A self-evolving system for robotic disassembly sequence planning under uncertain interference conditions

Ye, Fei; Perrett, James; Zhang, Lin; Laili, Yuanjun; Wang, Yongjing

DOI:
10.1016/j.rcim.2022.102392

A self-evolving system for robotic disassembly sequence planning under uncertain interference conditions

Fei Ye [a], James Perrett [b], Lin Zhang [a, *], Yuanjun Laili [a], Yongjing Wang [b]

[a] School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China

[b] School of Engineering, University of Birmingham, Birmingham B15 2TT, UK

* Corresponding author: Lin Zhang, zhanglin@buaa.edu.cn

Abstract: Robotic disassembly sequence planning (DSP) is a research area that looks at the sequence of actions in the disassembly intending to achieve autonomous disassembly with high efficiency and low cost in remanufacturing and recycling applications. A piece of key input information being factored in DSP is the interference condition of a product, i.e., a mathematical representation of the spatial location of components in an assembly usually in the form of a matrix. An observed challenge in the area is that the interference condition can be uncertain due to variations in the end-of-life conditions, and there is a lack of tools available in DSP under uncertain interference. To address this challenge, this paper proposes a new DSP method that can cope with uncertain interference conditions enabled by the *fuzzification of DSP (FDSP)*. This new approach in the core is a fuzzy and dynamic modeling method in combination with an iterative re-planning strategy, and FDSP offers the capability for DSP to adapt to failures and self-evolve online. Three products are given to demonstrate FDSP.

Keywords: Robotic disassembly; Fuzzification; Sequence planning; Dual-loop Self-evolving; Uncertain interference

1. Introduction

Disassembly of end-of-life (EoL) products is a common step in remanufacturing [1], which extracts valuable parts and materials from EOL products through a series of operations, promoting the recycling of resources and reducing the environmental pollution. The disassembly process is often carried out in a certain sequence to improve disassembly efficiency and reduce disassembly costs. This process of obtaining the optimal disassembly sequence is called DSP [2]. A disassembly task usually starts with DSP, which is considered key to the efficiency of remanufacturing and a fundamental technique that underpins autonomous remanufacturing [3-5].

The traditional disassembly process relies on manual operation, as people have strong adaptability and can respond quickly and accurately to uncertain situations. However, with the increase in labor cost and the disassembly process having a specific risk, manual disassembly has become less and less advantageous. In such a situation, robot disassembly has been widely concerned. However, since the EoL product has been used for many years, its condition is unpredictable. This uncertainty forces DSP to change constantly, making automatic robot disassembly face significant challenges.

The automatic disassembly process usually contains three steps: deciding the disassembly mode, building a disassembly model, and applying a selected planning method [6]. Disassembly mode can be classified as complete disassembly and partial disassembly according to the disassembly completion rate or sequential disassembly and parallel disassembly in disassembly order. Disassembly modeling refers to generating component relations of a product through specific methods, including graph-based method, Petri net method, matrix-based method, and other methods. When the previous two steps have been completed, appropriate planning methods can be used to find the optimal disassembly sequence. The optimal disassembly sequence here refers to the sequence meeting specific disassembly objectives (such as disassembly cost, time, benefit, etc.). This process can be completed by many optimization methods, such as Nature Inspired heuristic algorithms (NIHA), rule-based methods (RBM), stochastic simulation (SSI) techniques, and so on [6].

However, most of the current research assumes that the component relationship of the product is constant and the product condition is completely predictable [6-8]. In fact, due to years of use, the state of the EoL product has been inconsistent with the original state, and the static disassembly models and methods are difficult to adapt to the uncertainties [1,9]. Therefore, to make optimization results more suitable for practical applications, it is essential to develop disassembly sequence re-planning methods.

To achieve full autonomy, robots must be able to make intelligent and informed decisions about stochastic, non-deterministic objectives online. An observed challenge in the area is that the interference condition can be uncertain due to variations in the end-of-life conditions, and there is a lack of tools available in DSP under uncertain interference. In line with recent advances in evolution algorithms, various kinds of bees algorithms, genetic algorithms, and memetic algorithms have been applied [10-12]. Agrawal and Nallamothu et al. employed genetic algorithms (GA) to produce optimized DSP from a geometric description of an assembly [13]. Tseng, Huang et al. then adapted conventional GA to form a novel 'Flatworm' algorithm of superior quality [14]. Since population-based evolution algorithms can be time-consuming in dynamic planning and re-planning, heuristics methods with fewer iterations were also proposed [6]. However, the performance of heuristics methods is also influenced by the number of population size and iterations. There is also some previous research about modeling uncertainty in disassembly using fuzzy data sets [15-19]; however, these models are more from the point of view of serviceability and economic viability rather than robotic DSP [20-22].

This paper proposes a method for the fuzzification of DSP (FDSP) as a new DSP system that can cope with uncertain interference conditions by learning from disassembly failures. This new approach in the core is a fuzzy and dynamic modeling method combined with an iterative re-planning strategy, such a combination of which offers the capability for DSP to adapt to failures and self-evolve online. The method involves designing two new fuzzy non-binary matrices combined with a traditional contact matrix to define an assembly in 3D space, which can dynamically represent the uncertainty interference of EoL products. A penalty mechanism is created as a tool to evaluate the cost of changing components and directions. As a product is being disassembled, the feasibility and confidence matrices are continuously updated online to accommodate disassembly failures and support re-planning. The dual-loop self-evolving algorithm allows online decision-making based on the previous operations. As more products are disassembled, the initial values of the feasibility and confidence matrices are renewed to reflect reality more accurately.

The rest of the paper is organized as follows. Section 2 summarizes the state-of-the-art in disassembly sequence planning and re-planning research. Section 3 defines the fuzzy mathematical representation of the physical assemblies. In section 4, the dynamic re-planning process is dissected. In section 5, iterative estimation and sequence ranking schemes are explained. Case studies are examined to assess the algorithm performance in Section 6. Conclusions are drawn in Section 7.

2. Related work

From the perspective of process automation, a robotic DSP mainly contains disassembly modeling and sequence planning [23]. Normally, disassembly modeling is a prerequisite for sequence planning, which eliminates infeasible disassembly sequences by modeling the spatial relationships between product components. The sequence planning methods will efficiently help to obtain the optimal disassembly sequence for given objectives. This section briefly reviews the related literature on disassembly modeling, automatic disassembly sequence planning and re-planning methods.

2.1 Disassembly modeling

Disassembly modeling refers to the representation of EoL products, mainly extracted from the original CAD model and adjusted by manual work to represent the actual product condition. In 1984, Bourjault was one of the first to investigate DSP modeling by using human-machine precedence relationships to create an assembly tree, or liaison graph [24]. This work was further extended by De Fazio and Whitney in 1987, who applied it to generate assembly sequences by defining connective states with binary variables [25]. Usually, the mainstream disassembly modeling methods include the graph-based method, matrix-based method, Petri net method, and other methods, among which the matrix and graph representations are the most popular mathematical representations adopted in DSP.

The disassembly sequence and constraints of a product can be expressed in a graph, which consists of nodes and edges [26]. Usually, a node represents a component, while the edges connecting the nodes represent the relationships between different components. Based on the graph, a disassembly sequence can be determined to describe the precedence relationships among the components in the product. Typical graph representations include the disassembly tree [27], disassembly network graph [28-30], AND/OR graph [31], transformed AND/OR graph [32], state representation-based disassembly graph [33], disassembly constraint graph [34], task precedence diagram [35], and directed flow disassembly network [36]. The disassembly tree starts from the root node to the branch nodes, which indicates the original EoL product and subassemblies or parts of the product, respectively. The disassembly network graph can be three-element, four-element, or five-element. The AND/OR graph can be deduced from the interference matrix. The transformed AND/OR graph uses two types of nodes, artificial nodes and normal nodes, to describe disassembly components and operations. The state representation-based disassembly graph describes the structure of the EoL product at several disassembly levels or disassembly stages. The disassembly constraint graph uses lines to stand for the disassembly operations and nodes representing components. The task precedence diagram is essentially the immediate preceded matrix [37] in the form of graphs. The directed flow disassembly network starts from one node and ends in another node, i.e., from the assembled product and ends at the complete disassembly of all parts.

The graph representations and component precedence relationships are relatively intuitive and easy to understand. However, when the number of components of a complex product is vast, the relationship diagram is prone to a combinatorial explosion. In addition, to facilitate computer processing, graph-based descriptions have to be converted to matrix-based representations. Matrix-based methods employ multiple matrices to describe the disassembly precedence relationships between different parts. The interference matrix is the most commonly used matrix representing the interference of any two components in the product [38]. Dini and Santochi [39] described a procedure for the assembly sequences of a product based on 3 matrices: the interference matrix, the contact matrix, and the connection matrix. To deal with the dynamic changes of the actual situation, the dynamic disassembly precedence matrix was proposed [40]. The immediate preceded matrix focuses on the disassembly sequence wherein the components are disassembled rather than interference [37]. The hybrid disassembly matrix based on both topological disassemblability and tool accessibility includes connectivity matrix, topological disassemblability matrix, and rating matrix of disassemblability level, dedicated to forming an effective and optimized disassembly sequence for selective-disassembly [41]. Luo et al. introduced an efficient method based on matrix representation for selective disassembly planning for product maintenance and recycling to reduce the product operation time and cost [42]. To deal with uncertain interference, Laili et al. created an interference probability matrix to represent the uncertainty in interference conditions [9]. Furthermore, there also exist other matrix-based methods [43], such as disassembly transition matrix, enhanced support matrix [44], and so on.

Another typical expression for disassembly is Petri net method. It can represent the detailed geometrical and topological information on the components of the EoL product. A simplified representation of the disassembly Petri net includes places, transitions, and arcs, respectively, which represent the sets of EoL products, disassembly operators, and disassembly routes [45]. To enrich the description of the disassembly structure, more elements can be added to the simplified disassembly Petri net to form new ones, such as five-tuple disassembly Petri net [46] and eight-tuple disassembly Petri net [47]. Tang et al. [48] proposed a workstation Petri net and product Petri net method for its hierarchical and modular modeling to derive the disassembly path with the maximal end-of-life value. Grochowski and Tang [49] proposed a disassembly Petri net combined with a hybrid Bayesian network to automatically decide the optimal disassembly action without the input of a human operator. Kuo [50] proposed a Petri net-based analysis approach to deal with the disassembly and recycling problems in EOL Electrical and Electronic Equipment, finding the optimal tradeoff between the cost and environmental effectiveness of the disassembly processes. Hsu [51] proposed a fuzzy knowledge-based disassembly process planning system for disassembly process planning based on a novel high-level Petri net, termed fuzzy attributed and timed predicate/transition net.

In addition to the above mainstream methods, other methods have also been studied [19,52-54]. In general, these methods have both advantages and disadvantages. Graphics provide an intuitive way to describe the priority relationships between parts of an EoL product. Petri nets give detailed geometric and topological information about EoL product components. However, as the number of product components increases, both methods are prone to combinatorial explosions. The matrix-based method is more conducive to computer processing, but because the relationship among all components needs to be recorded, it is easy to lead to data redundancy. In addition, since the condition of EoL products changes with time, how to express this uncertainty is a more challenging problem.

2.2 Methods of disassembly sequence planing and re-planning

The goal of the DSP is to find the optimal disassembly sequence under given objectives. The disassembly objectives consist of several categories, namely disassembly time, disassembly cost, economic value, and environmental index [55]. The disassembly time can be divided into the time for component removing, direction changing, tool changing, and tool traveling among different points. The disassembly cost includes tools, energy, labor, etc. The economic value refers to the profitability of the disassembly operation, and environmental index represents the impact on the environment.

In order to find the optimal sequence efficiently, many optimization algorithms are employed. In literature [6], the main methods are categorized as the nature inspired heuristic algorithms (NIHA), linear programming methods (LPM), rule-based methods (RBM), stochastic simulation techniques (SSI) and so on. Ren et al. [10] proposed a genetic algorithm based approach to solve the asynchronous parallel DSP problem. Han et al. [52] proposed an improved LPM to solve the selective disassembly sequencing under the sequential disassembly environment for the objective of minimizing the total disassembly cost. Tian et al. [56] Employed Monte Carlo simulation to obtain the optimal disassembly sequence with random removal time and different removal labor cost. In addition to using one optimization algorithm independently, recent research prefers to design hybrid methods to handle complex situations. Smith and Chen [57] combined RBM and genetic algorithm to narrow the search space and find the optimal disassembly sequence, thus reducing the search time.

However, the actual condition of an EoL product is often different from the initial condition, to obtain automatic disassembly, the robot must be able to make quick decisions against these uncertainties. Therefore, dynamic disassembly sequence re-planning methods is essential when one disassembly operation of the preliminary plan has failed. In the literature [58], the authors described the uncertainty-related difficulties in DSP, presenting a framework for dealing with uncertainty in DSP implementation and demonstrating the method by a simple example. Zussman et al. [59] proposed the use of a Petri net based pre-ordered transition list to adjust the disassembly in case of a failure. However, the ranking sorted used for online decision making may not be available. Further, the next transition may also fail if the sorted values and component relationships are not updated. ElSayed et al. [60] proposed an online decision making process using GA to generate an automated robot disassembly plan. However, the method lacks a re-planning strategy to quickly respond to disassembly failures. Laili et al. [1] proposed a robotic disassembly re-planning method combining a two-pointer detection strategy and a super-fast bees algorithm to identify detachable elements. This method can enable a robot to adapt for re-planning very fast in the presence of task failures. However, as the population size and number of iterations increase, the efficiency of the heuristic algorithm also decreases. The existing literature indicates that there are limited studies on robotic disassembly sequence re-planning and this work is a future research trend considering the uncertainty of the actual DSP problems [6].

3. Fuzzy mathematical representation of physical assemblies

In our previous research [61], we demonstrated a mathematical representation of an assembly through three matrices: a contact matrix, a space interference matrix, and a relation matrix. The space interference matrix indicates the existence or absence interference between components along different directions. The contact matrix is similar to space interference matrix but focuses on contact relationship. The relation matrix reveals the general contact status of components considering all directions.

However, binary matrices cannot describe products under uncertain interference conditions. Although Lai Li et al. [9] proposed an interference probability matrix as a new mathematical representation that uses probability to indicate uncertainty in the interference, the multi-threshold planning scheme uses the same threshold for all components, ignoring the variability among components. To solve this problem, we propose the use of two fuzzy dynamic matrices in combination with a contact matrix to define an assembly in 3D space. To illustrate the proposed method more clearly, a simple product in 2D space, shown in Figure 1, will be used as an example.

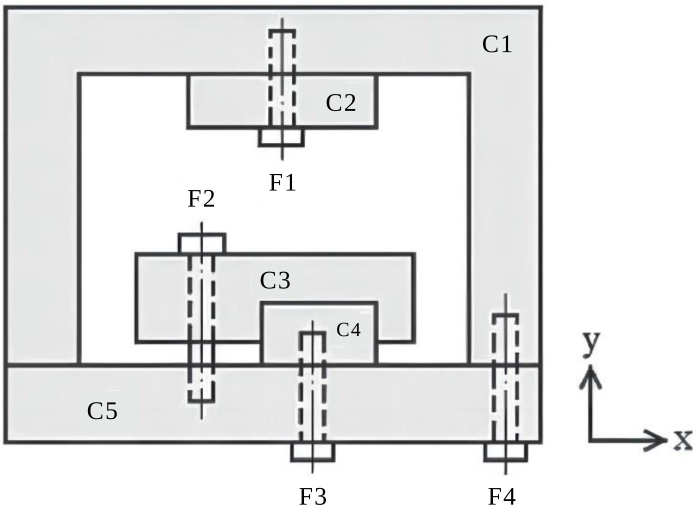

Figure 1. An example Product A (Wang et al. 2021 [21]).

The contact matrix defines the contact relationship among components in 4 or 6 directions. As the product in Fig. 1 has 9 components and 4 removal directions (X+, X-, Y+, Y-), a 9x36 matrix is populated with 1's for contact (in the direction of motion, relative to another part) and 0's for non-contact. The relationship matrix is a compression of the contact matrix that focuses on interference between components rather than directional interference, which functions like an OR gate, outputting a value of 1 if the component shares contact with another part in any given direction. The relationship between these two matrices can be seen from Equations 1 and 2.

$$C=\begin{bmatrix} 0 & 0 & 0 & 0 & C^{C2}_{C1x+} & C^{C2}_{C1x-} & C^{C2}_{C1y+} & C^{C2}_{C1y+} & \cdots & C^{F4}_{C1x+} & C^{F4}_{C1x-} & C^{F4}_{C1y+} & C^{F4}_{C1y-} \\ C^{C1}_{C2x+} & C^{C1}_{C2x-} & C^{C1}_{C2y+} & C^{C1}_{C2y-} & 0 & 0 & 0 & 0 & \cdots & C^{F4}_{C2x+} & C^{F4}_{C2x-} & C^{F4}_{C2y+} & C^{F4}_{C2y-} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ C^{C1}_{F4x+} & C^{C1}_{F4x-} & C^{C1}_{F4y+} & C^{C1}_{F4y-} & C^{C2}_{F4x+} & C^{C2}_{F4x-} & C^{C2}_{F4y+} & C^{C2}_{F4y+} & \cdots & 0 & 0 & 0 & 0 \end{bmatrix} \quad (1)$$

$$R = \begin{bmatrix} 0 & R_{C1}^{C2} & \cdots & R_{C1}^{F4} \\ R_{C2}^{C1} & 0 & \cdots & R_{C2}^{F4} \\ \vdots & \vdots & \ddots & \vdots \\ R_{F4}^{C1} & R_{F4}^{C2} & \vdots & 0 \end{bmatrix} \qquad (2)$$

where *C* is the contact matrix (shown in Table 2A), C_{C1x+}^{C2} is a binary number indicating whether component C2 impedes component C1 along X+ direction, *R* is the relation matrix (shown in Table 2D), and R_{C1}^{C2} indicates whether component C2 impedes component C1in any directions.

In this study, two new non-binary matrices are developed, one a mirror of the other, initially based on the fuzzification of Wang et al. s' space interference matrix [61, 62]. The feasibility matrix provides a value for each component direction of movement relative to each other. Scoring from 0 (completely free to move) to 1 (direct contact/impossible). The confidence matrix evaluates "How guaranteed is this move?" from 0 (absolutely confident) to 1 (very uncertain).

Table 1 indicates the a scoring system that has been used in this research. These initial conditions can influence the performance of DSP, and have been assumed as follows: Perpendicular interference, i.e., Pulling a component directly away from another, has the lowest feasibility and confidence scores, 0.05 and 0.01 respectively. Theoretically, this move is most likely to succeed for two contacting components, even though rust and temperature-induced adhesion remain a possibility. Parallel removal is considered slightly less feasible (0.1) and more uncertain (0.05) due to increased surface friction and prolonged contact time during removal. Similarly, threads are attributed 0.2 feasibility and 0.1 confidence, respectively. An impossible move scores 1 feasibility and 0 confidence as we can be sure that two physical objects have obvious spatial interference. Finally, remote interference, which refers to a future contact in a given direction, is considered somewhat feasible (0.3) but very uncertain (0.1). It is important to remember that although counter-intuitive, scoring low in both feasibility and confidence indicates a high probability of removal.

It is worth mentioning that 1) the scores provided are indicative, and can be changed to the user's preferences; and 2) they are initial parameters and will be adjusted automatically as DSP cases increase.

Table 1. Feasibility & Confidence scoring metric

Removal Relationship	Contact	Feasibility	Confidence
Free to move	0	0	0
Thread	0	0.2	0.1
Perpendicular	0	0.05	0.01
Parrallel	0	0.1	0.05
Remote	0	0.3	0.1
Direct/Impossible	1	1	0

Using Example 1 (Fig.1), the feasibility score for C1 in the X+ direction can be calculated as the sum of C1's feasibility when moved in the X+ direction relative to other components. Equation (2) indicates this compression:

$$F_{source}=\begin{bmatrix} 0 & 0 & 0 & 0 & F^{C2}_{C1x+} & F^{C2}_{C1x-} & F^{C2}_{C1y+} & F^{C2}_{C1y+} & \cdots & F^{F4}_{C1x+} & F^{F4}_{C1x-} & F^{F4}_{C1y+} & F^{F4}_{C1y-} \\ F^{C1}_{C2x+} & F^{C1}_{C2x-} & F^{C1}_{C2y+} & F^{C1}_{C2y-} & 0 & 0 & 0 & 0 & \cdots & F^{F4}_{C2x+} & F^{F4}_{C2x-} & F^{F4}_{C2y+} & F^{F4}_{C2y-} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ F^{C1}_{F4x+} & F^{C1}_{F4x-} & F^{C1}_{F4y+} & F^{C1}_{F4y-} & F^{C2}_{F4x+} & F^{C2}_{F4x-} & F^{C2}_{F4y+} & F^{C2}_{F4y+} & \cdots & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (3)$$

$$F_{C1x+}=\sum 0+F^{C2}_{C1x+}+F^{C3}_{C1x+}+F^{C4}_{C1x+}+F^{C5}_{C1x+}+F^{F1}_{C1x+}+F^{F2}_{C1x+}+F^{F3}_{C1x+}+F^{F4}_{C1x+} \qquad (4)$$

where F_{source} is the uncompressed feasibility matrix (shown in Table 2B), F^{C2}_{C1x+} represents the feasibility interference between C1 and C2 when moving C1 along the X+ direction, and F_{C1x+} is the feasibility score moving of C1 along the X+ direction at a given moment.

The same summation method is used to compress the confidence matrix. The full uncompressed and more digestible compressed matrices can be seen below in Table 2(A-F).

Table 2(A-F). Uncompressed C, F & U, and corresponding compressed R, F & U matrices

A) Contact Matrix

	C1				C2				C3				C4				C5				F1				F2				F3				F4			
	X+	X-	Y+	Y-	X+	X-	Y+	Y-	X+	X-	Y+	Y-	X+	X-	Y+	Y-	X+	X-	Y+	Y-	X+	X-	Y+	Y-	X+	X-	Y+	Y-	X+	X-	Y+	Y-	X+	X-	Y+	Y-
C1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
C2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
C3	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
C4	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
C5	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
F1	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F2	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F3	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F4	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

B) Feasibility Matrix

	C1				C2				C3				C4				C5				F1				F2				F3				F4			
	X+	X-	Y+	Y-	X+	X-	Y+	Y-	X+	X-	Y+	Y-	X+	X-	Y+	Y-	X+	X-	Y+	Y-	X+	X-	Y+	Y-	X+	X-	Y+	Y-	X+	X-	Y+	Y-	X+	X-	Y+	Y-
C1	0	0	0	0	0.3	0.3	0.05	1	0.3	0.3	0	0.3	0.3	0.3	0	0.3	0.1	0.1	0.05	1	1	1	1	1	0.3	0.3	0	0.3	0.3	0.3	0	0.3	1	1	1	1
C2	0.3	0.3	1	0.05	0	0	0	0	0	0	0	0.3	0	0	0	0.3	0	0	0	0.3	1	1	1	1	0	0	0	0.3	0	0	0	0.3	0	0	0	0
C3	0.3	0.3	0.3	0	0	0	0.3	0	0	0	0	0	1	1	0.1	1	0	0	0	0.3	0	0	0.3	0	1	1	1	1	0	0	0	0.3	0.3	0	0	0
C4	0.3	0.3	0.3	0	0	0	0.3	0	1	1	1	0.2	0	0	0	0	0.1	0.1	0.05	1	0	0	0.3	0	0	0.3	0	0	1	1	1	1	0.3	0	0	0
C5	0.1	0.1	1	0.05	0	0	0.3	0	0	0	0.3	0	0.1	0.1	1	0.05	0	0	0	0	0	0	0.3	0	1	1	1	1	1	1	1	1	1	1	1	1
F1	1	1	1	0.2	1	1	1	0.2	0	0	0	0.3	0	0	0	0.3	0	0	0	0.3	0	0	0	0	0	0	0	0	0	0	0	0.3	0	0	0	0
F2	0.3	0.3	0.3	0	0	0	0.3	0	1	1	0.2	1	0.3	0	0	0	1	1	0.2	1	0	0	0	0	0	0	0	0	0.3	0	0	0	0.3	0	0	0
F3	0.3	0.3	0.3	0	0	0	0.3	0	0	0	0.3	0	1	1	1	0.2	1	1	1	0.2	0	0	0.3	0	0	0.3	0	0	0	0	0	0	0.3	0	0	0
F4	1	1	1	0.2	0	0	0	0	0	0.3	0	0	0	0.3	0	0	1	1	1	0.2	0	0	0	0	0	0.3	0	0	0	0.3	0	0	0	0	0	0

C) Confidence Matrix

	C1				C2				C3				C4				C5				F1				F2				F3				F4			
	X+	X-	Y+	Y-	X+	X-	Y+	Y-	X+	X-	Y+	Y-	X+	X-	Y+	Y-	X+	X-	Y+	Y-	X+	X-	Y+	Y-	X+	X-	Y+	Y-	X+	X-	Y+	Y-	X+	X-	Y+	Y-
C1	0	0	0	0	0.1	0.1	0.01	0	0.1	0.1	0	0.1	0.1	0.1	0	0.1	0.05	0.05	0.01	0	0	0	0	0	0.1	0.1	0	0.1	0.1	0.1	0	0.1	0	0	0	0
C2	0.1	0.1	0	0.01	0	0	0	0	0	0	0	0.1	0	0	0	0.1	0	0	0	0.1	0	0	0	0	0	0	0	0.1	0	0	0	0.1	0	0	0	0
C3	0.1	0.1	0.1	0	0	0	0.1	0	0	0	0	0	0	0	0.05	0	0	0	0	0.1	0	0	0.1	0	0	0	0	0	0	0	0	0.1	0.1	0	0	0
C4	0.1	0.1	0.1	0	0	0	0.1	0	0	0	0	0.1	0	0	0	0	0.05	0.05	0.01	0	0	0	0.1	0	0	0.1	0	0	0	0	0	0	0.1	0	0	0
C5	0.05	0.05	0	0.01	0	0	0.1	0	0	0	0.1	0	0.05	0.05	0	0.01	0	0	0	0	0	0	0.1	0	0	0	0	0	0	0	0	0	0	0	0	0
F1	0	0	0	0.1	0	0	0	0.1	0	0	0	0.1	0	0	0	0.1	0	0	0	0.1	0	0	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0
F2	0.1	0.1	0.1	0	0	0	0.1	0	0	0	0.1	0	0.1	0	0	0	0	0	0.1	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0.1	0	0	0
F3	0.1	0.1	0.1	0	0	0	0.1	0	0	0	0.1	0	0	0	0	0.1	0	0	0	0.1	0	0	0.1	0	0	0.1	0	0	0	0	0	0	0.1	0	0	0
F4	0	0	0	0.1	0	0	0	0	0	0.1	0	0	0	0.1	0	0	0	0	0	0.1	0	0	0	0	0	0.1	0	0	0	0.1	0	0	0	0	0	0

D) Relation Matrix

	C1	C2	C3	C4	C5	F1	F2	F3	F4
C1	0	1	0	0	1	1	0	0	1
C2	1	0	0	0	0	1	0	0	0
C3	0	0	0	1	0	0	1	0	0
C4	0	0	1	0	1	0	0	1	0
C5	1	0	0	1	0	0	1	1	1
F1	1	1	0	0	0	0	0	0	0
F2	0	0	1	0	1	0	0	0	0
F3	0	0	0	1	1	0	0	0	0
F4	1	0	0	0	1	0	0	0	0

E) Compressed Feasibility Matrix

	X+	X-	Y+	Y-
C1	3.6	3.6	2.1	5.2
C2	1.3	1.3	2	2.55
C3	2.6	2.3	2	2.6
C4	2.7	2.7	2.95	2.2
C5	3.2	3.2	5.9	3.1
F1	2	2	2	1.6
F2	3.2	2.3	1	2
F3	2.6	2.6	3.2	0.4
F4	2	3.2	2	0.4

F) Compressed Confidence Matrix

	X+	X-	Y+	Y-
C1	0.55	0.55	0.02	0.4
C2	0.1	0.1	0	0.51
C3	0.2	0.1	0.35	0.2
C4	0.25	0.25	0.31	0.1
C5	0.1	0.1	0.3	0.02
F1	0	0	0	0.6
F2	0.4	0.1	0.4	0
F3	0.2	0.2	0.4	0.2
F4	0	0.4	0	0.2

It is worth noting that the relation matrix functions like an OR gate that focuses on interference between components rather than directional interference, while the feasibility and confidence matrices summarize the directional relationships of the components. The contact matrix can be obtained from a CAD model. The initial scores of feasibility and confidence are subjectively assigned and are continuously updated and infinitely approximated to the actual condition of the product as DSP cases increase.

4. Dual-loop self-evolving framework

The intention of the FDSP is to provide a self-evolving capability for a DSP system to learn from historical passes and failures. Breaking this task down into two parts: 1) as *a product* is being disassembled, the feasibility and confidence matrices are continuously updated online; and 2) as *more products* are disassembled, the initial values of the feasibility and confidence matrices are renewed to reflect reality more accurately.

To address the two components above, we designed a dual-loop self-evolving framework, which is built on the foundation of our work on disassembly re-planning [1]. This framework as given in Figure 3 adopts several techniques, including layered candidate sets, Monte Carlo forward propagation, penalty value matrices, etc.

4.1 Layered candidate set

In each iteration, a removable component is detected and removed. This iterative process can be represented as multiple layers, that is, the removable components detected in an iteration form a layer. All the layers formulate a layered candidate set, which can be represented by Equation (5) .

$$\begin{aligned} S &= \left\{ S_1 \middle| S_2 \middle| \cdots \middle| S_H \right\} \\ &= \left\{ E_1 \circ d_1, \ldots, E_{b1} \circ d_{b1} \middle| E_{b_1+1} \circ d_{b_{1+1}}, \ldots, E_{b_2} \circ d_{b_2} \middle| \ldots E_{b_{H-1}+1} \circ d_{b_{H-1+1}}, \ldots, E_n \circ d_n \right\} \end{aligned} \qquad (5)$$

where S_i represents the ith layer; E_i, $i \in [1, n]$ represents a component; d_i stands for removable direction. A boundary list $b = \{b_1, b_2, \ldots, b_{H-1}\}$ denotes the endpoint of each layer. A layer S_i, $i \in [1, H-1]$ contains $b_{i+1} - bi + 1$ components. As explained in [9], the components in a layer do not interfere with each other, so their disassembly orders are interchangeable, while different layers cannot be swappable.

Based on the layered candidate set, the algorithm will start the disassembly from the first layer in the layered order, prioritizing the search for disassembled components within the same layer and assigning different priorities to different layers according to the component space constraints. Such a layered structure can prevent incorrect disassembly sequences and promote the convergence of the feasibility and confidence matrices.

4.2 Monte Carlo propagation of uncertainty

As defined in Section 3, two new non-binary fuzzy matrices are designed to represent the uncertainty of the interference. The feasibility matrix assesses the feasibility of a disassembly operation, while the confidence matrix assesses the credibility of the feasibility matrix. Only these two matrices work together to score a disassembly operation. Scoring low in both feasibility and confidence values indicates a high probability of removal, so they should be summed to form a composite evaluation metric. Considering that the feasibility and confidence values follow different distribution functions, the Monte Carlo simulation is introduced to deal with the correlation between the distributions of various uncertain and random variables and the correlation between the input factors for the stochastic DSP model [63]. The Monte Carlo simulation assigns random values to each uncertain variable and then generates the output results.

Therefore, we replace the direct summation of feasibility and confidence with the Equation (6), where the last item, i.e., two times the standard deviation is to simulate a 95% confidence interval of the normal distribution. The decision on which component to remove depends on the Summation matrix. This matrix is the sum of the feasibility matrix, the mean of a significant times the normal distribution of the confidence matrix, and two times the standard deviation. This calculation applies a tolerance to the initial feasibility scores.

$$S=\begin{bmatrix} F_{C1X+} & F_{C1X-} & F_{C1Y+} & F_{C1Y-} \\ F_{C2X+} & F_{C2X-} & F_{C2Y+} & F_{C2Y-} \\ \vdots & \vdots & \vdots & \vdots \\ F_{f4X+} & F_{f4X-} & F_{f4Y+} & F_{f4Y-} \end{bmatrix}+\begin{bmatrix} \mu_{C1X+} & \mu_{C1X-} & \mu_{C1Y+} & \mu_{C1Y-} \\ \mu_{C2X+} & \mu_{C2X-} & \mu_{C2Y+} & \mu_{C2Y-} \\ \vdots & \vdots & \vdots & \vdots \\ \mu_{f4X+} & \mu_{f4X-} & \mu_{f4Y+} & \mu_{f4Y-} \end{bmatrix}+2\begin{bmatrix} \sigma_{C1X+} & \sigma_{C1X-} & \sigma_{C1Y+} & \sigma_{C1Y-} \\ \sigma_{C2X+} & \sigma_{C2X-} & \sigma_{C2Y+} & \sigma_{C2Y-} \\ \vdots & \vdots & \vdots & \vdots \\ \sigma_{f4X+} & \sigma_{f4X-} & \sigma_{f4Y+} & \sigma_{f4Y-} \end{bmatrix} \quad （6）$$

where S is the summation matrix and F_{C1X+}, μ_{C1X+} & δ_{C1X+} represent the directional feasibility, the mean of the normal distribution of directional confidence, and the standard deviation, respectively. This summation is conducted before each removal decision, and as the confidence matrix is updated every iteration, this uncertainty propagation can prove decisive.

4.3 Penalty matrix

It is reasonable to assume that upon failure, the uncertainty of a component will increase and become less feasible, as well as for its contacting components. In this case, variable penalties are applied to all contacting components with uncertainty (or confidence values). The coordinates of these components are taken from the Relation matrix.

The penalties are applied in the form of a penalty matrix determined as the product of the uncompressed confidence value of the contacting parts and a penalty cost. The confidence values can be considered as weights in a neural network - the dynamic properties of which will be described in section 4.5. This penalty matrix is then applied to the uncompressed feasibility matrix before the next iteration of compression, summation, and removal decision.

Taking product A as an example. When F1 and F4 are removed, assuming that C1 is chosen for removal in the Y+ direction but fails, then C2 and C5 will also be punished as C1, C2 and C5 are the only contacting components. It is important to note that relevant confidence matrices will undergo significant change throughout the program, thus determining the weight of these penalties.

4.4 Training data

Since the conditions of EOL products are usually different, a stochastic DSP model should be established to introduce uncertainties in various variables, and the resulting DSP problem is closer to the actual situation. For the purpose of this experiment, the training data is managed as a set of probability ranges or Chance of Failure (CoF). This will allow us to verify if the output is coherent with the input:

①The question is whether the steady-state feasibility and confidence matrices are rational after a significant number of cycles.

②Preparing the data in this manner allows easy adaptation for testing, rather than creating a finite list of instructions for each experiment.

③Using stochastic data will generate a degree of noise that is to be expected in situ.

These CoF ranges will be illustrated in the upcoming examples.

4.5 Dynamic uncertainty

The crucial element in adapting to uncertainties is how confidence is manipulated at each decision. The following pass/fail configuration will determine the stability and plasticity of our system. In this study, we plan to iterate thousands of times over, and that being the case, a relatively stable manipulation mechanism has been designed. Figure 2 presents the percentage change in confidence for both passes and failures.

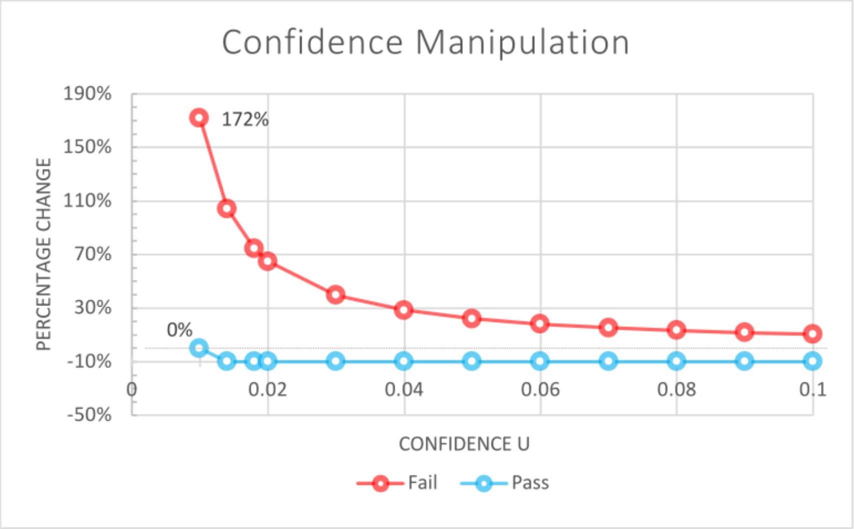

Figure 2. The effect of Passes and Failures on confidence values (U)

When a disassembly operation fails, the confidence values will increase upon failures due to the penalty. Therefore, this factor should be a number that is always greater than 1. Components with low confidence will be severely penalized if they fail, i.e., if a component we believe to be deterministically removable fails, it is severely punished. This means that if a part fails consecutively, significant penalties will be accumulated based on the increased uncertainty. This exponential curve provides the system with some agility. Conversely, if a component that is believed to be uncertain fails, the confidence values will increase almost linearly. The component was expected to fail, and it did, so the initial prediction was relatively appropriate. In view of the above reasons, a factor of $e^{(0.01/U)}$ is used to increase the confidence value upon failure. Here, U_{+1} denotes the next iteration's confidence value.

$$U_{+1} = U * e^{0.01/U} \qquad (7)$$

As we expect to witness passes several times more often than failures, a passing component receives a statutory 10% reduction in confidence value. Confidence values will rarely fall below 0.01, except for absolutely certain parts, which remain at confidence 0 throughout the program. With this stable system in place, we can expect most confidence values to remain in the range of 0.01-0.1.

4.6 Dual-loop self-evolving framework

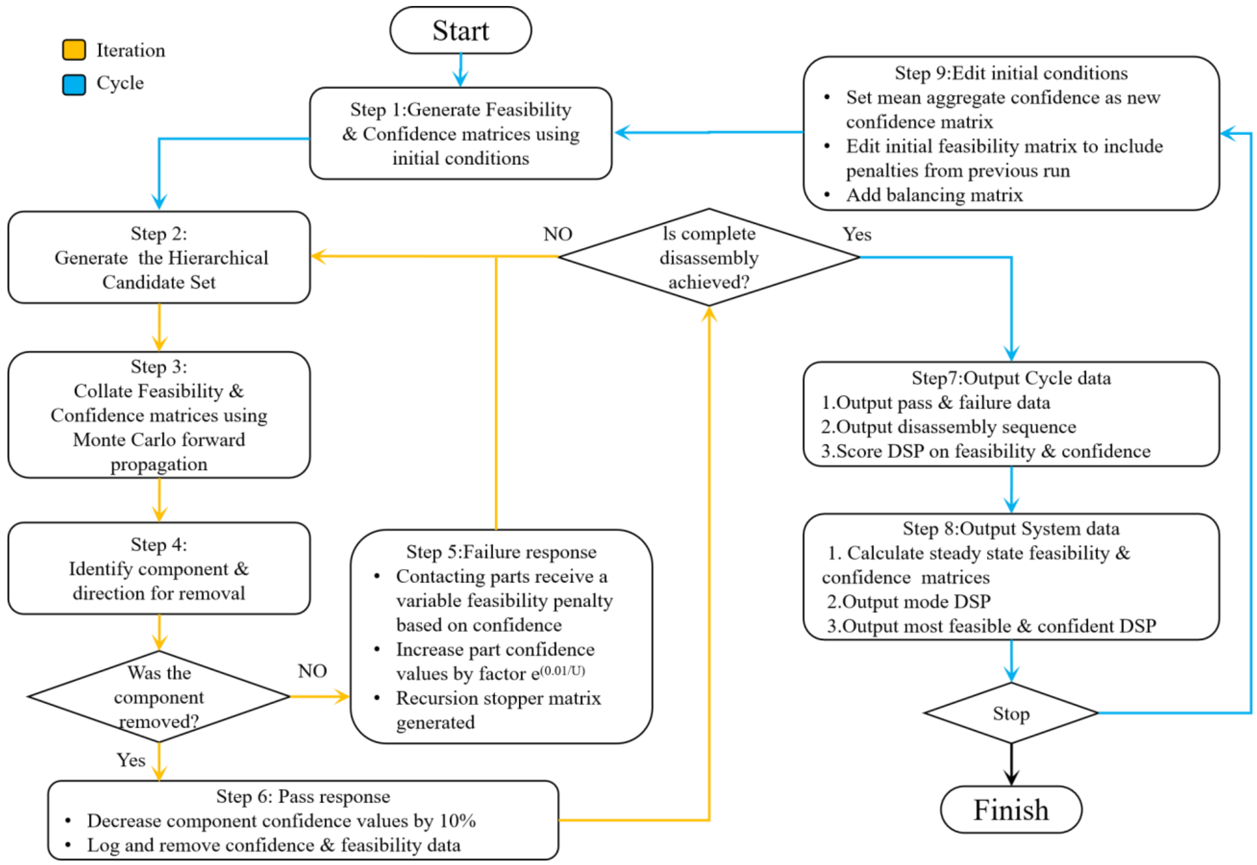
Iteration
Cycle
Start
Step 1:Generate Feasibility & Confidence matrices using initial conditions
Step 9:Edit initial conditions
• Set mean aggregate confidence as new confidence matrix
• Edit initial feasibility matrix to include penalties from previous run
• Add balancing matrix
Step 2: Generate the Hierarchical Candidate Set
NO
Is complete disassembly achieved?
Yes
Step 3: Collate Feasibility & Confidence matrices using Monte Carlo forward propagation
Step7:Output Cycle data
1.Output pass & failure data
2.Output disassembly sequence
3.Score DSP on feasibility & confidence
Step 4: Identify component & direction for removal
Step 5:Failure response
• Contacting parts receive a variable feasibility penalty based on confidence
• Increase part confidence values by factor e(0.01/U)
• Recursion stopper matrix generated
Step 8:Output System data
1. Calculate steady state feasibility & confidence matrices
2.Output mode DSP
3.Output most feasible & confident DSP
Was the component removed?
NO
Yes
Stop
Finish
Step 6: Pass response
• Decrease component confidence values by 10%
• Log and remove confidence & feasibility data

Figure 3. The algorithm flow chart

As shown in Figure 3, the dual-loop refers to the inner loop and the outer loop. In the inner loop (Step 2 to Step 6), the values of feasibility and confidence will be updated online according to historical passes and failures. When a complete disassembly is achieved, the relevant data generated by this disassembly will be obtained and then used as the initial conditions for the next disassembly. In the outer loop (Step 7 to Step 9), the initial conditions are continuously updated through multiple successful disassemblies, and the finally obtained feasibility and confidence matrices are closer to the actual situation of the product. The self-evolving is reflected in updating the feasibility and confidence matrices. The following are the specific implementation steps of this method.

Steps 1-4: The compressed Feasibility, Confidence, and Relation matrices are generated. A layered candidate set will be generated before determining the disassembly components and directions. Based on the generated layered candidate set, a component is identified and an attempt is made to remove it.

Step 5 - Failure Response: Upon failure, parts contacting the identified component receive a variable feasibility penalty based on the uncompressed confidence matrix and a fixed value, penalty cost. The confidence values of the components are then increased by a factor $e^{(0.01/U)}$ – increasing the uncertainty. Notably, if the penalty is less than the difference that prevents the component from re-suggesting, a recursion stopper matrix is generated and stored to provide a temporary penalty for the next iteration.

Step 6 - Pass Response: Upon successful removal of a given component, the corresponding confidence value is decreased by 10%. The component feasibility and confidence data are stripped from the source data and stored in the corresponding Aggregate matrices. Likewise, the component feasibility, confidence and their product at that moment are kept as scores for later evaluation of sequences.

Step 7 – Cycle Data: The cycle completes and the algorithm outputs pass/failure data as well as the achieved DSPs. Total scores are calculated for the completed sequences.

Step 8 – System Data: The system outputs mean aggregate feasibility and confidence. The steady-state matrices display the effect of all past cycles. A dictionary of sequences is created to find the mode DSP. The scores are then used to locate the most feasible, most confident, and best overall DSP.

Step 9 – Update Initial Conditions: The Algorithm runs until the number of cycles equals the run_size criterion. Otherwise, the initial confidence is set equal to mean aggregate confidence. The previous cycle penalties are applied to the stored feasibility matrix and a balancer is subtracted to allow for iterative estimation, preventing the new feasibility matrix from converging. The new feasibility matrix is then stored and prepared for the next cycle.

To reflect the self-evolving process of the algorithm more intuitively, Product A is selected to demonstrate the evolution of the confidence values with different failure probabilities. Figure 4 shows the variation of the confidence values of component C1 along the X- direction with the cycles, and the results in Figures 4(a), 4(b), and 4(c) correspond to the CoFs of 0, 5%-15%, and 60%-80%, respectively. It can be seen that when the planned operation succeeds, the confidence value of the component along the corresponding direction will keep decreasing with the cycles. When the CoF is low, the overall trend decreases due to the limited number of failures, although the confidence values will increase upon failure due to the penalty. Conversely, when the CoF is high, the overall trend of the confidence values increases because of the predominance of punishment due to the excessive number of failures. Figure 5 shows the trend of the confidence values of component F1 along the Y-direction with the cycles, which followings the same pattern as the results in Figure 4.

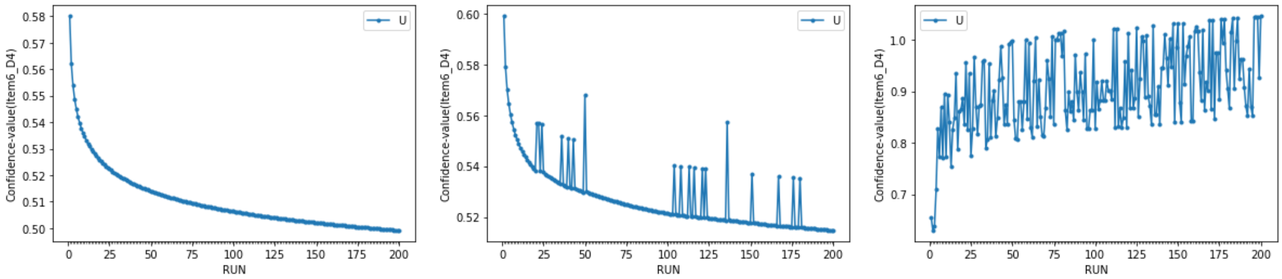

Figure 4(a). CoF is 0 Figure 4(b). CoF is 5%-15% Figure 4(c). CoF is 60%-80%

Figure 4. The variation of the confidence value of component C1 along the X- direction under different CoFs

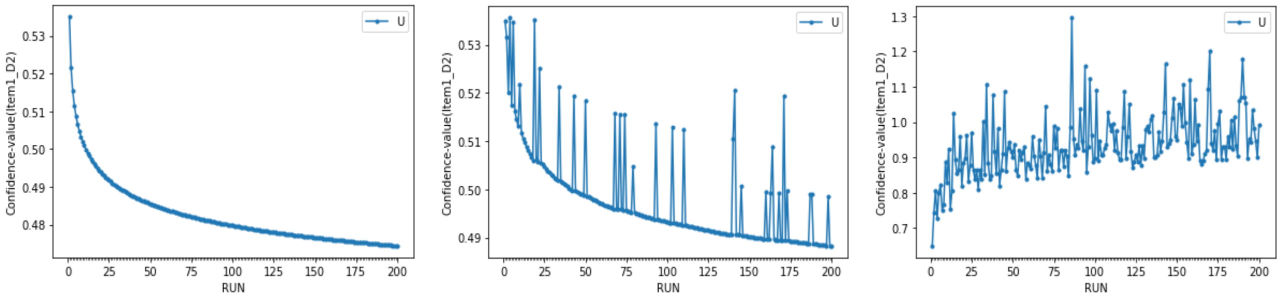

Figure 5(a). CoF is 0 Figure 5(b). CoF is 5%-15% Figure 5(c). CoF is 60%-80%

Figure 5. The variation of the confidence value of component F1 along the X- direction under different CoFs

5. Iterative estimation and ranking systems

There are two feedback loops within the program. The first determines passes and failures, defining each sequence while creating penalties and updating the confidence matrix. The second extrapolates the final data and feeds it back into the first loop through both feasibility and confidence matrices. This section will describe this final data, how it is calculated and fed back into the system.

5.1 Steady-state Feasibility/Confidence

After simulating a significant number of cycles, steady-state F and U (feasibility & confidence) matrices are acquired. This is achieved by storing the removed component data in Aggregate matrices, then calculating the average values. Thus, two new matrices will more accurately depict the product conditions at the end of the program. Table 3(A-C) present the steady-state outputs for example 1 under *Training data 1a*. Table 3(C) uses color shades to indicate the level of confidence—the lighter the color, the lower the confidence values, and vice versa. Initial F and U matrices are available for comparison in Table 2.

Table 3(A-C). Example 1a A) Training Data and B&C) Steady-state Matrices

A) Training Data 1a

	CoF
C1	8 - 12%
C2	8 - 12%
C3	8 - 12%
C4	8 - 12%
C5	8 - 12%
F1	15 - 20%
F2	8 - 12%
F3	8 - 12%
F4	8 - 12%

B) Compressed mean F 1a

	X+	X-	Y+	Y-
C1	4.70	4.70	2.45	5.20
C2	1.98	1.98	2.00	2.71
C3	2.60	2.30	2.16	2.60
C4	2.97	2.97	3.05	2.51
C5	3.90	3.90	5.90	3.39
F1	2.00	2.00	2.00	2.09
F2	3.20	2.30	2.29	2.00
F3	2.60	2.60	3.20	2.19
F4	2.00	3.20	2.00	2.08

C) Compressed mean U 1a

	X+	X-	Y+	Y-
C1	0.50	0.50	0.03	0.40
C2	0.06	0.06	0.00	0.51
C3	0.20	0.10	0.33	0.20
C4	0.23	0.23	0.31	0.06
C5	0.07	0.07	0.30	0.03
F1	0.00	0.00	0.00	0.52
F2	0.40	0.10	0.33	0.00
F3	0.20	0.20	0.40	0.13
F4	0.00	0.40	0.00	0.14

Once a cycle has finished, the data is fed back into the system in two ways:

①The new confidence is set equal to mean aggregate confidence, resulting in an almost fixed U matrix formed over time (Equation 8). Interestingly, if the confidence matrix is given the freedom to maintain its values from one cycle to the next, an extremely dynamic system would be created for real-time, in-situ use.

$$U_{new} = \frac{Total\_aggregate\_U}{Run\_size} \quad (8)$$

where U_{new} *is* the new confidence matrix for current cycle, *Total_aggregate_U represents the s*um of all Aggregate Confidence matrices and *Run_size* indicates the number of cycles completed.

②The new ceasibility matrix is updated to accommodate the penalties of the last iteration. Then a balancer matrix, equal to the sum of these penalties, is subtracted. This prevents the matrix from converging. Please note that both the penalties and the balancer matrix will only affect the directional feasibility of components with contact interference (parallel, perpendicular, or thread interference). The method is explained in Equation 7.

$$F_{new} = (F_{store} + P_{aggregate}) - \left(\frac{\sum P_{aggregate}}{N_{U-contact}} \begin{bmatrix} \frac{U_{c1x+}^{c1}}{U_{c1x+}^{c1}} & \frac{U_{c1x-}^{c2}}{U_{c1x-}^{c2}} & \cdots & \frac{U_{c1x-}^{f4}}{U_{c1x-}^{f4}} \\ \frac{U_{c2x+}^{c1}}{U_{c2x+}^{c1}} & \frac{U_{c2x-}^{c2}}{U_{c2x-}^{c2}} & \cdots & \frac{U_{c2y-}^{f4}}{U_{c2y-}^{f4}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{U_{f4x+}^{c1}}{U_{f4x+}^{c1}} & \frac{U_{f4x-}^{c2}}{U_{f4x-}^{c2}} & \cdots & \frac{U_{f4y-}^{f4}}{U_{f4y-}^{f4}} \end{bmatrix} \right) \quad (9)$$

where $N_{u\text{-}contacts}$ is the number of contacting components with parallel, perpendicular or thread interference. This is multiplied by a binary matrix representing the location of contacts. The $P_{aggregate}$ matrix is set to zero after this calculation, and F_{new} becomes equal to F_{store} for next cycles use.

5.2 Recommended disassembly sequence plans

As shown in Figure 3, before performing the iterative operation, it is necessary to generate a layered candidate set based on space constraints. The component with less interference has a higher priority and a more advanced level. Then, after simulating a significant number of cycles, the sequence obtained can be determined as the systems overall DSP recommendation.

Continuing with example 1a, its layered candidate set contains 3 layers, denoted as {F1, F2, F3, F4| C1, C2, C3, C5| C4}.Then the program runs for 100 cycles to obtain the steady-state feasibility and confidence matrices. Given F1’s failure rate and relationships with C1 and C2, the proposed DSP method effectively adapt to this uncertainty by disassembling the bottom half of the product first. Furthermore, the direction of disassembly is predominantly in the Y-, which is a reasonable suggestion considering the internal elements are blocked by C1. Table 4 lists three different DSPs in order of cycles, illustrating that the FDSP offers the capability for DSP to adapt to failures and self-evolve online. For Example 1a, the 73rd cycle received the recommended disassembly sequence.

Table 4. Three different DSPs in 100-cycle run of Example 1a

Cycle	Cycle = 100, penalty cost = 0.1								
1	F4	F3	F2	F1	C5	C1	C2	C3	C4
	Y-	Y-	Y+	Y-	Y-	Y+	Y+	Y+	Free
29	F4	F3	F2	C5	F1	C1	C4	C3	C2
	Y-	Y-	Y+	Y-	Y-	Y+	Y-	Y-	Y-
73	F3	F4	F2	C5	C4	C3	F1	C1	C2
	Y-	Y-	Y+	Y-	Y-	Y-	Y-	Y+	Y-

6. Experiments and discussion

To validate the performance of the proposed re-planning, dynamic uncertainty approach, example 1 is revisited with new training data, and 2 new products (Product B and Product C) varying in structure and complexity are considered in six directions (X+, X-, Y+, Y-, Z+, Z-) for performance validation. The component relationships of these two new products cover the removal relationships listed in Table 1. Therefore, they can be used as typical cases to verify the performance of the proposed method.

First, the CoFs of each product component are set in a range rather than a fixed value to simulate the actual situation. Second, the proposed fuzzification DSP algorithm (FDSPA) is applied to provide the re-planning solutions for the three products in both solution quality and time consumption. FDSPA will be compared with our previous research [1] in this paper. According to section 5, when the value of U converges, it is considered that the recommended disassembly sequence is the optimal sequence at this time, and the feasibility and superiority of this sequence will be evaluated. The time consumption directly determines whether the robot can cope with disassembly failures online in real-time, which is an essential indicator of the quality of the re-planning algorithms.

All experiments are implemented by Python and tested in a Windows 10 environment. The hardware configuration is 3.6-GHz Intel Core i7-7700 CPU with 32 GB of 2,400-MHz DDR4 RAM.

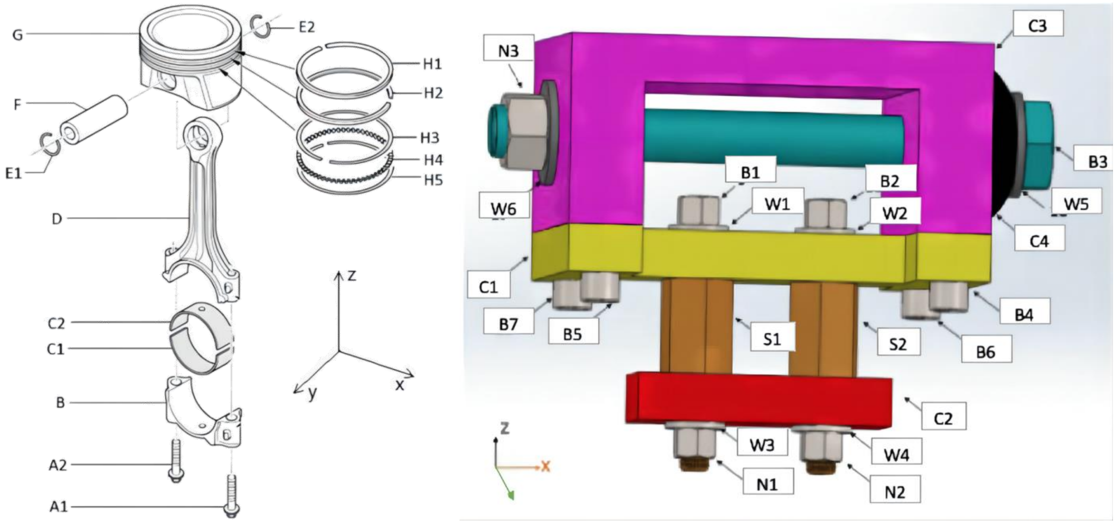

Figure 6(a). Product B

Figure 6(b). Product C

Figure 6. Two typical products applied to validate the method proposed in this study (Laili et al. 2019 [23])

6.1 Performance analysis of the solution quality

6.1.1 Example 1

To assess the plasticity of the method, the training data of Example 1 (Figure 1) is altered to increase the frequency of failures within the bottom half, i.e., the CoFs of F2 and F3 have been significantly increased. When F and U reach a steady state, the sequence generated demonstrates that the system has successfully adapted to the changes in product condition by dismantling the top half first. The overall results can be seen in Table 5. It is important to note that no run is the same, individual cycles can vary greatly due to noise in the training data, and in many cases the program can be creative.

Table 5(A-C). Example 1b B&C) Feasibility & Confidence matrices under A) stated training conditions

A) Training Data 1b

	CoF
C1	8 - 12%
C2	8 - 12%
C3	8 - 12%
C4	8 - 12%
C5	8 - 12%
F1	8 - 12%
F2	12 - 20%
F3	12 - 20%
F4	8 - 12%

B) Compressed mean F 1b

	X+	X-	Y+	Y-
C1	5.98	5.98	2.95	5.20
C2	2.91	2.91	2.00	3.02
C3	2.60	2.30	2.89	2.60
C4	3.44	3.44	3.31	3.79
C5	4.71	4.71	5.90	3.83
F1	2.00	2.00	2.00	4.19
F2	3.20	2.30	6.51	2.00
F3	2.60	2.60	3.20	5.59
F4	2.00	3.20	2.00	3.46

C) Compressed mean U 1b

	X+	X-	Y+	Y-
C1	0.47	0.47	0.03	0.40
C2	0.05	0.05	0.00	0.51
C3	0.20	0.10	0.33	0.20
C4	0.23	0.23	0.31	0.05
C5	0.05	0.05	0.30	0.03
F1	0.00	0.00	0.00	0.49
F2	0.40	0.10	0.30	0.00
F3	0.20	0.20	0.40	0.10
F4	0.00	0.40	0.00	0.09

Table 6. The recommended DSP for Example 1b

Component	F4	F1	C2	C1	F3	F2	C3	C5	C4
Direction	Y-	Y-	Y-	Y+	Y-	Y+	Y+	Y-	Free

Hence, we can more or less conclude that a new and more efficient DSP was constructed based on the knowledge of past failures. Interpreting the steady-state confidence matrix can be difficult because of remote interference. The enormous uncertainty adds aesthetic noise to the matrix, as only contacting relations receive a penalty upon failure. Thus, remote interference has a rare effect on penalties (A few instances where components share contact and remote interference, where remote is the most significant interference.), yet it is still vital to the initial conditions.

6.1.2 Example 2

Now let us consider a 3D example in 6 directions. For Product B in Figure 6(a), when set up with a relatively balanced training conditions, i.e., CoF ranges between 8-20% for each component. The layered candidate set and the recommended disassembly sequence are shown in Table 7.

Table 7. Results of Example 2a

A) Training Data 2a

	CoF
A1	8 - 20%
A2	8 - 20%
B	8 - 20%
C1	8 - 20%
C2	8 - 20%
D	8 - 20%
E1	8 - 20%
E2	8 - 20%
F	8 - 20%
G	8 - 20%
H1	8 - 20%
H2	8 - 20%
H3	8 - 20%
H4	8 - 20%
H5	8 - 20%

B) Compressed mean F 2a

	X+	X-	Y+	Y-	Z+	Z-
A1	2.00	2.90	2.00	2.00	2.30	1.05
A2	2.90	2.00	2.00	2.00	2.30	2.16
B	3.89	3.89	3.89	3.89	7.10	2.53
C1	2.32	2.32	2.02	2.02	2.78	1.43
C2	2.26	2.26	1.96	1.96	1.99	2.17
D	5.96	5.96	6.26	6.26	6.06	6.00
E1	1.35	1.35	0.82	2.60	1.35	1.35
E2	1.30	1.30	2.60	0.72	1.30	1.30
F	2.65	2.65	2.62	2.62	2.65	3.55
G	9.00	9.00	8.75	8.75	8.00	10.50
H1	1.00	1.00	1.00	1.00	0.67	5.60
H2	1.00	1.00	1.00	1.00	2.41	4.72
H3	1.00	1.00	1.00	1.00	3.04	3.88
H4	1.00	1.00	1.00	1.00	4.41	2.81
H5	1.00	1.00	1.00	1.00	4.65	1.88

C) Compressed mean U 2a

	X+	X-	Y+	Y-	Z+	Z-
A1	0.00	0.30	0.00	0.00	0.10	0.13
A2	0.30	0.00	0.00	0.00	0.10	0.13
B	0.07	0.07	0.07	0.07	0.70	0.05
C1	0.17	0.17	0.07	0.07	0.22	0.03
C2	0.17	0.17	0.07	0.07	0.23	0.01
D	0.07	0.07	0.17	0.17	0.55	0.00
E1	0.03	0.03	0.08	0.20	0.03	0.03
E2	0.03	0.03	0.20	0.07	0.03	0.03
F	0.07	0.07	0.14	0.14	0.07	0.37
G	0.00	0.00	0.07	0.07	0.00	0.50
H1	0.00	0.00	0.00	0.00	0.07	0.20
H2	0.00	0.00	0.00	0.00	0.08	0.21
H3	0.00	0.00	0.00	0.00	0.08	0.22
H4	0.00	0.00	0.00	0.00	0.08	0.21
H5	0.00	0.00	0.00	0.00	0.06	0.22

D) The layered candidate set for Example 2a

{A1, A2, E1, E2, H1 | B, F, H2 | C1, H3 | C2, H4 | D, H5 | G }

E) The recommended DSP for Example 2a

Component	E1	E2	H1	A1	A2	F	B	H2	C1	H3	C2	H4	D	H5	G
Direction	Y+	Y-	Z+	Z-	Z-	Y-	Z-	Z+	Z-	Z+	Z-	Z+	Z-	Z+	Free

This is a reasonable and valid DSP. One critique would be that adjacent components could be removed in a more sensible order, but this is understandable because the algorithm focuses on the effects of space constraints and CoFs on the disassembly sequence rather than considering the impact of disassembly time, disassembly tools, etc. Therefore, the program is removing them in order of how guaranteed it believes their removal to be.

So once again, to validate the performance and adaptability of the algorithm, the CoFs of E1, E2, and F are raised to 2 times as often. The result is shown in Table 8.

Table 8. Results of Example 2b

A) Training Data 2b

	CoF
A1	8 - 20%
A2	8 - 20%
B	8 - 20%
C1	8 - 20%
C2	8 - 20%
D	8 - 20%
E1	16 - 40%
E2	16 - 40%
F	16 - 40%
G	8 - 20%
H1	8 - 20%
H2	8 - 20%
H3	8 - 20%
H4	8 - 20%
H5	8 - 20%

B) Compressed mean F 2b

	X+	X-	Y+	Y-	Z+	Z-
A1	2.00	2.90	2.00	2.00	2.30	0.63
A2	2.90	2.00	2.00	2.00	2.30	0.64
B	3.33	3.33	3.33	3.33	7.10	2.24
C1	1.63	1.63	1.33	1.33	2.68	1.16
C2	1.63	1.63	1.33	1.33	1.76	2.07
D	5.33	5.33	5.63	5.63	5.73	6.00
E1	1.33	1.33	0.76	2.60	1.33	1.33
E2	1.29	1.29	2.60	0.67	1.29	1.29
F	2.63	2.63	2.19	2.21	2.63	3.53
G	9.00	9.00	8.51	8.51	8.00	10.50
H1	1.00	1.00	1.00	1.00	0.40	5.60
H2	1.00	1.00	1.00	1.00	1.43	4.68
H3	1.00	1.00	1.00	1.00	2.46	3.68
H4	1.00	1.00	1.00	1.00	3.37	2.71
H5	1.00	1.00	1.00	1.00	4.33	1.67

C) Compressed mean U 2b

	X+	X-	Y+	Y-	Z+	Z-
A1	0.00	0.30	0.00	0.00	0.10	0.12
A2	0.30	0.00	0.00	0.00	0.10	0.12
B	0.07	0.07	0.07	0.07	0.70	0.04
C1	0.17	0.17	0.07	0.07	0.21	0.03
C2	0.17	0.17	0.07	0.07	0.23	0.01
D	0.07	0.07	0.17	0.17	0.54	0.00
E1	0.05	0.05	0.10	0.20	0.05	0.05
E2	0.05	0.05	0.20	0.10	0.05	0.05
F	0.09	0.09	0.17	0.17	0.09	0.39
G	0.00	0.00	0.07	0.07	0.00	0.50
H1	0.00	0.00	0.00	0.00	0.09	0.20
H2	0.00	0.00	0.00	0.00	0.08	0.22
H3	0.00	0.00	0.00	0.00	0.11	0.22
H4	0.00	0.00	0.00	0.00	0.08	0.23
H5	0.00	0.00	0.00	0.00	0.06	0.21

D) The recommended DSP for Example 2b

Component	H1	A1	A2	E2	E1	B	H2	F	C1	H3	C2	H4	D	H5	G
Direction	Z+	Z-	Z-	Y-	Y+	Z-	Z+	Y+	Z-	Z+	Z-	Z+	Z-	Z+	Free

This is a successful experiment, and the overall system adapts very well to the changes in input. Compared with the results of Example 2a, E1 and E2 are ranked after A1, A2, and H1, at the end of the first layer. Likewise, F is ranked at the very end of the second layer.

6.1.3 Example 3

In the last example, Product C has 22 parts, as shown in Figure 6(b). The recommended DSP is shown in Table 9. This would be a sensible sequence, but for B1 and B2 being removed before the top half of the assembly. The reason for recommending this sequence is that the algorithm considers this remote interference to be ignored under the condition that the initial values of feasibility and confidence are fixed, respectively. However, the conditions of remote interference are not exactly the same, e.g., remote interference varies with the geometry and size of the assembly, so further refinement is needed according to the actual situation.

Table 9. Results of Example 3

A) Training Data 3

	CoF
C1	10 - 15%
S1	10 - 15%
S2	10 - 15%
C2	10 - 15%
W1	10 - 15%
W2	10 - 15%
W3	10 - 15%
W4	10 - 15%
B1	10 - 15%
B2	10 - 15%
N1	10 - 15%
N2	10 - 15%
C3	10 - 15%
C4	10 - 15%
B3	10 - 15%
W5	10 - 15%
W6	10 - 15%
N3	10 - 15%
B4	10 - 15%
B5	10 - 15%
B6	10 - 15%
B7	10 - 15%

B) Compressed mean F 3

	X+	X-	Y+	Y-	Z+	Z-
C1	6.66	6.66	6.66	6.66	6.40	8.18
S1	2.46	1.86	1.26	1.26	2.96	1.91
S2	1.85	2.45	1.25	1.25	2.96	1.89
C2	2.50	2.50	2.50	2.50	5.62	3.20
W1	2.02	1.42	1.12	1.12	1.66	2.44
W2	1.43	2.03	1.13	1.13	1.66	2.44
W3	1.85	1.25	1.25	1.25	3.56	1.32
W4	1.27	1.87	1.27	1.27	3.58	1.28
B1	8.10	6.30	6.00	6.00	2.82	6.00
B2	6.60	7.80	6.00	6.00	2.76	6.00
N1	1.72	1.12	1.12	1.12	3.80	0.29
N2	1.16	1.76	1.16	1.16	3.80	0.39
C3	7.72	6.95	5.40	5.40	5.32	9.54
C4	6.36	5.90	5.25	5.25	5.25	5.25
B3	1.96	5.00	5.00	5.00	5.00	9.80
W5	1.06	2.13	1.13	1.13	1.13	1.13
W6	2.66	1.31	1.26	1.26	1.26	1.26
N3	2.90	0.32	1.13	1.13	1.13	1.13
B4	3.00	3.90	3.00	3.30	3.30	1.56
B5	3.90	3.00	3.00	3.30	3.30	1.50
B6	3.00	3.90	3.30	3.00	3.30	1.45
B7	3.90	3.00	3.30	3.00	3.30	1.45

C) Compressed mean U 3

	X+	X-	Y+	Y-	Z+	Z-
C1	0.19	0.19	0.19	0.19	0.39	0.68
S1	0.47	0.27	0.07	0.07	0.31	0.29
S2	0.27	0.47	0.07	0.07	0.31	0.28
C2	0.14	0.14	0.14	0.14	0.53	0.35
W1	0.33	0.13	0.03	0.03	0.21	0.46
W2	0.14	0.34	0.04	0.04	0.22	0.46
W3	0.27	0.07	0.07	0.07	0.51	0.09
W4	0.07	0.27	0.07	0.07	0.52	0.07
B1	0.70	0.10	0.00	0.00	0.52	0.00
B2	0.20	0.60	0.00	0.00	0.51	0.00
N1	0.23	0.03	0.03	0.03	0.60	0.07
N2	0.04	0.24	0.04	0.04	0.60	0.08
C3	0.56	0.62	0.12	0.12	0.10	1.17
C4	0.11	0.28	0.07	0.07	0.07	0.07
B3	0.25	0.00	0.00	0.00	0.00	1.60
W5	0.01	0.36	0.03	0.03	0.03	0.03
W6	0.22	0.09	0.08	0.08	0.08	0.08
N3	0.30	0.09	0.04	0.04	0.04	0.04
B4	0.00	0.30	0.00	0.10	0.10	0.13
B5	0.30	0.00	0.00	0.10	0.10	0.13
B6	0.00	0.30	0.10	0.00	0.10	0.12
B7	0.30	0.00	0.10	0.00	0.10	0.12

D) The layered candidate set for Example 3

{N1, N2, N3, B4, B5, B6, B7 | W3, W4, B1, B2, B3, W6 | C1, S1, S2, C2, W1, W2, C3, C4, W5}

E) The recommended DSP for Example 3

Component	N1	N3	N2	B4	B7	B5	B6	W3	W6	W4	B3	B1	B2	W2	W5	W1	C4	C3	C1	S2	C2	S1
Direction	Z-	X-	Z-	Z-	Z-	Z-	Z-	Z-	X-	Z-	X+	Z+	Z+	Z+	X+	Z+	X+	Z+	Z+	Z+	Z-	Free

The above examples demonstrate FDSPA's ability to sequence re-planning under uncertainties by constantly updating the actual state of the product through learning from historical knowledge and thus recommending valid disassembly sequences. Furthermore, for the same product, when the CoFs of its components are different, the system can adapt well to such changes and form new valid sequences, e.g., Example 1b concerning Example 1a and Example 2b with respect to Example 2a.

6.2 Performance analysis of the time consumption

To ensure that the robot can automatically cope with disassembly failures, the algorithm must be able to make quick decisions online. In our earlier work [1], the ternary bees algorithm (TBA) was proposed and the results showed that TBAs have an advantage over other ad-hoc meta-heuristics in terms of time consumption for re-planning. In this paper, we will verify the performance of FDSPA by comparing it with TBAs. Considering that the search time does increase significantly without much performance improvement when the population size of TBAs increases, TBA-3 is chosen to solve the disassembly re-planning problem [1]. Since each component has a chance of failure, the algorithm is run 20 times to calculate the average execution time.

In this paper, 3 products are studied, where both Product A and Product B have 2 sets of data and Product C has one set of data, so there are 5 cases in total. The average execution time of the 2 algorithms on 5 disassembly cases is shown in Table 10. The results show that the performance of the two algorithms is similar in terms of time efficiency. Specifically, the FDSPA is more advantageous when there are fewer parts in the product, while the TBA is superior when there are more components. In addition, the re-planning time using the FDSPA grows with the number of components in a product, which is consistent with the evolutionary mechanism of the algorithm, i.e., when a disassembly operation fails, both the component and its contacting components will receive a variable feasibility penalty. The more components are penalized, the more time is consumed. Therefore, the FDSPA-based re-planning time grows as the number of components included in the product increases. However, this trend does not continue indefinitely as there is an upper limit to the number of components in contact with a component due to dimensional and geometry constraints. In contrast, the search time of TBA is influenced more by the number of population size and iterations than the number of components.

Table 10. Execution time for the 2 algorithms on 5 disassembly cases (The unit of time is millisecond)

No. of case	1	2	3	4	5
TBA-3	68.0811	73.6395	96.8605	102.3686	109.3639
FDSPA	19.3631	22.1930	87.411	91.413	184.1068

In summary, the proposed FDSPA performs well in both solution quality and time consumption. Faced with uncertain interference conditions of EoL products, FDSPA is able to continuously learn from historical operations and combines fuzzy dynamic modeling methods with self-evolutionary strategies, making the algorithm capable of coping with failures. The re-planning speed of the FDSPA proves to be very fast and can generate new sequences quickly, which ensures the possibility of online re-planning and automatic disassembly.

7. Conclusion

This paper proposes a disassembly sequence planning method for products with uncertain interference, using a fuzzy ranking scheme to evaluate the predicted product condition. Using two fuzzy dynamic matrices combined with a traditional contact matrix, the assembly is defined in 3D space, and informed online removal decisions are made based on the previous actions through Monte Carlo forward uncertainty propagation and iterative estimation. The re-planning algorithm is designed for complete disassembly that can be easily adapted to different product conditions in real-time.

It has successfully provided reasonable and creative solutions, including the use of layered candidate sets to circumvent the generation of irrational sequences, the introduction of a penalty value matrix to evaluate the cost of changing components and directions, recommending disassembly sequences based on steady-state matrices, etc., which depict the product condition with greater accuracy than predicted.

Utilized as an analytical tool in this study, the method proved to generate valid sequences through multiple outputs. However, there are still some limitations of this paper. On the one hand, the interference between components is not sufficiently detailed, i.e., the definition of remote interference is relatively coarse. On the other hand, the current work only considers the space constraints, with the primary goal of obtaining a feasible DSP. Future work will take into account the conditions of time, value, and tools to generate sequences that satisfy multiple objectives.

Acknowledgement

This work is supported by the National Key Research and Development Program of China under Grant (Grant No. 2018YFB1700603), the National Natural Science Foundation of China (NSFC) (Grant No. 62173017), the Royal Society (IEC\NSFC\181018) and by the Engineering and Physical Sciences Research Council (EPSRC) under the funded projects AUTOREMAN (EP/N018524/1) and ATARI (EP/W00206X/1).

References

[1] Laili Y, Tao F, Pham D T, et al. Robotic disassembly re-planning using a two-pointer detection strategy and a super-fast bees algorithm[J]. Robotics and Computer-Integrated Manufacturing, 2019, 59: 130-142.

[2] Sanjeev Kumar R, Padmanaban K P, Rajkumar M. Minimizing makespan and total flow time in permutation flow shop scheduling problems using modified gravitational emulation local search algorithm[J]. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 2018, 232(3): 534-545.

[3] Zheng Z, Xu W, Zhou Z, et al. Dynamic modeling of manufacturing capability for robotic disassembly in remanufacturing[J]. Procedia Manufacturing, 2017, 10: 15-25.

[4] Ilgin M A, Gupta S M. Remanufacturing modeling and analysis[M]. CRC Press, 2012.

[5] Liu J, Zhou Z, Pham D T, et al. Robotic disassembly sequence planning using enhanced discrete bees algorithm in remanufacturing[J]. International Journal of Production Research, 2018, 56(9): 3134-3151.

[6] Zhou Z, Liu J, Pham D T, et al. Disassembly sequence planning: recent developments and future trends[J]. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 2019, 233(5): 1450-1471.

[7] Rickli J L, Camelio J A. Partial disassembly sequencing considering acquired end-of-life product age distributions[J]. International Journal of Production Research, 2014, 52(24): 7496-7512.

[8] Kim H W, Park C, Lee D H. Selective disassembly sequencing with random operation times in parallel disassembly environment[J]. International Journal of Production Research, 2018, 56(24): 7243-7257.

[9] Laili Y, Ye F, Wang Y, et al. Interference probability matrix for disassembly sequence planning under uncertain interference[J]. Journal of Manufacturing Systems, 2021, 60: 214-225.

[10] Ren Y, Zhang C, Zhao F, et al. An asynchronous parallel disassembly planning based on genetic algorithm[J]. European Journal of Operational Research, 2018, 269(2): 647-660.

[11] Tseng H E, Chang C C, Lee S C, et al. Hybrid bidirectional ant colony optimization (hybrid BACO): An algorithm for disassembly sequence planning[J]. Engineering Applications of Artificial Intelligence, 2019, 83: 45-56.

[12] Tseng H E, Chang C C, Lee S C, et al. A block-based genetic algorithm for disassembly sequence planning[J]. Expert Systems with Applications, 2018, 96: 492-505.

[13] Agrawal D, Pande S S. Automatic disassembly sequence planning and optimization[J]. Dual Degree Thesis, IIT Bombay, 2011.

[14] Tseng H E, Huang Y M, Chang C C, et al. Disassembly sequence planning using a Flatworm algorithm[J]. Journal of Manufacturing Systems, 2020, 57: 416-428.

[15] Wang J, Allada V. Hierarchical fuzzy neural network-based serviceability evaluation[J]. International Journal of Agile Management Systems, 2000, 2(2):130-141.

[16] Tang Y, Zhou M C, Gao M. Fuzzy-Petri-net-based disassembly planning considering human factors[J]. IEEE Transactions on Systems, Man, and Cybernetics-Part A: systems and humans, 2006, 36(4): 718-726.

[17] Behdad S, Berg L, Vance J, et al. Immersive computing technology to investigate tradeoffs under uncertainty in disassembly sequence planning[J]. Journal of Mechanical Design, 2014, 136(7): 071001.

[18] Tian G, Zhou M C, Li P. Disassembly sequence planning considering fuzzy component quality and varying operational cost[J]. IEEE Transactions on Automation Science and Engineering, 2017, 15(2): 748-760.

[19] Zhang X F, Yu G, Hu Z Y, et al. Parallel disassembly sequence planning for complex products based on fuzzy-rough sets[J]. The International Journal of Advanced Manufacturing Technology, 2014, 72(1-4): 231-239.

[20] Perskaya V V, Revenko N I. Made in China 2025: chinese experience in achieving national development goals[J]. Asia and Africa today, 2020 (7): 19-25.

[21] Kerin M, Pham D T. A review of emerging industry 4.0 technologies in remanufacturing[J]. Journal of cleaner production, 2019, 237: 117805.

[22] Cao J, Chen X, Zhang X, et al. Overview of remanufacturing industry in China: Government policies, enterprise, and public awareness[J]. Journal of Cleaner Production, 2020, 242: 118450.

[23] Aguinaga I, Borro D, Matey L. Parallel RRT-based path planning for selective disassembly planning[J]. The International Journal of Advanced Manufacturing Technology, 2008, 36(11): 1221-1233.

[24] Boujault A. Contribution à une approche méthodologique de l'assemblage automatisé: élaboration automatique des séquences opératoires[J]. Thése d'Etat, Université de Franche-Comté, 1984.

[25] De Fazio T, Whitney D. Simplified generation of all mechanical assembly sequences[J]. IEEE Journal on Robotics and Automation, 1987, 3(6): 640-658.

[26] Zhang H C, Kuo T C. A graph-based disassembly sequence planning for EOL product recycling[C]//Twenty First IEEE/CPMT International Electronics Manufacturing Technology Symposium Proceedings 1997 IEMT Symposium. IEEE, 1997: 140-151.

[27] Xia K, Gao L, Wang L, et al. Service-oriented disassembly sequence planning for electrical and electronic equipment waste[J]. Electronic Commerce Research and Applications, 2016, 20: 59-68.

[28] Fang H C, Ong S K, Nee A Y C. An integrated approach for product remanufacturing assessment and planning[J]. Procedia CIRP, 2016, 40: 262-267.

[29] Yu B, Wu E, Chen C, et al. A general approach to optimize disassembly sequence planning based on disassembly network: A case study from automotive industry[J]. Advances in Production Engineering & Management, 2017, 12(4): 305-320.

[30] Song X, Zhou W, Pan X, et al. Disassembly sequence planning for electro-mechanical products under a partial destructive mode[J]. Assembly Automation, 2014.

[31] Altekin F T, Kandiller L, Ozdemirel N E. Profit-oriented disassembly-line balancing[J]. International Journal of Production Research, 2008, 46(10): 2675-2693.

[32] Koc A, Sabuncuoglu I, Erel E. Two exact formulations for disassembly line balancing problems with task precedence diagram construction using an AND/OR graph[J]. Iie Transactions, 2009, 41(10): 866-881.

[33] Tian G, Liu Y, Tian Q, et al. Evaluation model and algorithm of product disassembly process with stochastic feature[J]. Clean Technologies and Environmental Policy, 2012, 14(2): 345-356.

[34] Li J R, Khoo L P, Tor S B. An object-oriented intelligent disassembly sequence planner for maintenance[J]. Computers in Industry, 2005, 56(7): 699-718.

[35] Pintzos G, Triantafyllou C, Papakostas N, et al. Assembly precedence diagram generation through assembly tiers determination[J]. International Journal of Computer Integrated Manufacturing, 2016, 29(10): 1045-1057.

[36] Mitrouchev P, Wang C G, Lu L X, et al. Selective disassembly sequence generation based on lowest level disassembly graph method[J]. The International Journal of Advanced Manufacturing Technology, 2015, 80(1): 141-159.

[37] Kalayci C B, Polat O, Gupta S M. A variable neighbourhood search algorithm for disassembly lines[J]. Journal of Manufacturing Technology Management, 2015.

[38] Zhang W, Ma M, Li H, et al. Generating interference matrices for automatic assembly sequence planning[J]. The International Journal of Advanced Manufacturing Technology, 2017, 90(1): 1187-1201.

[39] Dini G, Santochi M. Automated sequencing and subassembly detection in assembly planning[J]. CIRP annals, 1992, 41(1): 1-4.

[40] Pornsing C, Watanasungsuit A. Discrete particle swarm optimization for disassembly sequence planning[C]//2014 IEEE International Conference on Management of Innovation and Technology. IEEE, 2014: 480-485.

[41] Chung C, Peng Q. A hybrid approach to selective-disassembly sequence planning for de-manufacturing and its implementation on the Internet[J]. The International Journal of Advanced Manufacturing Technology, 2006, 30(5): 521-529.

[42] Luo Y, Peng Q. Disassembly sequence planning for product maintenance[C]//International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers, 2012, 45042: 601-609.

[43] Huang J, Esmaeilian B, Behdad S. Multi-purpose disassembly sequence planning[C]//International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers, 2015, 57113: V004T05A006.

[44] Lu C, Liu Y C. A disassembly sequence planning approach with an advanced immune algorithm[J]. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2012, 226(11): 2739-2749.

[45] Zhao S, Li Y, Fu R, et al. Fuzzy reasoning Petri nets and its application to disassembly sequence decision-making for the end-of-life product recycling and remanufacturing[J]. International Journal of Computer Integrated Manufacturing, 2014, 27(5): 415-421.

[46] Xia K, Gao L, Li W, et al. A Q-learning based selective disassembly planning service in the cloud based remanufacturing system for WEEE[C]//International Manufacturing Science and Engineering Conference. American Society of Mechanical Engineers, 2014, 45806: V001T04A012.

[47] Guo X, Liu S, Zhou M C, et al. Disassembly sequence optimization for large-scale products with multiresource constraints using scatter search and Petri nets[J]. IEEE transactions on cybernetics, 2015, 46(11): 2435-2446.

[48] Tang Y, Zhou M C, Caudill R J. An integrated approach to disassembly planning and demanufacturing operation[J]. IEEE Transactions on Robotics and Automation, 2001, 17(6): 773-784.

[49] Grochowski D E, Tang Y. A machine learning approach for optimal disassembly planning[J]. International Journal of Computer Integrated Manufacturing, 2009, 22(4): 374-383.

[50] Kuo T C. Waste electronics and electrical equipment disassembly and recycling using Petri net analysis: Considering the economic value and environmental impacts[J]. Computers & Industrial Engineering, 2013, 65(1):

54-64.

[51] Hsu H P. A fuzzy knowledge-based disassembly process planning system based on fuzzy attributed and timed predicate/transition net[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2016, 47(8): 1800-1813.

[52] Han H J, Yu J M, Lee D H. Mathematical model and solution algorithms for selective disassembly sequencing with multiple target components and sequence-dependent setups[J]. International Journal of Production Research, 2013, 51(16): 4997-5010.

[53] Zhu B, Sarigecili M I, Roy U. Disassembly information model incorporating dynamic capabilities for disassembly sequence generation[J]. Robotics and Computer-Integrated Manufacturing, 2013, 29(5): 396-409.

[54] Li W D, Xia K, Gao L, et al. Selective disassembly planning for waste electrical and electronic equipment with case studies on liquid crystaldisplays[J]. Robotics and Computer-Integrated Manufacturing, 2013, 29(4): 248-260.

[55] Igarashi K, Yamada T, Gupta S M, et al. Disassembly system modeling and design with parts selection for cost, recycling and CO2 saving rates using multi criteria optimization[J]. Journal of Manufacturing Systems, 2016, 38: 151-164.

[56] Tian G, Zhou M C, Chu J, et al. Probability evaluation models of product disassembly cost subject to random removal time and different removal labor cost[J]. IEEE Transactions on Automation Science and Engineering, 2012, 9(2): 288-295.

[57] Smith S S, Chen W H. Rule-based recursive selective disassembly sequence planning for green design[J]. Advanced Engineering Informatics, 2011, 25(1): 77-87.

[58] Gungor A, Gupta S M. Disassembly sequence planning for products with defective parts in product recovery[J]. Computers & Industrial Engineering, 1998, 35(1-2): 161-164.

[59] Zussman E, Zhou M. A methodology for modeling and adaptive planning of disassembly processes[J]. IEEE Transactions on Robotics and Automation, 1999, 15(1): 190-194.

[60] ElSayed A, Kongar E, Gupta S M, et al. A robotic-driven disassembly sequence generator for end-of-life electronic products[J]. Journal of Intelligent & Robotic Systems, 2012, 68(1): 43-52.

[61] Wang Y, Lan F, Liu J, et al. Interlocking problems in disassembly sequence planning[J]. International Journal of Production Research, 2021, 59(15): 4723-4735.

[62] Jin G, Li W, Wang S, et al. A systematic selective disassembly approach for waste electrical and electronic equipment with case study on liquid crystal display televisions[J]. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 2017, 231(13): 2261-2278.

[63] Guo X, Zhou M C, Abusorrah A, et al. Disassembly sequence planning: a survey[J]. IEEE/CAA Journal of Automatica Sinica, 2020, 8(7): 1308-1324.