Stackelberg model-based human-robot collaboration in removing screws for product remanufacturing

Zhou, Yong; Peng, Yiqun; Li, Weidong; Pham, Duc

DOI:
10.1016/j.rcim.2022.102370

Document Version
Peer reviewed version

Citation for published version (Harvard):
Zhou, Y, Peng, Y, Li, W & Pham, D 2022, 'Stackelberg model-based human-robot collaboration in removing screws for product remanufacturing', *Robotics and Computer-Integrated Manufacturing*, vol. 77, 102370. https://doi.org/10.1016/j.rcim.2022.102370

Stackelberg Model-based Human-Robot Collaboration in Removing Screws for Product Remanufacturing

Y. Zhou[1], Y.Q. Peng[1], W.D. Li[2*], D.T. Pham[3]

[1] School of Transportation and Logistics Engineering, Wuhan University of Technology, China

[2] School of Mechanical Engineering, University of Shanghai for Science and Technology, China

[3] Department of Mechanical Engineering, University of Birmingham, U.K.

[*] Corresponding author: weidongli@usst.edu.cn

Abstract

Remanufacturing end-of-life (EoL) products is a critical step to effectively retrieve high-value components or materials from the products. Statistics show that unscrewing is one of major activities in remanufacturing. Human-robot collaboration (HRC) is a sensible strategy to leverage the strengths of human operators and robots to take off screws under various rust conditions. Nevertheless, the capabilities of HRC need to be enhanced in terms of cooperation, safety and disassembly efficiency. To address this issue, in this paper, an innovative HRC approach enabled by the Stackelberg model for removing screws in EoL products is developed. The approach can address the dynamic and uncertain characteristics of human operators to achieve human-centric HRC disassembly. In this approach, utilities for the Stackelberge model are represented by considering the disassembly efficiency and safety of humans and robots. An innovative Particle Swarm Optimisation (PSO)-Pareto algorithm (PSO-Pareto) is designed to achieve the best performance in terms of safety and disassembly efficiency. The effectiveness and generality of the approach were validated via experiments and case studies, and benchmarks with different designs proved the superiority of the approach.

Keywords: Human-robot collaboration (HRC), Stackelberg model, Optimisation.

1. Introduction

Disassembly is a vital process in remanufacturing to retrieve high-value components or materials from end-of-life (EoL) products. In current practice, almost all disassembly operations on EoL products are performed manually. Nevertheless, it is a labour intensive and costly process. In the meantime, EoL products usually contain hazardous and pollutant materials, which are harmful to human operators. Thus, manual disassembly has been becoming less favourite in modern industries and societies. The future trend is to develop robotic and intelligent technologies to facilitate automatic or semi-automatic disassembly processes [1-3].

Screws are essential joining components in products. Statistics show that 39.9% of disassembly operations are unscrewing [2]. Therefore, effective removal of screws from an EoL product is necessary to accomplish disassembly automation. Presently, investigations on the disassembly of screws from EOL products using robotics have been undertaken [6]. Most of

the developed approaches have been employed to disassemble screws that are under good conditions ("normal screws" in the following). Instead, rusted screws, which are inevitable on EoL products after their long service lifespans, are challenging to process by robotics. Human operators are apt to handle non-standard conditions such as rusted screws, but they are restricted in providing highly efficient operations. Therefore, it has stimulated the adoption of collaborative robots (cobots) in industries to empower robots to operate with humans jointly, i.e., human-robot collaboration (HRC), to meet sophisticated disassembly requirements. Presently, an imperative research issue is how to develop an effectual HRC strategy to leverage the strengths of cobots and humans to achieve disassembly flexibility and efficiency.

Game theory was originally designed to help solve economic decision-making problems. It was later successfully applied in a wide range of applications in finance, biology and computer science. In game theory, the Stackelberg model is an effective collaborative strategy [5]. In the model, a leader and a follower are specified during collaboration. The leader would choose his/her initial actions according to his/her own goals. The follower determines his/her actions accordingly to respond to the leader's actions. Then, both the leader and the follower adjust their actions iteratively to pursuit their best interests until a Nash equilibrium between them is achieved (a Nash equilibrium means that the leader or the follower does not gain a better result furthermore by changing his/her current actions). The Stackelberg model is appropriate for designing HRC-based disassembly on EoL products. The rationale is that during HRC-based disassembly processes, a human operator should play a leading role in actions, and a cobot should play a subsequent role to ensure human safety. After a Nash equilibrium is obtained in the model, the human and the cobot fulfil the best disassembly strategies and functionalities.

This paper presents an innovative approach for disassembling screws of EoL products based on the Stackelberg model. In the approach, a cobot under HRC would only disassemble normal screws, while a human operator under HRC would disassemble rusted screws and normal screws to leverage the flexibility of the human. Utility functions for the human's and the cobot's actions are determined. Critical factors contributing to the utility functions include moving distance of the human or the cobot required from the current job's location to the next job's location (the shorter, the more efficient in disassembly), the relative distance between the human and the cobot (for safety assurance between them), and their state continuity (to ensure efficiency and stability in disassembly). During HRC, the human would take an initial action. Then, the cobot would choose an action in the highest utility from its candidate actions that respond to the human's action. Furthermore, the human adjusts his/her action to achieve the highest utility under the cobot's new action. The above process is iteratively undertaken until a Nash equilibrium is obtained. Each utility is calculated by summing contributing factors using weights. In this research, an innovative Particle Swarm Optimisation (PSO)-Pareto algorithm is designed to identify the optimal weights of the utilities to ensure HRC safety and the best

overall disassembly efficiency. Research innovations are as follows:

- To the best of the authors' knowledge, this is the first study to devise the Stackelberg model-based approach to optimise HRC in EoL disassembly. The approach can address the dynamic and uncertain characteristics of the human to achieve human-centric disassembly via close and safe cooperation with the cobot;
- Experiments and benchmarks were conducted to demonstrate the effectiveness of the innovative design of the PSO-Pareto optimisation algorithm in improving the disassembly efficiency and safety of HRCs.

The remainder of this paper is organised as follows. Section 2 reviews the literature related to this work. In Sections 3 and 4, after the introduction of the HRC-based disassembly scenario, the research methodology is detailed. Section 5 describes experiments to validate the methodology and discusses the results obtained. Section 6 concludes the paper.

2. Related Work

2.1 Human-robot collaboration for disassembly

In recent years, HRC has become an actively investigated research topic due to its advantages of leveraging the characteristics of humans and robots (cobots) in operations. Michalos et al. [16] introduced the vision and architectures proposed by the EU project ROBO-PARTNER. The project was designed to aspire to combine cobot's strength, velocity, predictability, repeatability and precision with the human intelligence and skills to obtain a hybrid solution that would be involving the safe cooperation of human operators with autonomous and adapting robotic systems. Hjorth et al. [17] reviewed the current situation of HRC-based disassembly and analysed the principles and elements of HRC in the industrial environment, such as safety standards, collaborative operation modes, human-robot communication interfaces and the design characteristics of the disassembly process. Liu et al. [18] designed a system to manage HRC-based disassembly. Multi-modal perception of disassembly tasks, sequencing planning of the disassembly tasks, safety issues for the human operator in HRC, and motion driven control for the cobot were developed. Nevertheless, in the research, detailed disassembly processes were not specified for research validation. Xu et al. [19] developed a disassembly sequencing planner for HRC using a discrete Bees algorithm. In this research, first, feasible disassembly sequences for HRCs were generated. The sequences were then evaluated. Finally, an optimal disassembly sequence was determined to achieve minimised disassembly time, cost and difficulty. In addition, Xu et al. [20] designed an improved discrete Bees algorithm to solve the HRC-enabled disassembly line balancing problem, which aims to minimize the number of disassembly workstations, the smoothness index of disassembly time, and the demand index of components. In the research the safety of the human at work was considered through establishing the relationship between the cobot's

speed and HRC's distance. Parsa et al. [21] developed an HRC-based disassembly planning approach for EoL products. Evaluation criteria for remanufacturability on EoL products were defined, and suitable components to be disassembled were identified. The criteria include cleanability, repairability and economic values of the disassembled components. A genetic algorithm was employed to solve the optimal disassembly sequence based on the disassembly efficiency and cost. Li et al. [21] proposed an approach for HRC-based disassembly sequencing planning. A disassembly task was assigned to either a human operator or a cobot according to their characteristics and human fatigue. For instance, highly complex and/or highly flexible tasks were assigned to the human, while tasks with high repetitiveness or harmfulness to the human were dispatched to the cobot. Huang et al. [23] devised a cobot to dismantle press-fitted components originating from automotive water pumps. Active force control provided by the cobot was used to ensure that disassembly processes were performed flexibly. However, in the research, human factors in the processes were not considered fully, so the system's flexibility was still limited. Huang et al. [24] presented an experimental robotic disassembly cell comprising two cobots and a human operator. A case study was designed in the authors' autonomous remanufacturing laboratory, involving the disassembly of an automotive turbocharger, to validate the developed methodology. In the disassembly cell, sequential disassembly and parallel disassembly were combined to make efficient use of resources and to reduce disassembly time and cost. Force control by touching was developed for human-robotic interaction to trigger disassembly processes when necessary.

In summary, the reviewed research has mainly focused on operational sequence planning for HRC-based disassembly. Limited research has discussed closer collaboration between humans and cobots during disassembly, which should be an important issue to be addressed.

2.2 Removal of screws in EoL products

Wegener et al. [1] designed an HRC-based disassembly framework for automobile battery disassembly, in which a human operator conducted complex disassembly and a cobot was employed for taking simple and repetitive work. In the case study presented in the research, the human separated components joined with snap fits or glue and the cobot unfastened screws. When the position or orientation of a hexagonal screw in an EoL product was not accurately identified, Li et al. [6] designed an approach for automatic screw loosening. When a cobot's screwdriver reached a screw, a spiral search strategy was designed to ensure that the screwdriver head engaged the screw properly. However, this approach is only applicable to screws without defects on the surface and can be removed with standard disassembly tools. Yildiz et al. [7] developed an approach combining a Hough circle detection method and a deep convolutional neural network to detect screws. By comparison with several classification models, such as YOLO-v3, the achieved screw detection rate of this approach was the best, reaching 99%.

Difilippo et al. [8] devised an automated screw disassembly system that used a SOAR cognitive architecture to improve the system's performance. SOAR was employed to memorize the location of circles that contained screws, which is a useful function to decrease the trial time for a cobot to determine whether the circle is a screw. Cruz-Ramírez et al. [9] proposed an approach to successfully remove screws that hold the ceiling boards to the light gauge steel (LGS) in the construction industry. A hierarchical vision system detected the LGS, and then multi-template matching algorithm identified screws. Bdiwi et al. [10] designed an image processing algorithm for the automatic detection of motor screws. Screws were recognised based on their conditions, such as the greyscale, depth and HSV values.

The limitation of the approaches presented earlier is that they are only applicable to screws without rusted and damaged conditions. It is imperative to design a more flexible and robust approach to remove screws in EoL products that are under more complicated conditions.

2.3 Game theory for human-robot collaboration

Research combining game theory and HRC has been conducted in recent years, where game theory has been introduced to optimise cooperation between a human and a cobot. Liu et al. [11] developed an approach of task allocation for HRC-based manufacturing processes. In the approach, the tasks of a human and a cobot were defined based on a bilateral game, and then a clan game was used to calculate the execution sequence of the tasks. However, in the work, utility functions for the actions of the human and the cobot were not explicitly stated. Gabler et al. [12] designed an HRC-based game model, which described a scene where a human and a cobot interact within proximity in a shared workspace and generated an optimal task arrangement by using Nash equilibrium. Nevertheless, the presented model considered the human and the cobot equally, which might lead to unsafe risks of the human during HRC due to vicious competition. In a scenario where a human is operating a cobot for task learning, Li et al. [13] created a game theory-based model to calculate the control force of a cobot. A cost function was modelled to specify the contact force between a human and the cobot and the control force of the cobot. A dynamic Nash equilibrium was maintained to obtain the most appropriate cost functions and forces during HRC. In contrast to the cobot's adaptation to a human's actions, Nikolaidis et al. [14] designed a game theory-based model to ensure that the human is adaptive to the cobot. In the model, the reward functions of the human's and cobot's actions were continuously learned through HRC using game theory. However, it was unable to guarantee that the human would take an adaptive action in accordance with the reward function. Messeri et al. [15] developed a game-theoretic model that enables cobot to adapt its behaviour online to simultaneously optimise the human physiological stress and productivity in real-time, where the human's stress and productivity were quantified by using the heart-rate variability and the average working time in unit time period, respectively. The action-value functions of

the human and the cobot were built with the goal of minimising stress and maximising productivity, respectively. A Nash equilibrium solution was taken as the cobot's behaviour adjustment strategy.

For the aforementioned research, the roles of a human and a cobot in HRC have not yet been clearly differentiated. Considerations were not taken on how to regulate the human's action to improve the efficiency of HRC under the requirement that the cobot adapts to human's action. Therefore, a more effective game theory model needs to be explored to better support HRC-enabled disassembly of EoL products.

3. Stackelberg model-based Disassembly

3.1 The disassembly scenario

The disassembly scenario in this research is illustrated in Fig. 1, where screws on EoL products are removed using HRC. The rectangular area is the top cover of an EoL product, where screws under rusted conditions (in square) and normal conditions (in circle) for removal are distributed. A human operator ("the human") and a cobot ("the cobot") would cooperate to perform screw removal. In this research, it is assumed that the human can move flexibly to reach any screw in the scenario. The two circles in Fig. 1 indicate the safety ranges of the human when removing a screw. The area between two arcs represents the working range of the cobot when it is located in the upper right corner.

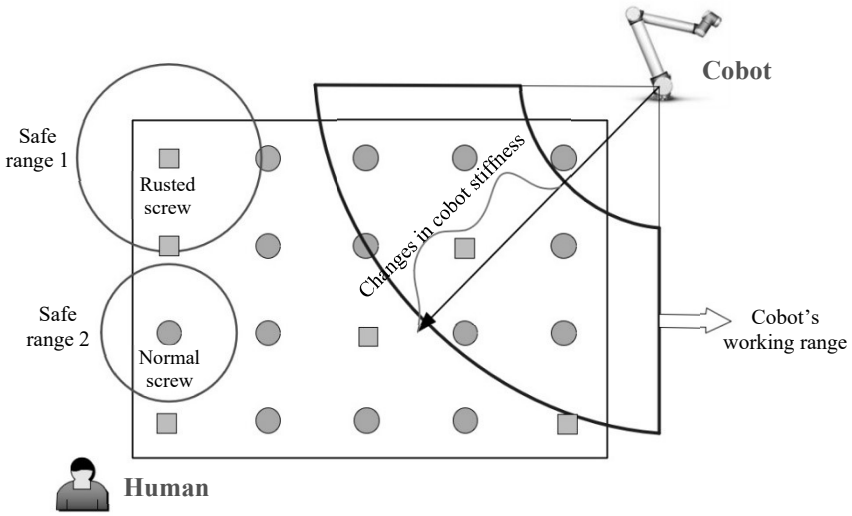


Fig. 1: Removal of screws in an EoL product.

3.1.1 Screw classification

Some screws in an EoL product could be corroded and rusted (rusted screws). Others may still be well preserved and have clean surfaces (normal screws). A rusted screw is more difficult to remove than a normal screw, as greater preloading force and friction are necessary. The removal of screws by the cobot is usually accomplished with an electric screwdriver mounted on the end-effector of the cobot. Nevertheless, the cobot is not suitable for rusted screw removal,

as the required operation torque is usually greater than the maximum torque exerted by the cobot. On the other hand, humans can remove rusted screws when assisted by external tools (e.g., a hammer to loosen and remove rusted screws). Thus, in this research, the cobot was only used to disassemble normal screws, and humans disassembled both normal screws and rusted screws.

3.1.2 Safety ranges of the human

During disassembly, the human and the cobot operate simultaneously under the same space. According to the ISO/TS 15066, to ensure the safety of the human, it is essential to set a safety sphere centred on the human ("safety range") where any part of the cobot is restricted not to interfere during HRC. A safety range for the human is determined by the maximum of the human's change in location. Safety ranges of the human during disassembly are defined below:

- Safety range I: For disassembling a normal screw, the human only needs to use one of his/her hands to operate an electric screwdriver to take off a screw, and he/she basically stays still;
- Safety range II: For disassembling a rusted screw, the human needs to engage both hands to complete the disassembly task by using an electric screwdriver and assisted tools, and he/she would take a greater space to adjust his/her posture in order to remove the rusted screw more conveniently.

Therefore, the safety range for safety range II is greater than that for safety range I. The ranges are specified below:

$$SR_2 = k_s * SR_1 \tag{1}$$

where SR_1 and SR_2 are the safety ranges of the human; k_s is a preset value (greater than 1).

3.1.3 States of the human and the cobot

A state reflects the condition of the human or the cobot at a certain time. An action is to change or maintain the current state of the human or the cobot throughout disassembly. The definitions of the states are given in Table 1. Fig. 2 illustrates two states of the cobot.

Table 1: States for the human and the cobot.

State		Definition
Human	$state_{h-dis}$	The human is disassembling a screw
	$state_{h-mov}$	The human is moving his/her location
	$state_{h-sta}$	The human is in standby and waits for the next disassembly job
	$state_{h-end}$	The human finishes all his/her disassembly jobs
Cobot	$state_{c-rep}$	The cobot is fine-tuning the posture of its end-effect in order to disassemble a screw
	$state_{c-dis}$	The cobot is disassembling a screw
	$state_{c-mov}$	The cobot is moving its location

	$state_{c-sta}$	The cobot is in standby and waits for the next disassembly job
	$state_{c-end}$	The cobot finishes all its disassembly jobs

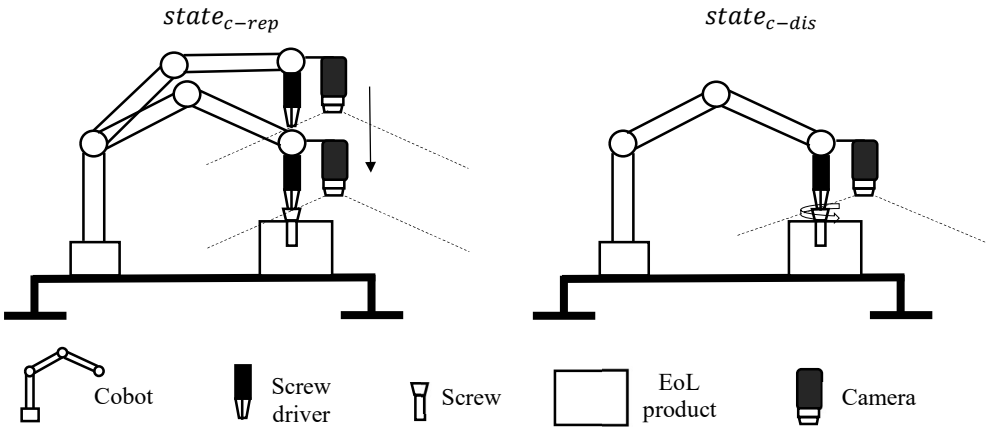


Fig. 2: Some states of the cobot in disassembly operations.

3.2 Methodology for human-robot collaborative disassembly

Humans are characterised by dynamics and uncertainties in actions. To ensure the human-centric role in HRCs and the safety of humans, in this research, a Stackelberg model is designed [4]. In the model, the utilities of the human (leader) and the cobot (follower) are determined by their states and mutual actions. The definitions in the model are given below:

- The actions for the human and cobot should satisfy constraints: (1) the relative distance between the human and the cobot needs to be greater than the safety range of the human; (2) the screw for removal by the cobot should be within the working range of the cobot;
- The action set that the human can take is $\boldsymbol{A_h}$, and the action set that the cobot can take is $\boldsymbol{A_c}$. $|\boldsymbol{A_h}|$ represents the total number of the human's actions, and $|\boldsymbol{A_c}|$ represents the total number of the cobot's actions;
- a_{h-i} represents the action that the human is currently taking ($a_{h-i} \in \boldsymbol{A_h}$, $1 \le i \le |\boldsymbol{A_h}|$). a_{c-j} represents the action that the cobot is currently taking ($a_{c-j} \in \boldsymbol{A_c}$, $1 \le j \le |\boldsymbol{A_c}|$);
- $U_h(a_{h-i}, a_{c-j})$ and $U_c(a_{h-i}, a_{c-j})$ represent the utilities of the human and the cobot, respectively, when the human takes a_{h-i} and the cobot takes a_{c-j}.

The HRC process of the Stackelberg model is below:

1. $a_{c-j_i'}$ is denoted as an action of the cobot with the highest utility when the human takes action a_{h-i}. A set $\left\{(a_{h-i}, a_{c-j_i'}) \middle| i \in (1, \dots, |\boldsymbol{A_h}|)\right\}$ is determined;
2. $a_{h-i'}$ is denoted as the action of the human with the highest utility in the set of $\{a_{h-i} | i \in (1, \dots, |\boldsymbol{A_h}|)\}$;
3. $(a_{h-i'}, a_{c-j_{i'}'})$, which satisfies a Nash equilibrium as follows:

$$U_h(a_{h-i'}, a_{c-j_{i'}'}) \ge U_h(a_{h-i}, a_{c-j_i'}), \begin{matrix} 1 \le i, i' \le |\boldsymbol{A_h}|, 1 \le j_i', j_{i'}' \le |\boldsymbol{A_c}| \\ i \ne i' \end{matrix} \quad (2)$$

$$U_c(a_{h-i'}, a_{c-j'_{i'}}) \geq U_c(a_{h-i'}, a_{c-j}), \quad \begin{matrix} 1 \leq i' \leq |\boldsymbol{A_h}|, 1 \leq j, j'_{i'} \leq |\boldsymbol{A_c}| \\ j \neq j'_{i'} \end{matrix} \tag{3}$$

The flowchart of the Stackelberg model is detailed in Fig. 3. The utilities of the human and the cobot are calculated in the following subsections.

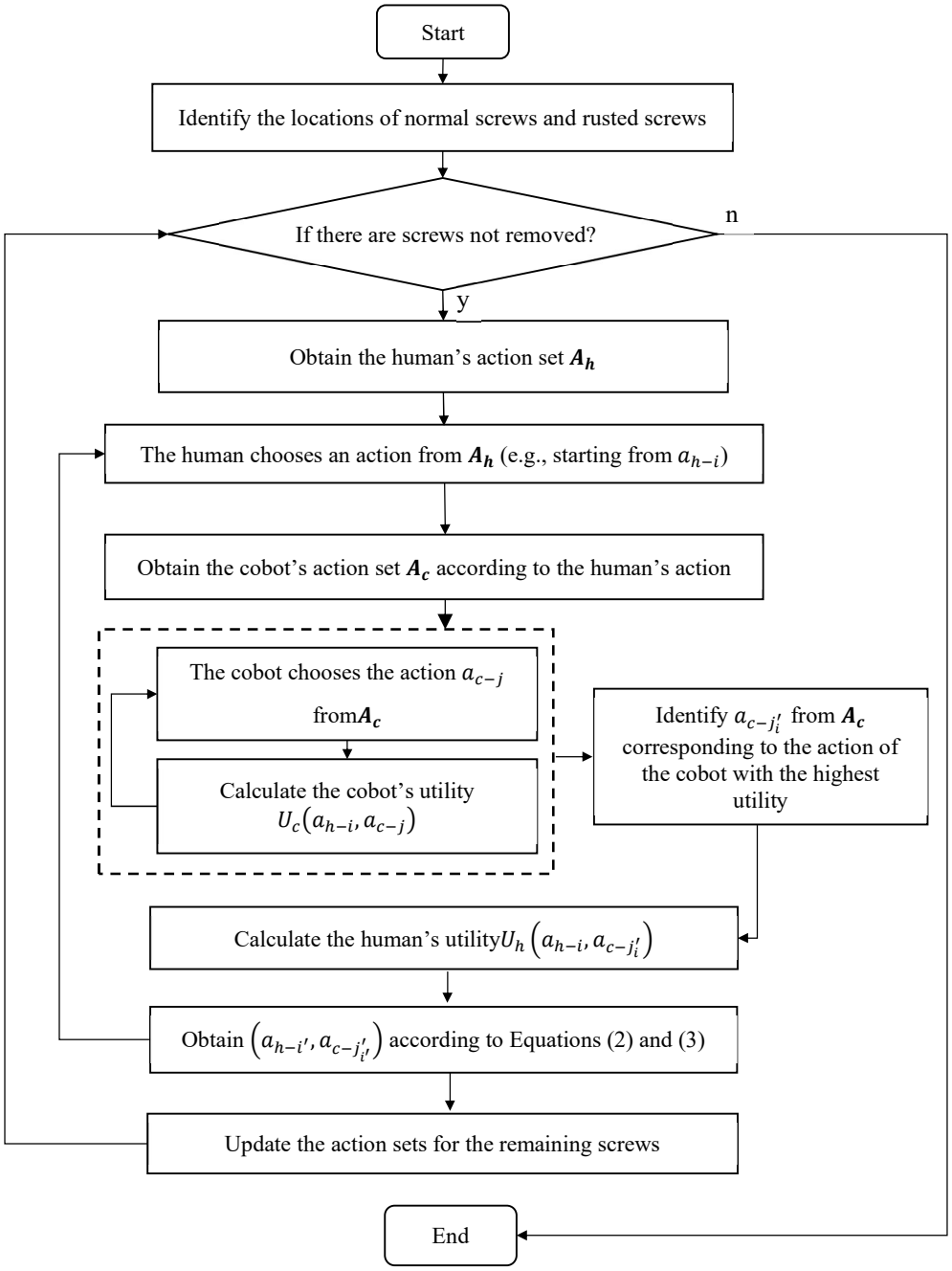


Fig. 3: The Stackelberg model for HRC-based disassembly.

3.2.1 The utility of the human

The utility of the human, $U_h(a_{h-i}, a_{c-j})$, is related to three factors: safety between the human and cobot, disassembly efficiency and state continuity. The factors could be contradictory so that they should be considered combinedly.

During the disassembly process, the human prefers to work away from the cobot. The farther the human-robot distance is, the safer the human will feel (less safety anxiety). Thus, the safety feeling of the human is directly determined by the distance between the human and the cobot (the minimum safety ranges of the human and the cobot must be met as a hard constraint). The larger the distance, the greater the safety felling is (smaller the safety anxiety). However, a too greater distance between the human and the cobot deteriorates the disassembly efficiency of the human. Disassembly efficiency is determined by the required moving distance of the human from his/her current location to that of the screw for removal. The shorter the distance is, the higher the disassembly efficiency. State continuity means whether the human continues or abandons his/her current state. The utilities of the human's safety feeling and disassembly efficiency ($U_{h-safety}$ and $U_{h-efficiency}$) are calculated according to Equations (4) and (5), respectively:

$$U_{h-safety}(a_{h-i}, a_{c-j}) = \frac{D_{h-i/c-j} - min(\boldsymbol{D}_{hc})}{max(\boldsymbol{D}_{hc}) - mi\ (\boldsymbol{D}_{hc})} \quad (4)$$

where $D_{h-i/c-j}$ is the relative distance between the human and the cobot when they take the action set (a_{h-i}, a_{c-j}) and $\boldsymbol{D}_{hc} = \{D_{h-i/c-j}(1 \le i \le |\boldsymbol{A_h}|, 1 \le j \le |\boldsymbol{A_c}|)\}$.

$$U_{h-efficiency}(a_{h-i}, a_{c-j}) = -\frac{D_{h-i} - min(\boldsymbol{D}_{h-*})}{max(\boldsymbol{D}_{h-*}) - min(\boldsymbol{D}_{h-*})} \quad (5)$$

where D_{h-i} is the required moving distance of the human to the screw for removal when he/she takes a_{h-i}; $max(\boldsymbol{D}_{h-*})$ and $min(\boldsymbol{D}_{h-*})$ are the maximum and minimum of the human's moving distances to the screw for removal when he/she takes an action from D_{h-i} $(1 \le i \le |\boldsymbol{A_h}|)$.

The values of $U_{h-state}$ (the utility for the human's state continuity) are defined in Table 3.

Table 3: $U_{h-state}$ when the human changes from a state (in column) to another state (in row).

	$state_{h-mov}$	$state_{h-sta}$	$state_{h-dis}$	$state_{h-end}$
$state_{h-mov}$	0	1	-1	-
$state_{h-sta}$	-	-1	-	-
$state_{h-dis}$	-	1	1	-
$state_{h-end}$	0	0	-1	0

Under the stochastic optimisation process of the above Stackelberg model, a potential negative issue is that the human could pick up easier tasks to accomplish (i.e., remove normal screws), which compromises the overall optimisation convergence of the Stackelberg model for HRC. Thus, to accelerate the optimisation process by ensuring the human chooses more rusted screws, a penalty utility ($U_{h-penalty-1}$) will be applied to the utility of the human's action if the current screw for removal is a normal screw. In addition, some of the human actions may interrupt the current work of the cobot and affect the overall disassembly efficiency. Thus, to avoid interfering with the current work of the cobot as much as possible, another penalty utility ($U_{h-penalty-2}$) will be added to the utility of the human's action. $U_{h-penalty-}$ and $U_{h-penalty-2}$

are defined below, respectively:

$$U_{h-penalty-} = \begin{cases} -1 & if\ the\ human\ removes\ a\ normal\ screw \\ 0 & otherwise \end{cases} \quad (6)$$

$$U_{h-penalty-2} = \begin{cases} U_{c-sta} & if\ U_{c-sta} < 0 \\ 0 & otherwise \end{cases} \quad (7)$$

The definition of $U_{c-state}$ will be explained in the next subsection.

$U_h(a_{h-i}, a_{c-j})$ is defined below:

$$U_h(a_{h-i}, a_{c-j}) = w_{h-1} * U_{h-safety}(a_{h-i}, a_{c-j}) + w_{h-2} * U_{h-efficiency}(a_{h-i}, a_{c-j}) + w_{h-3} * U_{h-state} + w_{h-4} * (0.5 * U_{h-penalty-} + 0.5 * U_{h-penalty-}) \quad (8)$$

where w_{h-1}-w_{h-4} ($\sum_{i=1}^{4} w_{h-i} = 1$) are weights to normalise the four utilities.

3.2.2 The utility of the cobot

The utility of the cobot, $U_c(a_{h-i}, a_{c-j})$, is related to three factors, namely, safety feeling of the human, disassembly efficiency and state continuity.

The utility of the cobot's safety ($U_{c-safety}$) is the same as that for the human defined in Equation (4):

$$U_{c-safety}(a_{h-i}, a_{c-j}) = \frac{D_{h-i/c-j} - min(\boldsymbol{D}_{hc})}{max(\boldsymbol{D}_{hc}) - min(\boldsymbol{D}_{hc})} \quad (9)$$

Disassembly efficiency is determined by the required moving distance of the end-effector of the cobot to a screw for removal. The utility for the cobot's disassembly efficiency ($U_{c-efficien}$) is calculated below:

$$U_{c-efficiency}(a_{h-i}, a_{c-j}) = -\frac{D_{c-j} - min\,(\boldsymbol{D}_{c-*})}{max(\boldsymbol{D}_{c-*}) - min\,(\boldsymbol{D}_{c-*})} \quad (10)$$

where D_{c-j} is the required moving distance of the cobot to the screw for removal; $\boldsymbol{D}_{c-*} = \{D_{c-j} | 1 \le j \le |A_c|\}$; and $max(\boldsymbol{D}_{c-*})$ and $min(\boldsymbol{D}_{c-*})$ are the maximum and minimum of the moving distances of the cobot to the screw for removal when the cobot takes an action from D_{c-i} ($1 \le i \le |\boldsymbol{A_h}|$).

The values of $U_{c-state}$ (the utility for the cobot's state continuity) are defined in Table 4.

Table 4: $U_{c-state}$ when the cobot changes from a state (in column) to another state (in row).

	$state_{c-mov}$	$state_{c-sta}$	$state_{c-dis}$	$state_{c-rep}$	$state_{c-end}$
$state_{c-mov}$	0	1	-1	-1/2	-
$state_{c-sta}$	-	-1	-	-	-
$state_{c-dis}$	-	1	1	-	-
$state_{c-rep}$	-	1	-	½	-
$state_{c-end}$	0	0	-1	-1/2	0

$U_c(a_{h-i}, a_{c-j})$ is defined below:

$$U_c(a_{h-i}, a_{c-j}) = w_{c-1} * U_{c-safety}(a_{h-i}, a_{c-j}) + w_{c-2} * U_{c-efficiency}(a_{h-i}, a_{c-j}) +$$

$w_{c-3} * U_{c-sta}$ (11)

where w_{c-1}-w_{c-3} ($\sum_{j=1}^{3} w_{c-j} = 1$) are weights used to normalise the cobot's utilities.

4. Optimisation Strategy for HRC Disassembly

4.1 Evaluation criteria for HRC

In this research, the following evaluation criteria for the disassembly process are specified.

Total time (*T*): *T* is the total disassembly time from removing the first screw to removing the last screw on an EoL product via HRC. The smaller *T is*, the higher the disassembly efficiency.

The safety index (*S*): In HRCs, it is necessary to maintain a safe distance between the human and the cobot. Apart from safety assurance, distance can relieve the human's mental pressure during collaboration with the cobot. To better represent safety during the entire disassembly process, a safety index (*S*) is defined below.

$$S = \frac{1}{T} * \sum_t \left(\frac{SR_t}{D_t}\right) \quad (12)$$

where SR_t is the safety range of the human at time t and D_t denotes the relative distance between the human and the cobot at time *t*.

The smaller the safety index is, the farther the average distance between the human and the cobot, and the safer the HRC.

4.2 Optimisation for the weights of utilities

In the utilities described in Section 3.2, different values of weights ($w_{h-1}, w_{h-2}, w_{h-3}, w_{h-4}, w_{c-1}, w_{c-2}, w_{c-3}$) would affect the disassembly efficiency (represented in *T*) and safety (represented in *S*).

T and *S* are considered optimisation objectives, and weights are variables. Pareto optimisation is employed here, as it is a sensible strategy to resolve multi-objective optimisation problems [25]. In the meantime, an effective algorithm is needed to search for feasible solutions to support Pareto optimisation. Particle Swarm Optimisation (PSO) is a population-based stochastic optimisation technique inspired by the social behaviour of bird flocking or fish schooling. PSO has been proven to perform well in continuous optimisation problems with constraints [26]. In this research, Pareto and PSO are combined as a PSO-Pareto algorithm to optimise the weights of utilities for HRC-based disassembly.

In the algorithm, $\boldsymbol{w_i(k)} = (w_{h-1}, w_{h-2}, w_{h-3}, w_{h-4}, w_{c-1}, w_{c-2}, w_{c-3})$ represents the *i*-th particle in the population $\boldsymbol{S}$ ($(\boldsymbol{w_1(1)}$,…,$\boldsymbol{w_1(k)}$,…, $\boldsymbol{w_1(m)})$,…,($\boldsymbol{w_n(1)}$,…,$\boldsymbol{w_n(k)}$,…, $\boldsymbol{w_n(m)})$) under the *k*-th iteration ($1 \leq \boldsymbol{k} \leq \boldsymbol{m}$) during optimisation. As shown in the following formula, the objectives of minimising *T* and *S* are defined below:

$$min\, f(\boldsymbol{w_i(k)}) = min[T(\boldsymbol{w_i(k)}), S(\boldsymbol{w_i(k)})\,] \quad (13)$$

where $T(\boldsymbol{w_i(k)})$ and $S(\boldsymbol{w_i(k)})$ represent T and S for $\boldsymbol{w_i(k)}$ respectively.

For $\boldsymbol{w_1}$ and $\boldsymbol{w_2}$, $\boldsymbol{w_2}$ is dominated by $\boldsymbol{w_1}$ if one of the following relations is satisfied:

$$T(\boldsymbol{w_1}) \leq T(\boldsymbol{w_2}) \text{ and } S(\boldsymbol{w_1}) < S(\boldsymbol{w_2}) \tag{14}$$

$$T(\boldsymbol{w_1}) < T(\boldsymbol{w_2}) \text{ and } S(\boldsymbol{w_1}) \leq S(\boldsymbol{w_2}) \tag{15}$$

$$T(\boldsymbol{w_1}) < T(\boldsymbol{w_2}) \text{ and } S(\boldsymbol{w_1}) < S(\boldsymbol{w_2}) \tag{16}$$

Pareto optimal solutions (***PS***) refer to the particles that are not dominated by other particles. The values of their objectives (T and S) are Pareto fronts (***PF***). The details of the algorithm are described below.

4.2.1 Particle instantiation

The particle instantiation and updating processes of the PSO-Pareto algorithm are depicted in Fig. 4. There are two attributes for each particle, i.e., position and velocity. The position can be represented using $\boldsymbol{w_i(k)} = (w_{h-1}, w_{h-2}, w_{h-3}, w_{h-4}, w_{c-1}, w_{c-2}, w_{c-3})$, which needs to satisfy two constraints, i.e., $\sum_{i=1}^{4} w_{h-i} = 1$ and $\sum_{j=1}^{3} w_{c-j} = 1$. The velocity refers to the magnitude and direction of the changes in the weights.

$$\boldsymbol{w_i(k)'} = \boldsymbol{w_i(k)} + \boldsymbol{v_i(k)} \tag{17}$$

where $\boldsymbol{w_i(k)}$ is the position of a particle before the change; $\boldsymbol{w_i(k)'}$ is the position of the particle after the change; and $\boldsymbol{v_i(k)}$ is the velocity.

According to Equation (17), $\sum \boldsymbol{w_i(k)'} = \sum \boldsymbol{w_i(k)} + \sum \boldsymbol{v_i(k)}$, where $\sum \boldsymbol{w_i(k)'}$ represents the sum of weights after the change. As $\sum \boldsymbol{w_i(k)'} = \sum \boldsymbol{w_i(k)}$, the initialisation of the velocity needs to meet the constraint $\sum \boldsymbol{v_i(k)} = 0$, i.e., $\sum_{i=1}^{4} v_{h-i} = 0$ and $\sum_{j=1}^{3} v_{c-j} = 0$.

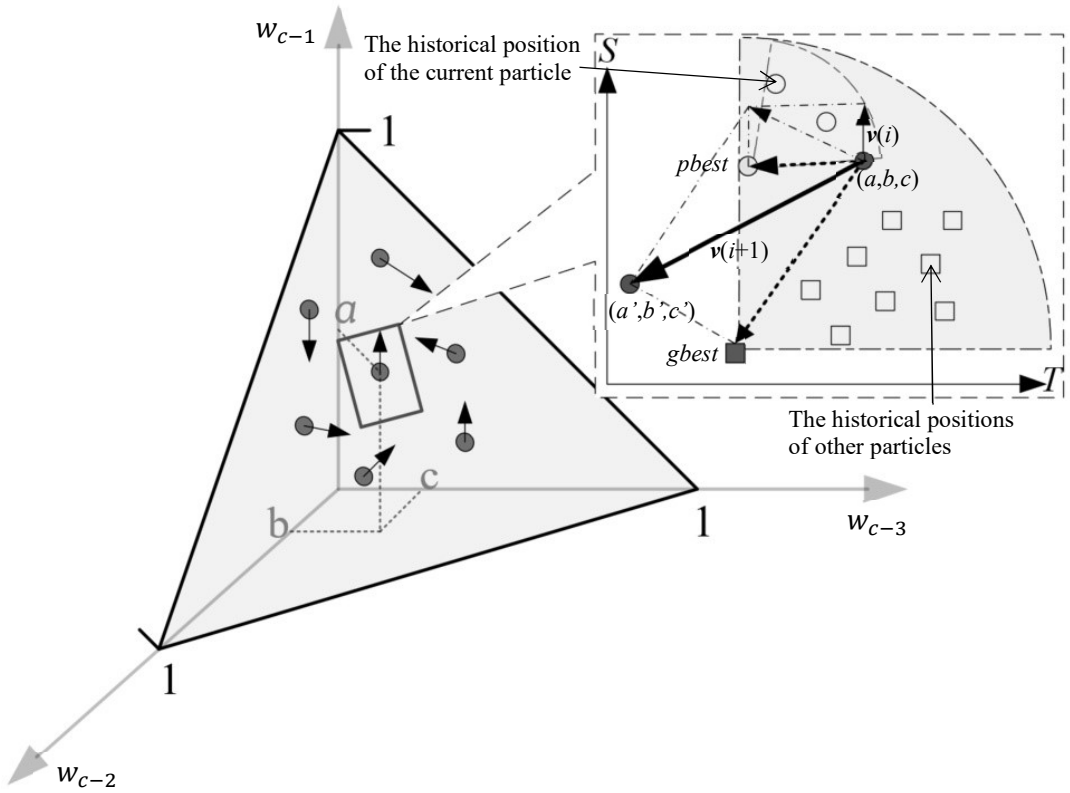


Fig. 4: The PSO-Pareto algorithm (the figure only shows the updating process of w_{c-j}. Each particle is positioned on the triangular plane to meet the constraint $\sum_{j=1}^{3} w_{c-j} = 1$. The position of a particle is the coordinates. The velocity of a particle is a vector parallel to the triangular plane to meet the constraint $\sum_{j=1}^{3} v_{c-j} = 0$. (*a,b,c*) and $\boldsymbol{v}(i)$ are the position and velocity of the particle at iteration *i*. (*a',b',c'*) and $\boldsymbol{v}(i+1)$ are the position and velocity of a particle at iteration *i*+1. *pBest* refers to the Pareto optimal position in the history of the particle, and *gBest* refers to the Pareto optimal position in the history of all the particles.)

4.2.2 Determination of gBest and pBest

pBest refers to the Pareto optimal position of a particle in its history, and *gBest* refers to the Pareto optimal position of all the particles in their history. To obtain *pBest* and *gBest*, the following two strategies are adopted, and Fig. 5 is used to illustrate the process.

The filtering strategy: For a set of particles ***S***, the Pareto optimal solution set ***PS*** obtained from ***S*** is shown in triangles, and other particles are in squares. For a particle $\boldsymbol{w_i(k)}$ (represented as a square in black), dominated particles are filtered out from ***PS*** according to Equations (14)-(16) as a new set $\boldsymbol{PS}_m$. The solutions in triangles in the dashed box belong to $\boldsymbol{PS}_m$.

Particle density strategy: The solution space is divided into rectangular grids. The length and width of the grid are set as *dt* and *ds*. The density of a particle is defined as the number of particles in the same grid.

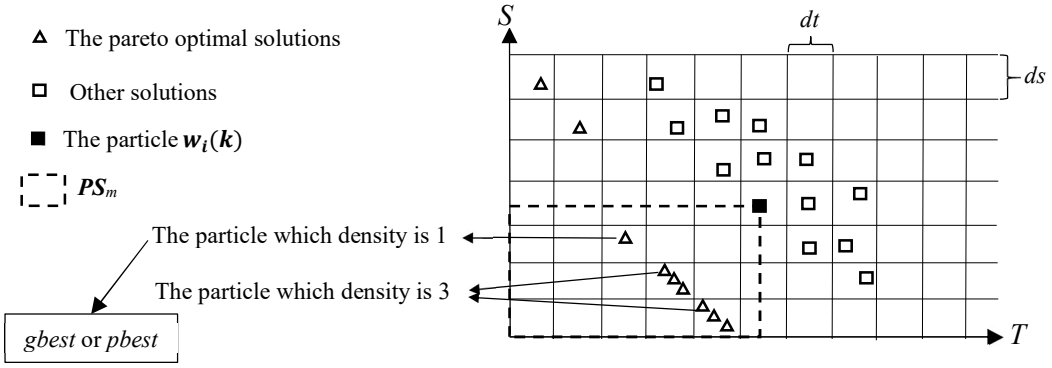


Fig. 5: The process of identifying *gbest* and *pBest* of a particle.

The algorithm is an iterative process towards optimal solutions, and the current particle is represented as $\boldsymbol{w_i(k)}$. Since particle $\boldsymbol{w_i(k)}$ tends to move towards *gBest* during the next iteration, to avoid local convergence, the particle with the lowest density in $\boldsymbol{PS}_m$ is chosen as *gBest* to explore more spaces in the following search process. The determination of *pBest* for particle $\boldsymbol{w_i(k)}$ is similar to that of *gBest* except that the Pareto optimal solution set $\boldsymbol{PS}$ is obtained from the historical solution set of particle $\boldsymbol{w_i(k)}$.

4.2.3 Particle updating strategy

In each iteration, the position and velocity of each particle are adjusted according to its *gBest* and *pBest* as follows:

$$\boldsymbol{v_i(k+1)} = c * \boldsymbol{v_i(k)} + c_1 * r_1 * (pBest - \boldsymbol{w_i(k)}) + c_2 * r_2 * (gBest - \boldsymbol{w_i(k)}) \quad (18)$$

$$\boldsymbol{w_i(k+1)} = \boldsymbol{w_i(k)} + \boldsymbol{v_i(k)} \quad (19)$$

where c is an inertia factor ($0 < c < 1$); c_1 and c_2 are acceleration factors; and r_1 and r_2 are random numbers in the interval [0, 1].

A larger c is beneficial, as the algorithm can jump out from a local minimum point and facilitate a global search. A smaller c is useful to fine-tune a local search to ensure the algorithm's convergence. Thus, to optimise the process, a linearly changing inertia factor can be used, i.e., let c decrease from the maximum value to the minimum value in a linear way. The formula for the change of c is as follows：

$$c = c_{max} - \frac{k*(c_{max}-c_{min})}{m} \quad (20)$$

where c_{max} and c_{min} represent the maximum and minimum values of c, respectively; k represents the current iteration step; and m represents the maximum number of iteration steps.

Based on the above procedures, the flowchart of the algorithm is described in Fig. 6.

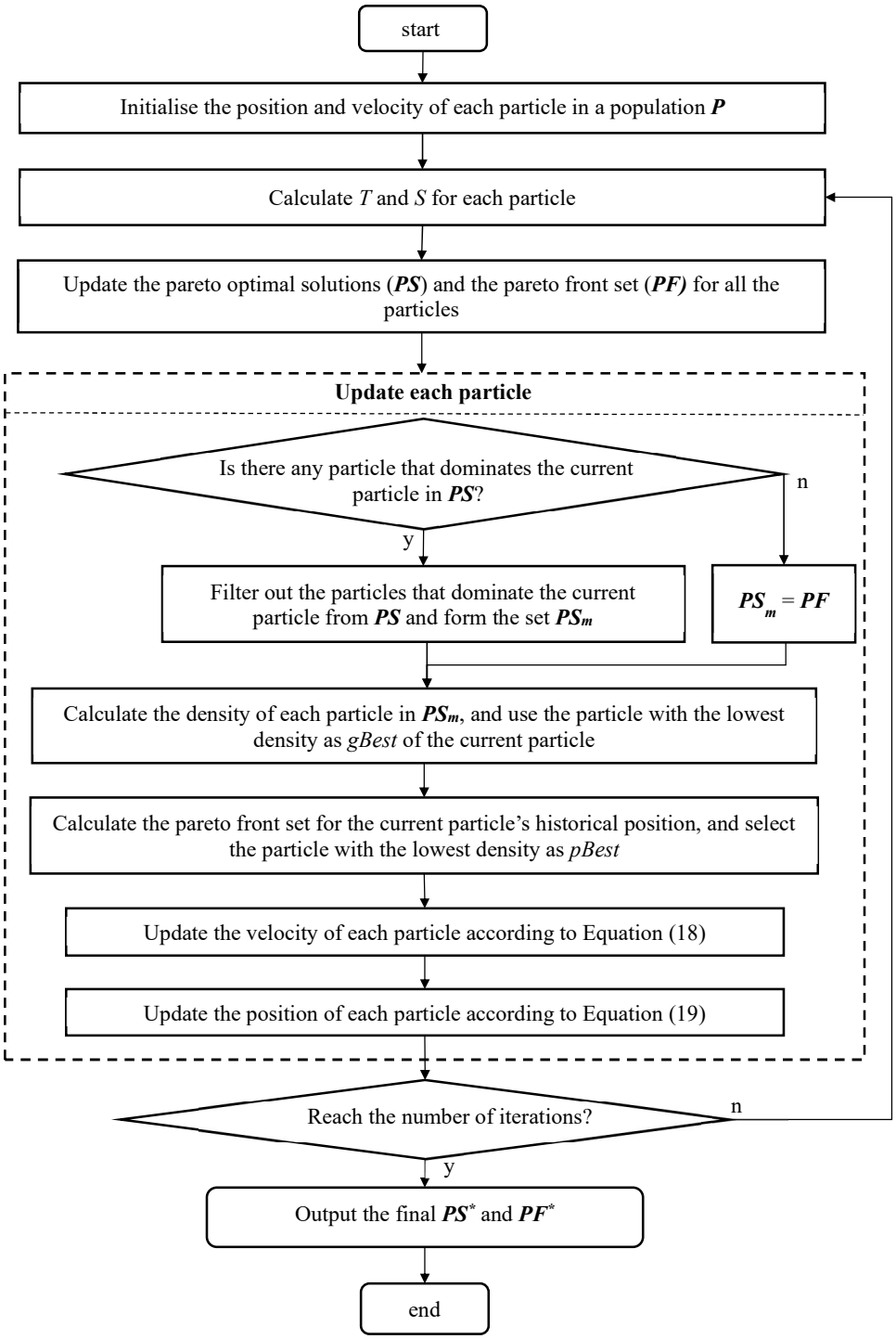


Fig. 6: The flowchart of the PSO-Pareto algorithm.

5. Validation Experiments for the Approach

5.1 Case studies

5.1.1 Descriptions of case studies

The conditions of screws for disassembling (rusted or normal), the spatial relationship between the screws, the human and the cobot, and the working state of the cobot will impact

on HRC. As shown in Table 5, some cases are established to demonstrate the process and effectiveness of the research presented in this paper according to the above factors, respectively (in each case there is a human operator, a cobot and three/four screws denoted as S_1, S_2, S_3/S_4). The parameters in the case are given in Table 6.

Table 5: The description of case studies.

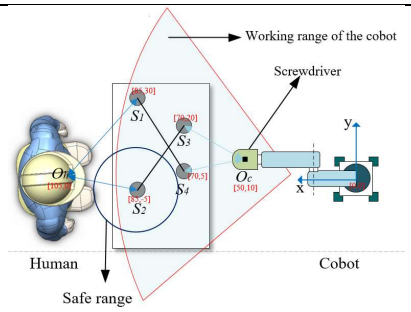
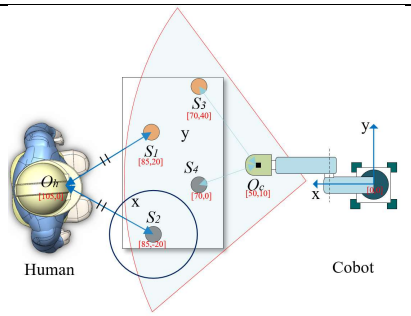
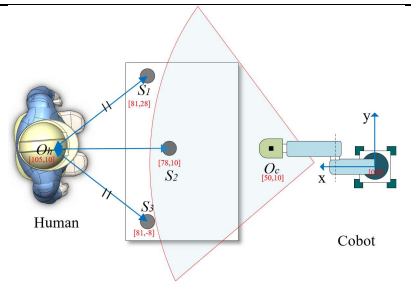
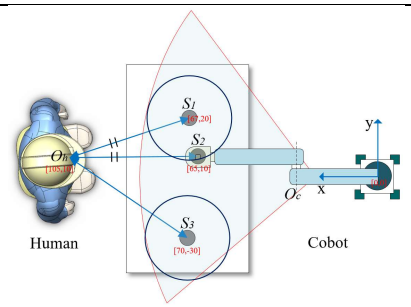
Cases	Illustration	Description
Case 1		• All screws are in the working range of the cobot • No rusted screw • $D_{h-S_2} < D_{h-S_1}$, $D_{c-S_4} < D_{h-S_3}$ • $U_{h-safety}(S_1, S_4) = U_{h-safety}(S_2, S_3)$
Case 2		• The screws are in the working range of the cobot • S_1 and S_3 are rusted • $D_{h-S_2} = D_{h-S_1}$
Case 3		• S_1 and S_3 are not in the working range of the cobot • No rusted screw • $D_{h-S_1} = D_{h-S_3} > D_{h-S_2}$
Case 4		• The screws are in the working range of the cobot • No rusted screws • The cobot is disassembling $S2$ • $D_{h-S_1} = D_{h-S_2} < D_{h-S_3}$

Table 6: Parameters in the case study.

Parameter	Time
The speed of the human for disassembling a rusted screw	8s
The speed of the human for disassembling a normal screw	6s

The speed of the cobot for disassembling a normal screw	2s
The speed of the cobot for repositioning a normal screw	2s
Moving speed of the hand of the human for disassembly	10 cm/s
Moving speed of the end-effector of the cobot for disassembly	5 cm/s
Safe range I of the human	15 cm
Safe range II of the human	25 cm

5.1.2 Comparative analysis for the Stackelberg approach and the non-game approach

To demonstrate the effectiveness of the Stackelberg model-based approach, a comparison was made with a non-game theory-enabled approach ("non-game approach" in the following). The non-game approach refers to a situation in which both the human and the cobot take the most profitable actions only for him/her/its-self regardless of considering each other's action (illustrated in Fig. 7).

In the non-game approach, the utilities of the human and the cobot are re-defined, denoted as $U'_h(a_{h-i}, a_{c-j})$ and $U'_c(a_{h-i}, a_{c-j})$, respectively. Different from the defined utilities of the human in the Stackelberg model-based approach where the penalty items (for the interactions/collaborations between the human and the cobot) are considered, the penalty items are not considered in the non-game approach, so that the formulas of $U'_h(a_{h-i}, a_{c-j})$ and $U'_c(a_{h-i}, a_{c-j})$ are represented as follows:

$$U'_h(a_{h-i}, a_{c-j}) = w_{h-1} * U_{h-safety} + w_{h-2} * U_{h-efficiency} + w_{h-3} * U_{h-sta} \quad (21)$$

$$U'_c(a_{h-i}, a_{c-j}) = U_c(a_{h-i}, a_{c-j}) \quad (22)$$

As shown in Fig. 7, it is assumed that the action set of the human in the non-game approach is $\{a_{h-1}, \dots, a_{h-m}\}$, and the action set of the cobot is $\{a_{c-1}, \dots, a_{c-n}\}$. It is assumed that when the action combination (a_{h-k}, a_{c-k}) is selected by the human and the cobot, the human obtains the highest utility; however, when the action combination (a_{h-l}, a_{c-l}) is selected, the cobot has the highest utility. That is:

$$(a_{h-k}, a_{c-k}) = \underset{1 \le i \le m, 1 \le j \le n}{\operatorname{argmax}} U'_h(a_{h-i}, a_{c-j}) \quad (23)$$

$$(a_{h-l}, a_{c-l}) = \underset{1 \le i \le m, 1 \le j \le n}{\operatorname{argmax}} U'_c(a_{h-i}, a_{c-j}) \quad (24)$$

For the non-game approach, based on the above processes, the optimal actions chosen by the human and the cobot will be (a_{h-k}, a_{c-l}).

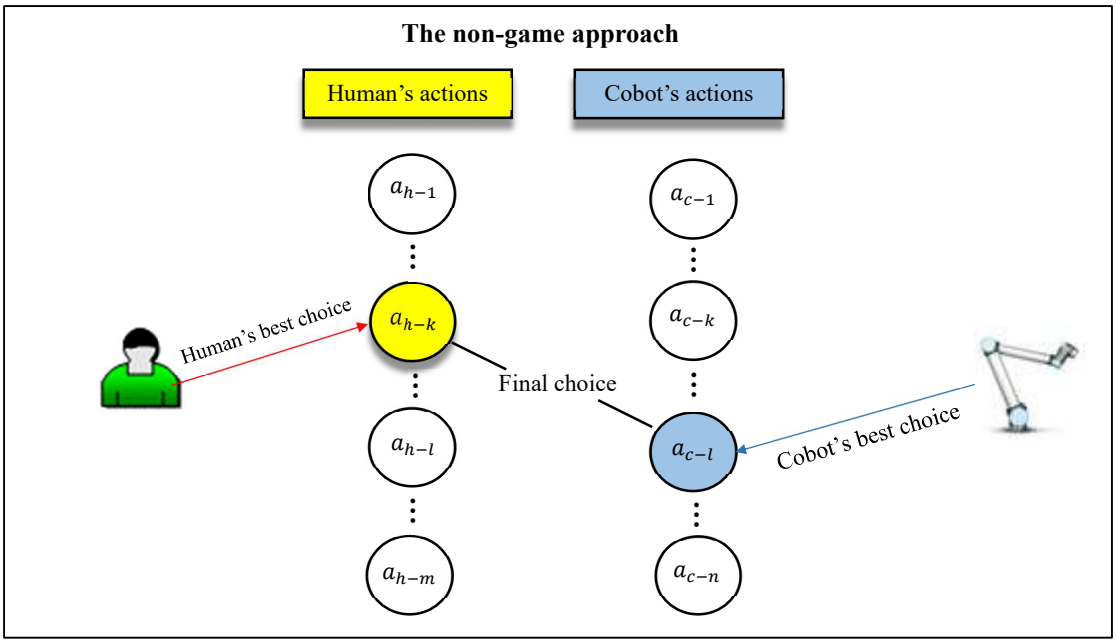


Fig. 7: The description of the non-game approach.

The four cases presented in Table 5 are analysed and compared by using the Stackelberg approach and the non-game approach, respectively (it is assumed that the weight values in the utilities of the human and cobot in both approaches are the same). The results are show in Table 7 ($S_i(\mathrm{H})$ means that the human removes S_i, $S_i(\mathrm{C})$ means that the cobot removes S_i).

Table 7: The analysis and comparative results of the approaches for the four cases.

Cases	The Stackelberg approach	The non-game approach
Case 1	Disassembly Process: $S_2(\mathrm{H}) \rightarrow S_3(\mathrm{C}) \rightarrow S_4(\mathrm{C}) \rightarrow S_1(\mathrm{H})$ ***T*: 16.7** *S*: 0.63	Disassembly Process: $S_2(\mathrm{H}) \rightarrow S_4(\mathrm{C}) \rightarrow S_3(\mathrm{H}) \rightarrow S_1(\mathrm{C})$ The cobot enters the safe range of the human. Thus, the task fails.
Case 2	Disassembly Process: $S_1(\mathrm{H}) \rightarrow S_4(\mathrm{C}) \rightarrow S_3(\mathrm{H}) \rightarrow S_2(\mathrm{C})$ ***T*: 20.3** *S*: 0.71	Disassembly Process: $S_2(\mathrm{H}) \rightarrow S_4(\mathrm{C}) \rightarrow S_1(\mathrm{H}) \rightarrow S_3(\mathrm{H})$ *T*: 31.3 *S*: 0.76
Case 3	Disassembly Process: $S_3(\mathrm{H}) \rightarrow S_2(\mathrm{C}) \rightarrow S_1(\mathrm{H})$ ***T*: 18.6** *S*: 0.76	Disassembly Process: $S_2(\mathrm{H}) \rightarrow S_3(\mathrm{H}) \rightarrow S_1(\mathrm{H})$ *T*: 28.4 *S*: 0.67
Case 4	Disassembly Process: $S_2(\mathrm{C}) \rightarrow S_3(\mathrm{H}) \rightarrow S_1(\mathrm{C})$ ***T*: 11.0** *S*: 0.35	Disassembly Process: $S_1(\mathrm{H}) \rightarrow S_2(\mathrm{C})$ The cobot enters the safe range of the human. Thus, task fails.

In Case 1 and Case 3, results generated by the non-game approach are failed to ensure the safety requirement (the cobot enters the safe range of the human).

In Case 2, results generated by the Stackelberg approach are both better than those by the non-game approach in terms of disassembly efficiency and safety index. The main reason is that $U_{h-penalty-}$ in the Stackelberg approach functions, which leads the human to remove the rusted screw S_1.

In Case 3, when the Stackelberg approach is used, T=18.6 and S=0.76; When the non-game approach is used, T=28.4 and S=0.67. It indicates that the Stackelberg approach can achieve a better disassembly efficiency than the non-game approach (the safety requirements of both the approaches are satisfactory). The reason is that in the non-game approach, all the screws are assigned to the human to be removed, while in the Stackelberg approach, $U_{h-penalty-2}$ takes the working state of the cobot into account, so that the disassembly tasks are allocated to the human and cobot more reasonably.

The above case studies demonstrate that the Stackelberg approach can achieve safe and more efficient disassembly for HRC in comparison with the non-game approach. The Stackelberg approach is also more robust in solution generation.

5.2 Case study 2

5.2.1 The EoL product, essential parameters and optimised results

To further prove the applicability of the Stackelberg approach to more complex conditions, Fig. 8 shows an EoL product in which the top cover and the main body are joined using eight M6 screws. Some screws were rusted, and the rest were normal screws. The parameters in the case are given in Table 8.

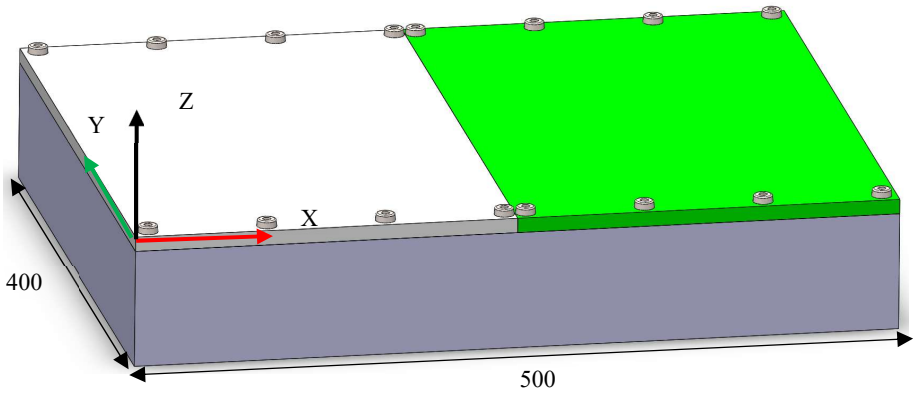


Fig. 8: Disassembly of a product.

Table 8: Screw information of the case study.

Screw no.	**S1**	**S2**	**S3**	**S4**
Coordinate	[1,1,0]	[1,39,0]	[9,1,0]	[9,39,0]
Classification	Normal	Rusted	Normal	Normal
Screw no.	**S5**	**S6**	**S7**	**S8**
Coordinate	[17,1,0]	[17,39,0]	[24.3,1,0]	[24.3,39,0]
Classification	Rusted	Normal	Rusted	Rusted

Screw no.	**S9**	**S10**	**S11**	**S12**
Coordinate	[25.7,1,0]	[25.7,39,0]	[34.7,1,0]	[34.7,39,0]
Classification	Rusted	Normal	Rusted	Normal
Screw no.	**S13**	**S14**	**S15**	**S16**
Coordinate	[42,1,0]	[42,39,0]	[49,1,0]	[49,39,0]
Classification	Normal	Normal	Rusted	Normal

Some near-optimal solutions obtained by the Stackelberg approach are shown in Table 9. When the value of *T* increases, the value of *S* will decrease accordingly. Decision makers can choose the appropriate solutions according to their needs.

Table 9: Some near-optimal solutions obtained by the approach presented in this research.

	w_{h-1}	w_{h-2}	w_{h-3}	w_{h-4}	w_{c-1}	w_{c-2}	w_{c-3}	T	S
n.1	0.209	0.309	0.336	0.146	0.180	0.230	0.590	85.7	0.527
n.2	0.016	0.195	0.787	0.002	0.617	0.031	0.352	88.0	0.495
n.3	0.078	0.421	0.404	0.097	0.549	0.146	0.305	88.4	0.47
n.4	0.260	0.193	0.524	0.023	0.540	0.103	0.357	89.8	0.466
n.5	0.359	0.179	0.430	0.032	0.566	0.054	0.380	90.3	0.451
n.6	0.08	0.416	0.341	0.163	0.555	0.082	0.364	91.7	0.443
n.7	0.394	0.120	0.371	0.115	0.499	0.198	0.303	97.4	0.442

5.2.2 Comparative analysis

5.2.2.1 Comparison with a non-game theory-enabled approach

Similarly, to demonstrate the effectiveness of the Stackelberg approach, a comparison was made with the non-game approach.

Both of the approaches were executed for 200 times. Statistical results are summarised in Table 10. The minimum *T* and *S* values obtained with the game approach are both smaller than those of the non-game method.

Table 10: Statistical results of the optimal results obtained by the two approaches.

Method	*T* values	*S* values
Game method	85.7	0.41
Non-game method	89.6	0.42

5.2.2.2 Comparison of solutions with optimisation and without optimisation

To demonstrate the effectiveness of PSO-Pareto in this research, the results with/without optimised weights were compared. Both approaches were executed 200 times with/without optimised weights 200 times. The statistical results are shown in Table 11.

Table 11: Solution comparison with/without optimisation.

	With optimisation	Without optimisation
The proportion of successes	100%	62.1%
The smallest T value	85.7	91.6
The average T value	95.4	101.5
The smallest S value	0.41	0.46
The average S value	0.47	0.54

During the process, non-optimised weights could cause the human and the cobot to make wrong decisions that lead to incompletion of a disassembly task, i.e., they gave up the disassembly task considering safety. It can be seen from Table 11 that with optimised weights, all the obtained disassembly solutions were able to conduct 100%, and without optimised weights, only 62.1% of the disassembly solutions were acceptable from the safety consideration. Meanwhile, from the smallest T value and S value, the average T value and S value, solutions with optimised weights were statistically better than those without optimised weights.

5.2.2.3 PSO-Pareto under different parameters

The performance of PSO-Pareto under different parameters was compared by using a hypervolume (HV) value [27]. HV measures the volume of the areas surrounded by the Pareto front $\boldsymbol{PF}^*$ and a reference point. When the reference point is fixed, the larger the HV value is, the smaller the T and S corresponding to $\boldsymbol{PF}^*$, indicating better results. To facilitate computation, T and S were normalised.

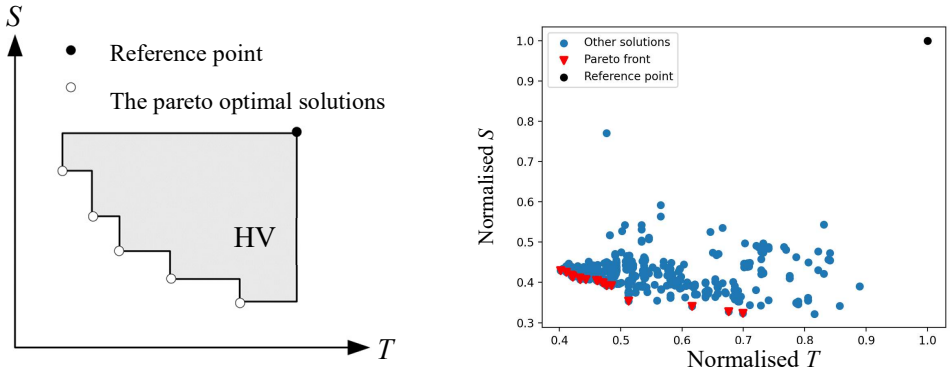


Fig. 9: The HV value for a Pareto front.

For instance, the performance of PSO-Pareto under different numbers of populations was compared in terms of HV. The population numbers were set to (10, 20, 30, 40, 50, 60). For each population, the experiment was executed ten times, and each run was carried out until the HV value converged. The average HV value and running time are shown in Table 12. The ratio of the increase in the HV value to the increase in the running time (ΔHV/Δtime) indicates the efficiency of the run with a specific number of populations. The higher the ΔHV/Δtime is, the higher the efficiency of the run is. Fig. 10 describes ΔHV/Δtime under different numbers of

populations. When the number of populations is 30, the ΔHV/Δtime is the highest, indicating that 30 is the optimal number of populations. Through the same procedure, optimal values of other parameters in the algorithm were obtained (shown in Table 13).

Table 12: The algorithm with different numbers of populations.

Number of populations	Average HV	Average running time (s)
10	0.37423	221.3
20	0.37656	423.5
30	0.38213	612.6
40	0.38352	785.7
50	0.38471	987.6
60	0.38578	1135.3

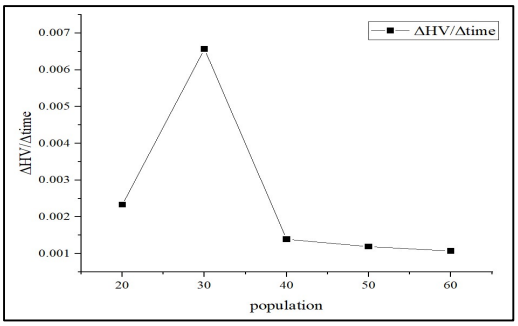


Fig. 10: ΔHV/Δtime under different numbers of populations.

Table 13: Other optimal parameters for the algorithm.

Parameter	Value
c_{max}, c_{min}	0.9,0.5
c_1	2
c_2	2

5.2.2.4 Comparison of different optimisation algorithms

The performance of PSO-Pareto was also compared with the Non-dominated Sorting Genetic Algorithm II (NSGA II), which is another mainstream multi-objective optimisation algorithm [28]. In NSGA II, the crossover rate and the mutation rate were set to 0.9 and 0.1, respectively. For both of the algorithms, the population number was set to 50. The experiments were executed for ten times. The average HV values of the two algorithms are shown in Fig. 11. It clearly demonstrates that the PSO-Pareto algorithm outperformed NSGA II.

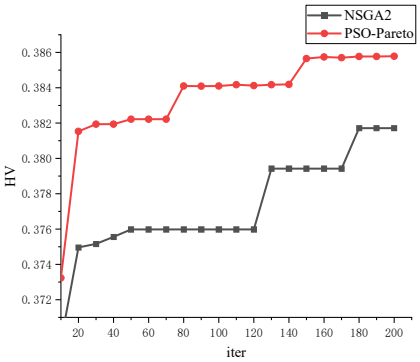


Fig. 11: The HV values of two optimisation algorithms.

5.3 Practical experiment

The Stackelberg approach was implemented in the authors' laboratory. As shown in Fig. 12, the UR5 cobot were used to cooperate with a human operator to carry out joint disassembly of screws on an EoL product. OnRobot's screwdriver was mounted into the end-effector of the cobot for screw disassembly, and a 3D structural light camera was equipped to locate the screw positions intelligently. The blue area represents the safety range of the human operator.

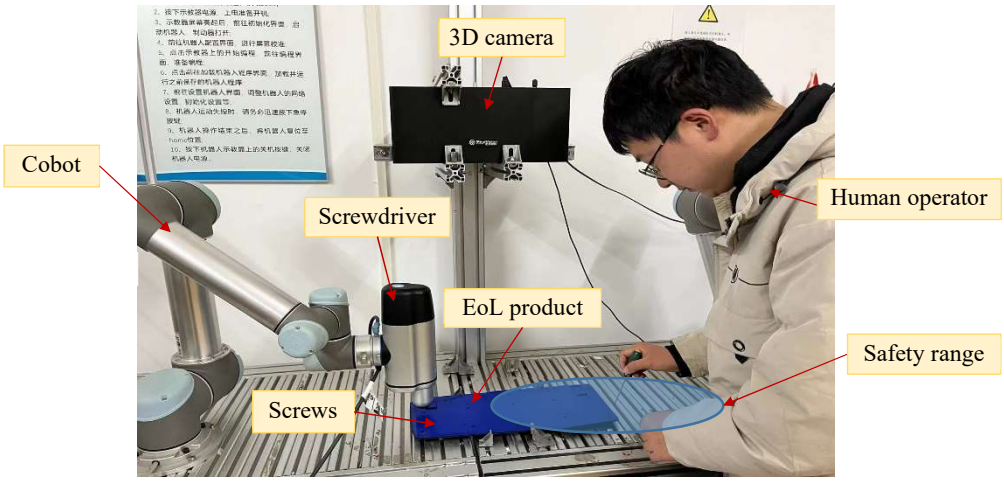


Fig. 12: The developed HRC platform for screw removal.

The technical framework of a more practical experiment is shown in Fig. 13. Fig. 14 shows the distribution of screws on the keyboard. Key parameters for the case are given in Tables 14. The positions of the screws are relative to the base coordinate system of the cobot. Through using PSO-Pareto algorithm, a set of weight values that make the human-cobot collaborative disassembly time shortest is obtained, and it is used to carry out the actual HRC dismantling screw experiment. The Gantt chart of the screw disassembly process is shown in Fig. 15.

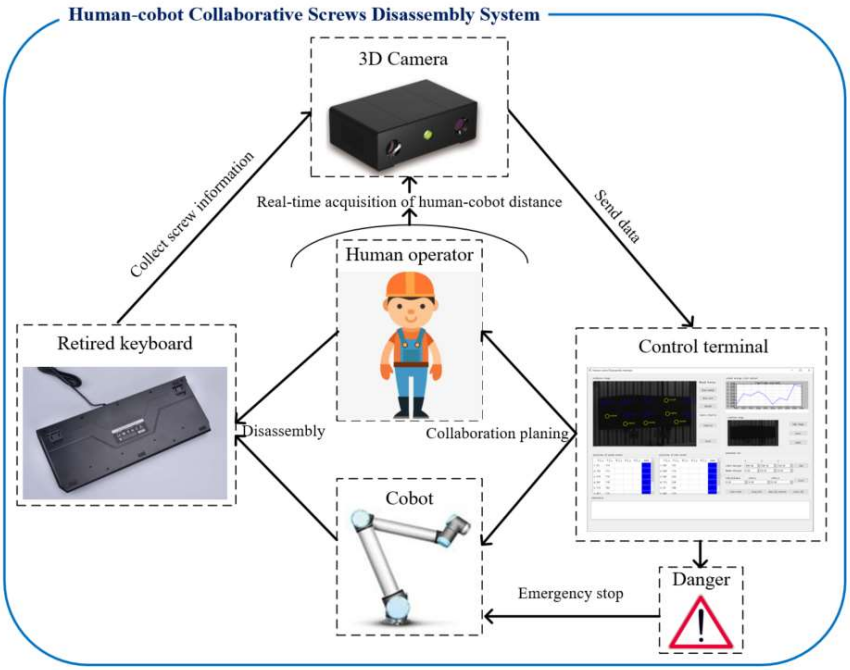


Fig. 13: The technical framework of the practical experiment.

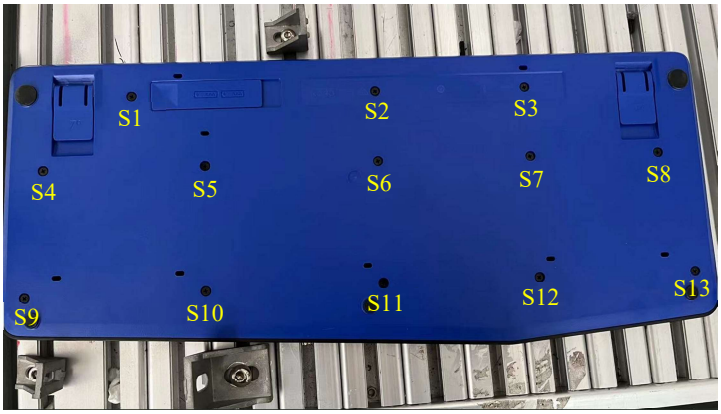


Fig. 14: The distribution of screws on the EoL product.

Table 14: Screw information of the case study.

Screw no.	**S1**	**S2**	**S3**	**S4**
Coordinate	[-633,-24,18]	[-643,-24,18]	[-649,-24,18]	[-629,-20,18]
Classification	Normal	Rusted	Normal	Normal
Screw no.	**S5**	**S6**	**S7**	**S8**
Coordinate	[-636,-20,18]	[-643,-20,18]	[-649,-20,18]	[-655,-20,18]
Classification	Rusted	Normal	Rusted	Normal
Screw no.	**S9**	**S10**	**S11**	**S12**
Coordinate	[-629,-16,18]	[-636,-16,18]	[-643,-16,18]	[-649,-16,18]
Classification	Normal	Normal	Rusted	Normal
Screw no.	**S13**			
Coordinate	[-655,-16,18]			
Classification	Normal			

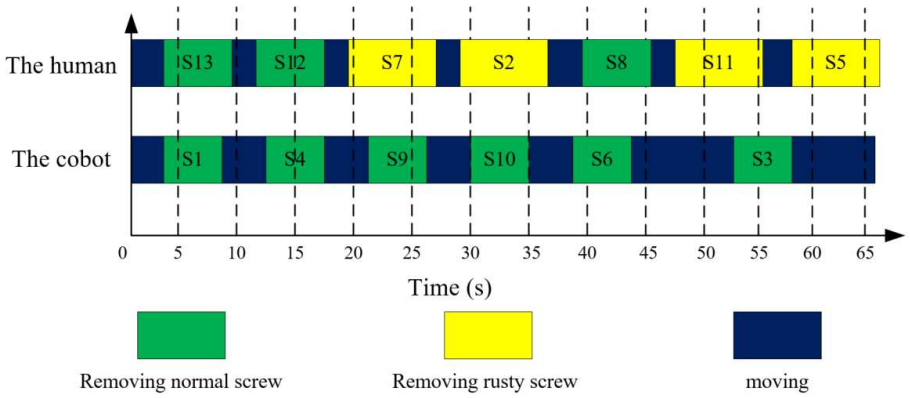


Fig. 15: The Gantt chart of an optimal solution for HRC in disassembly screws.

6. Conclusions and Future Work

In remanufacturing an EoL product, there is often a need to remove the screws that hold it together. This paper presents an innovative HRC approach enabled by the Stackelberg model to optimise the process of removing screws. With this approach, the dynamic and uncertain characteristics of a human operator are well addressed to achieve human-centric HRC disassembly. Meanwhile, utilities in the Stackelberg model are represented by considering the disassembly efficiency and safety between humans and robots, and an innovative PSO-Pareto algorithm is designed to achieve the best performance in terms of safety and efficiency. Finally, case studies were used to validate the effectiveness of the approach, and the superiority of the approach was justified through comparison with a non-game model-based approach and different optimisation algorithms.

In the future, the generality of the designed approach will be further verified in actual and complex disassembly scenarios. Functions of predicting human operators' actions are expected to be added to enhance the safety resilience of the approach. In addition, multi-agent reinforcement learning algorithms could be considered to improve the ability of the approach to address uncertain situations.

Acknowledgement:

This research was sponsored by the National Natural Science Foundation of China (Project No. 51975444). The authors would acknowledge many colleagues who provided constructive comments for improving the research.

References:

[1]. W.H. Chen, G. Foo, S. Kara, M. Pagnucco, Automated generation and execution of disassembly actions, Robotics and Computer-Integrated Manufacturing, 68 (2021), 102056.

[2]. J. Liu, Z. Zhou, D.T. Pham, W. Xu, C. Ji, Q. Liu, Collaborative optimisation of robotic

disassembly sequence planning and robotic disassembly line balancing problem using improved discrete bees algorithm in remanufacturing, Robotics and Computer-Integrated Manufacturing, 61 (2020), 101829.

[3]. G. Pintzos, M. Matsas, N. Papakostas, D. Mourtzis, Disassembly Line Planning Through the Generation of End-of-Life Handling Information from Design Files, in: Procedia CIRP, 57(2016), 740-745.

[4]. K. Wegener, W.H. Chen, F. Dietrich, K. Dröder, S. Kara, Robot assisted disassembly for the recycling of electric vehicle batteries, Elsevier B.V. 29 (2015), 716-721.

[5]. S.P. Anderson, M. Engers, Stackelberg versus Cournot oligopoly equilibrium, International Journal of Industrial Organization, 10 (1992), 127-135.

[6]. R. Li, C. Ji, Q. Liu, Z. Zhou, S. Su, Unfastening of hexagonal headed screws by a collaborative robot, IEEE Transactions on Automation Science and Engineering, 99 (2020), 1-14.

[7]. E. Yildiz, F. Wrgtter, DCNN-based screw detection for automated disassembly processes, in: Proceedings of the 2019 15th International Conference on Signal-Image Technology & Internet-Based Systems, 2019.

[8]. N. M. Difilippo, M. K. Jouaneh, Using the soar cognitive architecture to remove screws from different laptop models, IEEE Transactions on Automation Science and Engineering, 16 (2019), 767-780.

[9]. S. R. Cruz-Ramírez, Y. Mae, Y. Ishizuka, Detection of screws on metal ceiling structures for dismantling systems, in: Proceedings of the International Symposium on Automation and Robotics in Construction, 2008.

[10]. M. Bdiwi, A. Rashid, M. Putz, Autonomous disassembly of electric vehicle motors based on robot cognition, in: Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), 2016.

[11]. Z. Liu, Q. Liu, W. Xu, Z. Zhou, D.T. Pham, Human-robot collaborative manufacturing using cooperative game: framework and implementation, in: Proceedings of the 51st CIRP Conference on Manufacturing Systems, 2018.

[12]. V. Gabler, T. Stahl, G. Huber, O. Oguz, D. Wollherr, A game-theoretic approach for adaptive action selection in close proximity Human-Robot-collaboration, in: Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017.

[13]. Y. Li, K. P. Tee, W. L. Chan, R. Yan, D. K. Limbu, Role adaptation of human and robot in collaborative tasks, in: Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015.

[14]. S. Nikolaidis, S. Nath, A. D. Procaccia, S. Srinivasa, Game-Theoretic Modeling of Human Adaptation in Human-Robot Collaboration, in: Proceedings of the 2017 ACM/IEEE International Conference, 2017.

[15]. C. Messeri, G. Masotti, A. M. Zanchettin, P. Rocco, Human-Robot Collaboration: Optimizing Stress and Productivity Based on Game Theory, IEEE Robotics and Automation Letters, 6 (2021), 8061-8068.

[16]. G. Michalos, S. Makris, J. Spiliotopoulos, I. Misios, P. Tsarouchi, G. Chryssolouris, ROBO-PARTNER: Seamless human-robot cooperation for intelligent, flexible and safe operations in the assembly factories of the future, in: Procedia CIRP. 23 (2014) 71–76.

[17]. S. Hjorth, D. Chrysostomou, Human–robot collaboration in industrial environments: a literature review on non-destructive disassembly, Robotics and Computer-Integrated Manufacturing, 73 (2022), 102208.

[18]. Q. Liu, Z. Liu, W. Xu, Q. Tang, Z. Zhou, D. T. Pham, Human-robot collaboration in disassembly for sustainable manufacturing, International Journal of Production Research, 57 (2019), 4027–4044.

[19]. W. Xu, Q. Tang, J. Liu, Z. Liu, D. T. Pham, Disassembly sequence planning using discrete bees algorithm for human-robot collaboration in remanufacturing, Robotics and Computer-Integrated Manufacturing, 62 (2020), 101860.

[20]. W. Xu, J. Cui, B. Liu, J. Liu, B. Yao, Z. Zhou, Human-robot collaborative disassembly line balancing considering the safe strategy in remanufacturing, J. Clean. Prod. 324 (2021) 129158.

[21]. S. Parsa, M. Saadat, Human-robot collaboration disassembly planning for end-of-life product disassembly process, Robotics and Computer-Integrated Manufacturing, 71 (2021), 102170.

[22]. K. Li, Q. Liu, W. Xu, J. Liu, Z. Zhou, H. Feng, Sequence planning considering human fatigue for human-robot collaboration in disassembly, in: Proceedings of the 11th CIRP Conference on Industrial Product-Service Systems, 2019.

[23]. J. Huang, D.T. Pham, Y. Wang, M. Qu, C. Ji, S. Su, W. Xu, Q. Liu, Z. Zhou, A case study in human–robot collaboration in the disassembly of press- fitted components, Proceedings of the Institution of Mechanical Engineers Part B Journal of Engineering Manufacture, 234 (2019), 654-664.

[24]. J. Huang, D.T. Pham, R. Li, M. Qu, Y. Wang, M. Kerin, S. Su, C. Ji, O. Mahomed, R. Khalil, D. Stockton, W. Xu, Q. Liu, Z. Zhou, An experimental human-robot collaborative disassembly cell, Computers & Industrial Engineering, 155 (2021), 107189.

[25]. R.T. Marler, J.S. Arora, Survey of multi-objective optimization methods for engineering, Structural & Multidisciplinary Optimization, 26 (2004), 369-395.

[26]. G. Venter, J. Sobieszczanski-Sobieski, Particle Swarm Optimization, AIAA Journal, 41 (2003), 129-132.

[27]. M. Emmerich, N. Beume, B. Naujoks, An emo algorithm using the hypervolume measure as selection criterion, Evolutionary Multi-Criterion Optimization, 3410 (2005), 62-76.

[28]. K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation, 6 (2002), 182-197.