. 6 and Eq. 7.

$$M_l = \begin{bmatrix} a^L_{1,1} & a^L_{1,2} & \cdots & a^L_{1,n} \\ a^L_{2,1} & a^L_{2,2} & \cdots & a^L_{2,n} \\ \vdots & \vdots & & \vdots \\ a^L_{n,1} & a^L_{n,2} & \cdots & a^L_{n,n} \end{bmatrix} \tag{6}$$

$$a^L_{i,j} = \begin{cases} 1, \ if\ part\ j\ is\ in\ level\ i \\ 0, \ otherwise \end{cases} \tag{7}$$

Algorithm 1: Layered Process for MSDHGM.

Input: Precedence matrix M_p and contact matrix M_c
Output: The level set L of parts

```
Def Generate Level ()
Initialize Level set L, l = 1, j = 1;
While (1)
Searching precedence matrix M_p and contact matrix M_c
Obtain all parts {p_1, p_2, ..., p_i} that can be removed simultaneously
If {p_1, p_2, ..., p_i} ≠ Ø
{p_1, p_2, ..., p_i} → L = {a_j1, a_j2, ..., a_ji, l}
Update M_p^{i+1}, M_c^{i+1} ← M_p^i, M_c^i
l←l+1
j←j+1
Else
Break
End If
End While
Return Level set L
```

Algorithm 2: Dynamic Adjustment of M_p and M_c

Input: Precedence matrix M_p, contact matrix M_c and *MP* (mp_1, mp_2, …, mp_v). The number of missing parts of EOL products *V*.
Output: M_p and M_c

```
Def Update ()
For i ≤V do:
mp = MP[0][i]
For j <n do:
a^DE_{mp,j}=0
a^E_{mp,j}=0
a^E_{j,mp}=0
End For
End For
Return M_p, M_c
```

B. Dynamic Adjustment of MSDHGM for Structure Uncertainty

The MSDHGM is dynamically updated based on the EOL product original structure. We assume that the missing parts of EOL products *MP* (mp_1, mp_2, …, mp_v) are obtained before they are disassembled. Thus, the $\boldsymbol{M_p}$ and $\boldsymbol{M_c}$ are updated according to *MP*, which is given in **Algorithm 2.**

C. Multi Objectives Optimization Model of SDSP

1) Objective Functions:

a) Disassembly Time: Disassembly time is the time consumption of operators in the disassembly process. Note that the disassembly time varies with the number of exchanges of disassembly tool and direction. So, the total disassembly time f_1 mainly contains the basic disassembly time, the penalty time for disassembly direction and tool changes, as shown in Eq. 8.

$$f_1 = \sum_{i=1}^{N} (t^i + x_i t^i_d + y_i t^i_{tool}) \tag{8}$$

where t^i, $t_d{}^i$, $t^i{}_{tool}$ are the consumed time of part *i*, the penalty time for changes in disassembly direction and tool, respectively. x_j, y_j are binary variables of either 0 or 1. $x_i = 1$ indicates that disassemble part i with changing disassembly direction.

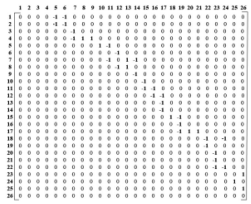

Fig. 5. The precedence matrix of case study.

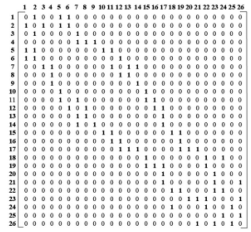

Fig. 6. The connection matrix of case study.

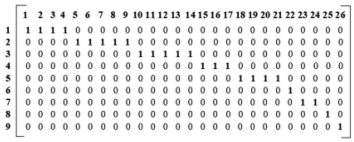

Fig. 7. The level matrix of case study.

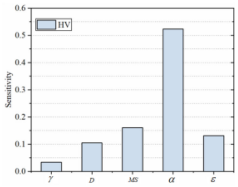

Fig. 8. Results of sensitivity analysis.

Similarly, $y_i = 1$ indicates that disassemble part *i* with changing disassembly tool.

b) Disassembly Profit: Another objective is disassembly profit f_2, which is formulated as follows:

$$f_2 = v_{TP} - \sum_{j=1}^{N}(c_j d_j) - h_m f_1 \tag{9}$$

where v_{TP}, c_j, h_m are the recycled value of target parts, cost of performing operation *j*, and basic working cost per unit time of operators, respectively. d_j is a binary variable of either 0 or 1 and $d_j = 1$ indicates that part *i* is handled.

2) Decision Variables and Optimization Constraints: The decision variable of SDSP is the sequence of disassembly parts *DS* (p_1, p_2, …, p_c) for handling the target part, which can be represented as a path from a start point to the target components. Note that the path must satisfy the constraints of the selective disassembly sequence, which as shown in Eq. 10-Eq. 14.

$$g_1(c) = c \geq 1 \tag{10}$$

$$g_2(a_{i,j}^{DE}) = \sum_{i=1}^{n} a_{i,j}^{DE} = 0 \tag{11}$$

$$g_3(a_{i,j}^{E}) = \sum_{i=1}^{n} a_{i,j}^{E} \geq 1 \tag{12}$$

$$g_4(x_j, y_j) = x_j, y_j \in \{0,1\}, j = 1, 2, \ldots, c \tag{13}$$

$$g_5(TP) = TP \in \{p_1, p_2, \ldots, p_n\} \tag{14}$$

Constraint 10 guarantees that at least one part is in the selective disassembly sequence. Constraint 11 guarantees that part *j* is not subject to the precedence relationship of other parts when part *j* is removed. Constraint 12 ensures at least one contact relationship among part *j*. Constraint 13 defines that the decision variables can only take 1 or 0. Constraint 14 ensures that the target part is in the disassembly sequence *DS* (p_1, p_2, …, p_n).

III. Deep Q Network For Selective Disassembly Sequence Optimization

To understand distinctly the logic of the proposed method, a framework of DQN-SDSP is shown in Fig. 3. Firstly, the SDSP is constructed as a finite Markov Decision Process. Then, in the pre-training process, the disassembly knowledge of SDSP is accumulated based on MSDHGM via an agent, which interacts over time with the states and actions in a discretized state space. In the on-line optimization process, the agent can use disassembly knowledge to obtain the optimal selective disassembly sequences for EOL products with structure uncertainty.

A. Reinforcement Learning

As for RL, a learning agent learns how to map situations to actions via interacting over time with its environment for maximizing the collected rewards. More specifically, at each time step *t*, the learning agent observes state s_t from a discrete state spaces *S*. Then an action $a_t \in A$ is selected based on its policyπ, which causes the environment transition of the environment to a new state s_{t+1} and generates an immediate reward r_t. Besides, it also outputs a numerical reward r_{t+1}, which is used to update a control policyπ^*.

To obtain selective disassembly sequences for the EOL products, the key elements of SDSP, such as state, action, reward, and environment, are introduced based on MDP. More specifically, at each time step *t*, the agent acquires disassembly part (action) that satisfies the precedence matrix $\boldsymbol{M_p}$, contact matrix $\boldsymbol{M_c}$ and level matrix $\boldsymbol{M_l}$, obtaining objective function of disassembly time and profit (reward). Then, the precedence matrix $\boldsymbol{M_p}$, contact matrix $\boldsymbol{M_c}$ and level matrix $\boldsymbol{M_l}$ are updated to a new state s_{t+1}. In short, the SDSP can be formulated as a Markov decision process, defined by:

a) $S = \{s_1, s_2, s_3 \ldots\}$ is a set of possible states
b) $A = \{a_1, a_2, a_3 \ldots\}$ is a set of possible actions namely disassembly parts
c) R (s_t, a_t, s_{t+1}) is a three-argument function for the next state given an action
d) $\gamma \in [0, 1]$, is the discount rate, which affects the performance of the learned policies

The policy π^* is designed to maximize the expected future discounted reward, shown in Eq. 15.

$$\pi^* = \arg\max_{\pi \in \mathrm{A}} R_k^{\pi} \tag{15}$$

where $R_k^{\pi} = \sum_{t=k}^{K} \gamma^{t-k} r^k$ and *K* are expected reward and total time step for SDSP. For the Q-Learning algorithm, the state action value Q (s_t, a_t) is used to denote the expected reward starts from state s_t and following action a_t, as given in Eq. 16.

$$\begin{aligned} Q(s_t, a_t) &= \mathbb{E}\left[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots | s_t, a_t\right] \\ &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_{t+k} | s_t, a_t\right] \end{aligned} \tag{16}$$

The optimal state action value function $Q*(s_t, a_t)$ is represented by the optimal Bellman equation as:

$$Q^*(s_t, a_t) = \mathbb{E}\left[R(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) | s_t, a_t\right] \tag{17}$$

The agent removes disassembly part from a given disassembly state s_t to find a new disassembly state s_{t+1}. Then, the environment gives rise to immediate reward r_t. The iterative value function can be calculated according to Eq. 18.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left[r(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)\right] \tag{18}$$

where α is the learning rate.

B. Deep Q-Network

To increase the convergence speed, the deep Q-Network is proposed to train a neural network to approximate the optimal state-action value function $Q*(s_t, a_t)$. The evaluation Q network and target Q network are defined to represent two Q values, given in Eq. 19, in which the Target Q-network is updated using parameters θ' of the evaluation Q network.

$$\begin{aligned} Q(s_t, a_t) &= Q(s_t, a_t; \theta) \\ Q^*(s'_t, a'_t) &= Q(s'_t, a'_t; \theta') \end{aligned} \tag{19}$$

where s'_t ,a'denote the state and action of target Q network. θ, θ' are the parameter of the evaluation and target Q network. In SDSP, the agent observes s_t ($\boldsymbol{M_p}$, $\boldsymbol{M_c}$, $\boldsymbol{M_l}$, $a_{t\text{-}1}$) and inputs feasible disassembly parts to the Q network. Then, the Q-values of each feasible disassembly part *i* is obtained to guide the agent to select the high value of the action. Besides, the DQN-SDSP is trained via minimizing the loss function $L(\theta)$, given below:

$$L(\theta) = \mathbb{E}\left[r_t + \gamma \max_{a'} Q^*(s_{t+1}, a', \theta') - Q(s_t, a_t, \theta)\right]^2 \tag{20}$$

In the pre-training process, we use the experience pool to store the disassembly knowledge consists of a (s_i, a_i, r_i, s_{i+1}) tuple at time step *i*. To avoid correlations of disassembly knowledge, the knowledge replay is proposed to randomly extract a batch of knowledge from the experience pool of selective disassembly.

1) Action and State-Space of SDSP: In SDSP, the action space is $a_t = \{a_1, a_2, a_3, \ldots, a_n\}$ representing all available parallel disassembly parts at state s_t. With the iterative generation increasing, the target part is selected from the EOL products. Note that the precedence and contact components among other parts should be released after part *i* is removed. The detailed procedure is given in **Algorithm 2**. The precedence matrix $\boldsymbol{M_p}$, contact matrix $\boldsymbol{M_c}$ and level matrix $\boldsymbol{M_l}$ are determined to constitute a four-dimensional selective disassembly state space based on $a_{t\text{-}1}$. The state space is defined as:

$$s_t = \{M_p, M_c, M_l, a_{t-1}\} \tag{21}$$

2) Reward Definition: In the pre-training process, the agent is given different reward when selecting different disassembly parts. The reward, as an essential part of disassembly knowledge, is designed to guide an agent to select the disassembly part at each time step. As for multi-objective optimization of SDSP, the agent is used to determine a strategy that can optimize the disassembly time and profit simultaneously [17]. So, we use Pareto dominance relationship evaluate disassembly actions from the reward perspective [18], which is defined as follows.

Firstly, the Pareto dominance relationship is used to evaluate the disassembly time and profit of each removed part. When determining the next disassembly part at state s_t, the agent will select one disassembly part from the next disassembly part set that obtained using **Algorithm 3**. To calculate the reward of disassembly sequence, we can evaluate all feasible selective disassembly sequences and obtain the corresponding rank. The procedure of calculating the rank of each selective disassembly sequence is shown in **Algorithm 4**.

Algorithm 3: Action and State for Disassembly Sequence Planning.

Input: Precedence matrix $\boldsymbol{M_p}$, Contact matrix $\boldsymbol{M_c}$, $s_t = \{\boldsymbol{M_p}, \boldsymbol{M_c}, \boldsymbol{M_l}, a_{t\text{-}1}\}$
Output: The available parallel disassembly parts $a_t = \{a_1, a_2, a_3, \ldots, a_n\}$
$s_{t+1} = \{\boldsymbol{M_p}, \boldsymbol{M_c}, \boldsymbol{M_l}, a_t\}$

Def Generate Action ()
For $i < N$ **do:**
  Search precedence matrix M_p and connection matrix M_c
  Calculate the number of precedence and contact constraint
  If $\sum_{i=1}^{N} a_{i,j}^{DE} = 0$ and $\sum_{i=1}^{N} a_{i,j}^{E} \geq 1$ **do:**
  Store a_i in $a_t = \{a_1, a_2, a_3, \ldots, a_n\}$
  End If
End For
Return $a_t = \{a_1, a_2, a_3, \ldots, a_n\}$
Def Update state ()
If Select a_i from $a_t = \{a_1, a_2, a_3, \ldots, a_n\}$ **do:**
  Update$M_p^i, M_c^i \rightarrow M_p^{i+1}, M_c^{i+1}$ where the constraints are released
End If
Obtain $s_{t+1} = \{\boldsymbol{M_p}, \boldsymbol{M_c}, \boldsymbol{M_l}, a_t\}$
Return $s_{t+1} = \{\boldsymbol{M_p}, \boldsymbol{M_c}, \boldsymbol{M_l}, a_t\}$

Secondly, the number of changes in disassembly direction and tool affects the disassembly time and profit. So, we can construct the penalty rewards, in which the agent can be given a penalty when the disassembly direction and tool of the current part are different from that of the next disassembly part.

Finally, we select the next disassembly part from the next disassembly part set and each next disassembly part may have different disassembly levels in SDSP. If the agent selects one disassembly part with a deeper-level, the length of selective disassembly will be reduced. Hence, we design the penalty rewards of disassembly level based on the MSDHGM to reduce the length of selective disassembly. In short, the reward definition of SDSP is shown as follows.

$$r(s_t, a_t) = \upsilon Rank_t + \varsigma_t PR_d + \nu_t PR_t + r_l(L_n - L_c) \tag{22}$$

where $\upsilon$is the reward factor of the rank. $Rank_t$ is the reward of action rank *t*. ς_t, ν_t denote binary variables of either 0 or 1. ς_t=1indicates that disassemble part *i* with changing disassembly direction. ν_t= 1indicates that disassemble part *i* with changing disassembly tool. PR_d, PR_t are penalty reward of disassembly direction and tool. L_n denotes the level of the next disassembly part, L_c represents the level of the current disassembly part, and r_l is the reward factor of the level.

3) Environment Description: At each time step *t*, the environment is used to transmit immediately a new state s_{t+1} to the learning agent. The environment makes the following contributions for SDSP. The first is to update the precedence matrix $\boldsymbol{M_p}$ and contact matrix $\boldsymbol{M_c}$, and present the new state

Algorithm 4: Calculate the Rank of Each Selective Disassembly Sequence.

Input: Current disassembly sequence $\{p_1, p_2, p_3, \ldots, p_i\}$, Next disassembly part set $\{a_1, a_2, a_3, \ldots, a_o\}$
Output: Rank set $<Rank_1, Rank_2, Rank_3, \ldots, Rank_o>$

Def Generate Rank ()
Initialize Rank set, $H = 1$
Obtain the next disassembly sequence $\{p_1, p_2, p_3, \ldots, p_i, a_1\}$, $\{p_1, p_2, p_3, \ldots, p_i, a_2\}$, $\ldots$, $\{p_1, p_2, p_3, \ldots, p_i, a_o\}$
Calculate the disassembly time and profit of each disassembly sequence, which stored in the objective function set $\Omega = \{(f_1^1, f_2^1), \ldots, (f_1^o, f_2^o)\}$
While (objective function set$\Omega \neq \emptyset$)
Obtain the Pareto Solutions from objective function set
Set $Rank_m, \ldots, Rank_l$ as H that stored in the corresponding Rank set.
Remove objective function values from the objective function setΩ.
$H \leftarrow H+1$
End While
Return Rank set $<Rank_1, Rank_2, Rank_3, \ldots, Rank_o>$

TABLE I
THE VALUE OF EACH PART

Part index	Disassembly time (s)	Tool (T)	Direction (D)	C_i(¥)
1	1.2	T2	+X	1.2
2	1.3	T2	+Y	2.3
3	2.6	T2	+X	0.35
4	0.8	T4	+Z	2.6
5	3.1	T2	+X	2.1
6	1.1	T3	+X	0.7
7	2.3	T2	+Y	0.6
8	6.5	T1	+X	1.6
9	2.7	T1	+Y	3.1
10	3.5	T1	+X	3.2
11	2.36	T2	-Y	0.3
12	2.55	T4	+X	2.2
13	6.6	T2	-Y	3.6
14	4.3	T4	-Y	1.36
15	2.42	T2	+X	0.8
16	5.6	T3	+X	2.3
17	0.2	T3	+X	0.9
18	4.2	T2	+X	3.34
19	2.65	T3	-Y	1.75
20	4.5	T3	+Z	4.564
21	7.6	T3	+X	7.932
22	2.65	T2	-Y	2.65
23	3.4	T2	+X	1.4
24	2.41	T3	+Y	2.1
25	6.7	T3	+X	1.7
26	3.5	T4	-Y	3.5

s_{t+1} to the learning agent. The next one is to determine whether the target part is selected. The third is to generate an immediate reward r_t, which guide a learning agent to select the optimal disassembly part.

IV. CASE STUDY

A. Experimental Setup and Pre-Training Process

To demonstrate the performance to respond to the EOL products with varying structure uncertainty, we compare the DQN-SDSP with the improved NSGA-II [19] and ABC [1] algorithms. The MSDHGM of the EOL product, which consists of 26 parts, is shown in Fig. 4. The contact matrix $\boldsymbol{M_c}$, precedence matrix $\boldsymbol{M_p}$, and level matrix $\boldsymbol{M_l}$ of the EOL products are presented in Fig. 5, Fig. 6 and Fig. 7. The disassembly time, disassembly tool, disassembly direction, and cost of each part are given in Table I . Besides, the sensitivity analysis of DQN-SDSP is carried out, which is shown in Fig. 8. The relevant parameters obtained in SDSP are shown in TABLE II. We use the artificial neural network (ANN) to approximate the optimal state-action value function. The input layer, hidden layer and output layer of DQN are set as 26-15-1.

TABLE II
PARAMETERS USED IN DQN-SDSP

Notation	Description	Value
t_d^j	Penalty time for changes in the disassembly direction	0.7 (s)
t_{tool}^j	Penalty time for changes in the disassembly tool	1 (s)
h_m	Cost of performing operation j,	0.4 (¥/s)
TP	Target part	23
v_{TP}	Recycled value of target part	200 (¥)
γ	Discount rate for DQN	0.95
D	Disassembly experience pool size	5000
MS	Minibatch size	20
α	Learning rate	0.001
ε	ε-greedy	from 1 to 0.01 and fixed at 0.01 thereafter

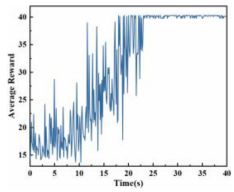

Fig. 9. Average reward of DQN-SDSP.

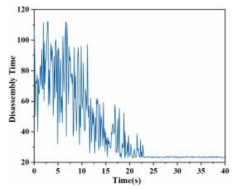

Fig. 10. Convergence curves of disassembly time.

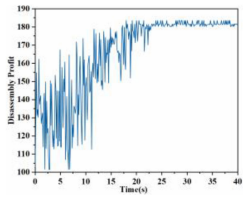

Fig. 11. Convergence curves of disassembly profit.

In order to demonstrate the performance of the proposed DQN-SDSP, two cases are designed as follow:

Case 1: DQN-SDSP is compared with NSGA-II and ABC algorithms in terms of the quality of solutions.
Case 2: The performance of adapting to structural uncertainty and stability for DQN-SDSP, NSGA-II and ABC are evaluated.

At the beginning of pre-training, the agent of DQN-SDSP cannot correctly make decision. So, the exploration strategy is adopted to make the agent exploit the selective disassembly sequences with a greater average reward. In Fig. 9, the trend of average reward rises with the accumulation of disassembly knowledge. The parameter ε in the ε-greedy algorithm is the probability of taking a random disassembly part and decreases with the increase of training iteration from 1 to 0.01 and fixed at 0.01 thereafter, improving the chances of selecting the optimal selective disassembly sequences.

Fig. 10 and Fig. 11 present the convergence curves of disassembly time and profit in the pre-training process. It can be

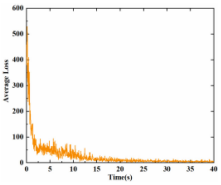

Fig. 12. Track of loss with network structures.

TABLE III
PARETO SOLUTIONS FOR EOL PRODUCTS WITH TARGET PARTS IS 23

No.	Sequence	Tool change	Direction change	f_1(s)	f_2(¥)
1	3→7→11→15→19→22→23	2	5	23.88	183.55
2	4→7→11→15→18→22→23	1	5	22.63	179.26
3	3→7→11→15→18→22→23	0	5	23.43	181.19
4	3→7→14→17→20→23	3	2	23.80	181.31
5	4→7→14→17→20→23	4	5	23.00	179.38
6	1→2→6→12→17→20→23	5	4	22.05	177.92
7	2→1→6→12→17→19→22→23.	5	3	22.15	178.99

TABLE IV
COMPARISON RESULTS FOR OPTIMIZATION USING DQN-SDSP, NSGA-II AND ABC

Method	Objective	Pareto solution							*HV*	*SP*
DQN-SDSP	Time	23.88	22.63	23.43	23.80	23.00	22.05	22.15	**1.55**	0.89
	Profit	183.55	179.26	181.19	181.31	179.38	177.92	178.99		
NSGA-II	Time	23.38	24.6	23.8	23. 00	22.63	22.15		1.52	0.52
	Profit	180.30	182.1	181.31	179.38	179.26	178.99			
ABC	Time	24.6	23.80	23.43	23.38	22.93	23. 00		1.38	**0.49**
	Profit	182.10	181.31	181.19	180.30	177.94	179.38			

seen that the disassembly time and profit decrease and increase respectively with the increase of iterative numbers. Note that the fluctuation of disassembly profit is more severe than that of disassembly time after 25s. The reason for this phenomenon is that the disassembly time, recycled value of target part, and cost of performing operation influence on disassembly profit simultaneously. So, the fluctuation of these factors will be superimposed on disassembly profit, which leads to it with higher fluctuation. Fig. 12 shows the track of the average loss of DQN-SDSP in pre-training process. It is found that there is no significant improvement after 25s, which indicates the convergence performance of DQN-SDSP.

B. Optimization Results

1) Pareto Solution of DQN-SDSP, NSGA-II and ABC: To compare the quality of solutions for multi-objective optimization, the Pareto dominance relationship is used to verify the performance of DQN-SDSP. In the pre-training process, the selective disassembly sequences for EOL products with target parts is 23 are shown in Table III. It is obvious that the number of tool changes is less than the number of tool direction changes. The reason is that the penalty time of disassembly direction and tool change is different, which causes the parts with the same disassembly tool are preferentially selected in SDSP. Table IV reports the Pareto solutions of DQN-SDSP, NSGA-II and ABC algorithms when the target part is 23, and the Pareto fronts of SDSP are depicted in Fig. 13. It can be seen that three methods obtain some of the same Pareto solutions and the proposed method can obtain a larger solution space, which offers a more selective disassembly sequences for operators.

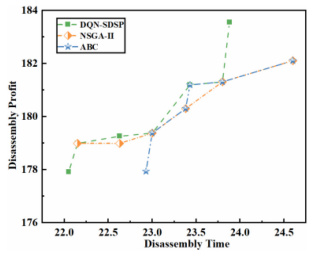

Fig. 13. Pareto solution of DQN-SDSP, NSGA-II and ABC.

TABLE V
RESULTS OF PERFORMANCE TEST BETWEEN DQN-SDSP, NSGA-II AND ABC

NO.	Method	Removed Parts	Selective Disassembly Sequences	(f_1, f_2)	CPU (s)
Scenario 1	DQN-SDSP	Part 1 Part 2 Part 4 Part 9	6→12→17→20→23	(16.15,183.78)	2.31
			14→17→20→23	(16.50,185.18)	
			14→17→19→22→23	(17.30,185.97)	
	NSGA-II		6→12→17→21→23	(17.85,179.73)	7.82
			14→3→17→20→23	(20.10,183.69)	
			14→17→21→23	(18.20,181.13)	
			7→11→15→19→22→23	(20.58,185.22)	
	ABC		14→6→17→19→22→23	(18.40,184.83)	7.56
			14→6→17→19→20→23	(17.60,184.04)	
			6→12→17→20→23	(16.15,183.78)	
			7→11→15→19→22→23	(20.58,185.22)	
Scenario 2	DQN-SDSP	Part 1 Part 2 Part 3 Part 4 Part 5 Part 7 Part 9	11→15→19→22→23	(17.58,187.02)	2.14
			14→17→20→23	(16.50,185.18)	
			6→12→17→20→23	(16.15,183.78)	
			14→17→19→22→23	(17.30,185.97)	
	NSGA-II		11→6→15→19→22→23	(20.68,185.08)	6.54
			10→15→19→22→23	(19.02,183.54)	
			10→15→18→22→23	(18.57,181.18)	
	ABC		6→12→17→20→23	(16.15,183.78)	6.15
			14→6→17→20→23	(17.6, 184.04)	
			11→6→15→19→22→23	(20.68,185.08)	

Hypervolume (HV) and Spacing (SP) are used to evaluate the closeness to the true Pareto front and the uniformity of the obtained solutions. More details of the aforementioned metrics can be found in [20] and [21]. To obtain HV, the reference point is set as $V = (30, 0.2)$. Note that the (f_1, f_2) should be changed into $(f_1, 1/f_2)$ when calculating the HV of solutions. It can be found from Table IV that the performance of the DQN-SDSP solutions is better than NSGA-II and ABC from the HV perspective, which reveals the Pareto front of DQN-SDSP better approach the true Pareto front. In terms of SP, when the obtained solutions are distributed more evenly, the SP is smaller. Fig. 13 shows that the solution of ABC is more evenly than those of DQN-SDSP and NSGA-II. Hence, the SP of ABC in Table IV is smaller than its competitor DQN-SDSP and ABC. As aforementioned, these metrics indicate that the DQN-SDSP has a better performance in terms of Pareto solutions.

2) The Performance of Adapting to Structural Uncertainty: We can remove some parts from the EOL product to simulate varying structural uncertainty, which is used to test the performance of adapting to structural uncertainty. We set two scenarios with different removed parts, and the selective disassembly sequences are shown in TABLE V. The varying structural uncertainty significantly affects the Pareto solutions in Table III. For instance, the obtained selective disassembly sequence {1→2→6→12→17→20→23} in Table III can be transformed into {6→12→17→20→23}in Scenario 1. Optimization results show that the selective disassembly sequences of DQN-SDSP in Table V are better than the most selective disassembly sequences

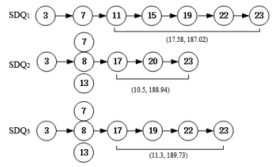

Fig. 14. Performance test of DQN-SDSP.

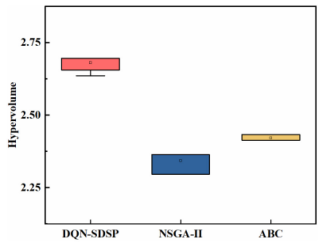

Fig. 15. Box plot of three methods on hypervolume value.

of Table III. It can be seen that the change in the product state significantly impacts on Pareto solution of SDSP. Thus, in SDSP, the selective disassembly sequences should be adaptive to the structure uncertainty of EOL products.

In SDSP, some parts have to be removed simultaneously because of structural damage or rust. To further test the performance of adapting to structural uncertainty, we select the selective disassembly sequence $SDQ_1 = \{3\rightarrow7\rightarrow11\rightarrow15\rightarrow19\rightarrow22\rightarrow23\}$ from Table III and assume that part 7, part 8, and part 13 have to be disassembled simultaneously. We can use DQN-SDSP to obtain selection disassembly sequences $SDQ_2 = \{3\rightarrow(7, 8, 13)\rightarrow17\rightarrow20\rightarrow23\}$ and $SDQ_3 = \{3\rightarrow(7, 8, 13)\rightarrow17\rightarrow19\rightarrow22\rightarrow23\}$where (7, 8, 13) denotes the part 7, part 8, and part 13 are disassembled simultaneously. It was not possible to obtain the disassembly time and profit of removing part 7, part 8, and part 13 simultaneously. Hence, we calculate the disassembly time and profit of partial disassembly sequences. Fig. 14 shows that the obtain SDQ_2 and SDQ_3 are better than SDQ_1 in terms of the disassembly time and profit of partial disassembly sequence. When some parts are removed simultaneously because of structural damage or rust, the proposed method can select the disassembly sequence based on the product state of the EOL product.

To compare the stability of the aforementioned algorithms, the hypervolume values of each algorithm running 20 times are recorded and the Box plot of the three methods is depicted in Fig. 15. It can be seen that the performance of three methods from best to worst is shown as follow: DQN-SDSP>NSGA-II>ABC.

V. Conclusion

In this work, we develop a DQN-SDSP to determine the selective disassembly sequences from efficient and economical perspectives. More specifically, firstly, the MSDHGM is explicitly developed to illustrate the contact, precedence, and level relationships between parts. Then, the SDSP is formulated as a finite Markov Decision Process and the DQN-SDSP is proposed. Finally, extensive comparative experiments are implemented to demonstrate the performance of the proposed method. The results show that the set of nondominated solutions and Pareto front from the proposed method are better than NSGA-II and ABC algorithms. Although the performance of the proposed DQN-SDSP has been verified, certain limitations exist. First, to obtain the decision support for SDSP in practice, we do not use actual disassembly data to validate this method. Second, the proposed method is designed for a specific EOL product and target part. Therefore, more advanced methods are developed in the future.

References

[1] Y. Ren, G. Tian, F. Zhao, D. Yu, and C. Zhang. "Selective cooperative disassembly planning based on multi-objective discrete artificial bee colony algorithm," *Eng. Appl. Artif. Intell.*, vol. 64, pp. 415–431, 2017.

[2] International Energy Agency (IEA). International energy outlook 2019. [Online]. Available: https://www.iea.org/reports/world-energy-outlook-2019

[3] L. Zhang, X. Zhao, Q. Ke, W. Dong, and Y. Zhong, "Disassembly line balancing optimization method for high efficiency and low carbon emission," *Int. J. Precis Eng Manuf-Green Technol.*, vol. 8, no. 1, pp. 233–247, 2021.

[4] A. Desai, and A. Mital, "Evaluation of disassembly ability to enable design for disassembly in mass production," *Int. J. Ind. Ergon.*, vol. 32, no. 4, pp. 265–281, 2003.

[5] S. S. Smith and W.-H. Chen, "Rule-based recursive selective disassembly sequence planning for green design," *Adv. Eng. Inform.*, vol. 25, no. 1, pp. 77–87, 2011.

[6] K. Wang, X. Li, L. Gao, and A. G. Akhil, "Partial disassembly line balancing for energy consumption and profit under uncertainty," *Robot. Comput. Integr. Manuf.*, vol. 59, pp. 235–251, 2018.

[7] S. Vongbunyong, S. Kara, and M. Pagnucco, "Basic behaviour control of the vision-based cognitive robotic disassembly automation," *Assem. Autom.*, vol. 33, no. 1, pp. 38–56, 2013.

[8] G. Tian, M. Zhou, and P. Li, "Disassembly sequence planning considering fuzzy component quality and varying operational cost," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 2, pp. 748–760, Apr. 2018.

[9] Y. Gao, Y. Feng, Q. Wang, H. Zheng, and J. Tan, "A multi-objective decision making approach for dealing with uncertainty in EOL product recovery," *J. Clean. Prod.*, vol. 204, pp. 712–725, 2018.

[10] Y. Tian, X. Zhang, Z. Liu, X. Jiang, and J. Xue, "Product cooperative disassembly sequence and task planning based on genetic algorithm," *Int. J. Adv. Manuf. Tech.*, vol. 105, no. 5-6, pp. 2103–2120, 2019.

[11] Y. Wang, F. Li, J. Li, J. Chen, J. Feng, and W. Wang, "Hybrid graph disassembly model and sequence planning for product maintenance," in *Proc. Int. Technol. Innov. Conf.*, Jinan, China: Mech. Eng. Sch., Shandong Univ., 2006, pp. 515–5519. [Online] Available: [Online]. Available: https: //digitallibrary.theiet.org /content/conferences /10.1049/cp20060816

[12] F. Liu, S. Zhang, and Y. Zhang, "Product cooperative disassembly sequence planning based on branch-and-bound algorithm," *Int. J. Adv. Manuf. Technol.*, vol. 51, no. 9, pp. 1139–1147, 2010.

[13] K. Xia, L. Gao, W. Li, L. Wang, and K.-M. Chao, "A q-learning based selective disassembly planning service in the cloud based remanufacturing system for WEEE," in *Proc. ASME Int. Manuf. Sci. Eng. Conf.*, Detroit, MI, USA, 2014, pp. 9–13.

[14] S. A. Reveliotis, "Modelling and controlling uncertainty in optimal disassembly planning through reinforcement learning," in *Proc IEEE Int Conf Rob Autom.*, New Orleans, LA, USA, 2004, pp. 2625–2632.

[15] J. Wu, H. He, J. Peng, Y, Li, and Z, Li, "Continuous reinforcement learning of energy management with deep q network for a power split hybrid electric bus," *Appl. Energy.*, vol. 222, pp. 799–811, 2018.

[16] G. Tian, Y. Ren, Y. Feng, M. Zhou, H. Zhang, and J. Tan, "Modeling and planning for dual-objective selective disassembly using AND/OR graph and discrete artificial bee colony," *IEEE Trans. Ind. Inf.*, vol. 15, no. 4, pp. 2456–2468, 2018.

[17] C. Liu, X. Xu, and D. Hu, "Multi objective reinforcement learning: A comprehensive overview," *IEEE Trans. Syst. Man Cybern. -Syst.*, vol. 45, no. 3, pp. 385–398, 2014.

[18] P. Vamplew, R. Dazeley, A. Berry, R. Issabekov, and E. Dekker, "Empirical evaluation methods for multi objective reinforcement learning algorithms," *Mach. Learn.*, vol. 84, no. 1-2, pp. 51–80, 2011.

[19] Y. Cui, D. Zhang, T. Zhang, P. Yang and H. Zhu, "A new approach on task offloading scheduling for application of mobile edge computing," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Nanjing, China, 2021, pp. 1–6.

[20] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.

[21] K. Zheng, R.-J. Yang, H. Xu, and J. Hu, "A new distribution metric for comparing pareto optimal solutions," *Struct. Multidiscip. Optim.*, vol. 55, no. 1, pp. 53–62, 2017.