Interlocking Problem in Automatic Disassembly Planning and Two Solutions

Lan, Feiying; Wang, Yongjing Wang; Pham, Duc; Liu, Jiayi; Huang, Jun; Ji, Chunqian; Su, Shizhong; Xu, Wenjun; Liu, Quan; Zhou, Zude

Interlocking Problem in Automatic Disassembly Planning and Two Solutions

Feiying Lan[1], Yongjing Wang[1(✉)], Duc Truong Pham[1], Jiayi Liu[1,2], Jun Huang[1], Chunqian Ji[1], Shizhong Su[1], Wenjun Xu[2], Quan Liu[2], and Zude Zhou[2]

[1] The University of Birmingham, Edgbaston, Birmingham B15 2TT, UK
{fxl655, jxll0l2}@student.bham.ac.uk,
{y.wang, d.t.pham, j.huang.l, c.ji, s.su.2}@bham.ac.uk
[2] Wuhan University of Technology, Wuhan 430070, China
{xuwenjun, quanliu, zudezhou}@whut.edu.cn

Abstract. In remanufacturing, disassembly is the first step to dismantle the end-of-life products into components, which is labour-intensive due to the variability of returned products. Compared to manual disassembly, robotic disassembly is a promising technique to automate remanufacturing processes, which liberates the human labours from the repetitive disassembly operations. However, it requires predesigned disassembly sequences which are planned manually. Several planning methods have been proposed to remove removable parts sequentially. However, those methods can fail in the disassembly sequence planning task if the product has interlocked components. This paper first explains the interlocking problem and then proposes two solutions. One solution is to identify subassemblies by using 'separable pairs'. It complements conventional sequential disassembly planning methods and enables automatic detection of subassemblies online. Another method is based on a divide-and-conquer disassembly strategy which allows subassemblies to be detected before disassembly. This approach generates disassembly sequence plans that are hierarchical to avoid interlocking problems and reduce computational complexity.

Keywords: Remanufacturing · Disassembly planning · Dismantling robotic disassembly · Divide-and-conquer strategy · Hierarchy analysis

1 Introduction

Remanufacturing is "the rebuilding of a product to specifications of the original manufactured product using a combination of reused, repaired and new parts" [6]. Disassembly is an important feature that distinguishes remanufacturing from conventional manufacturing. It is labour intensive with complex operations involved, which is carried out manually due to the variability in the condition of the returned products.

A study of the robotic disassembly of a PC [7] in the mid-1990s started the research on automated disassembly systems. Afterwards, the successful attempts at dismantling electrical devices and automotive components [1, 2, 14] showed more encouraging

O. Gusikhin and K. Madani (Eds.): ICINCO 2018, LNEE 613, pp. 193–213, 2020.
https://doi.org/10.1007/978-3-030-31993-9_9

results. These experiments were mostly product-oriented and based on pre-programmed sequences. Compared with 'automated disassembly with pre-programmed disassembly sequence, a key advance of 'autonomous' disassembly is that machine can interpret product structure and plan the disassembly sequence. A popular approach is based on graphs [8, 13]. Many algorithms and rule-based methods have been used to calculate disassembly sequences, for example, the Fuzzy Reasoning Petri Net proposed by Zhao and Li [16]. However, a graph cannot be generated exclusively by machine understanding. It requires human's interpretation.

Smith *et al.* proposed a method to plan disassembly sequences based on five matrices with several rules to represent an assembly [9, 11]. Tao *et al.* modified the matrices to enable partial/parallel disassembly [12]. However, the complexity of the mathematical representation of an assembly is not reduced in this optimisation-focused work. Besides, it requires to distinguish fasteners and general parts due to their fuzzy definitions. For example, it might be confused whether an object in press-fit component should be categorised as a fastener or a general part. Jin *et al.* [4, 5] proposed another matrix-based method in which the relationships of components were described by just a matrix.

The matrix-based methods work well on sequential disassembly tasks, but they fail if the product has interlocked components. In this case, breaking an assembly into subassemblies is required in disassembly plans. Based on an analysis of over 239 mechanical products by the authors' team, 23% of them contain interlocked components [3]. They cannot be correctly dealt with using conventional sequential disassembly planning methods.

This paper presents two solutions. One is to identify subassemblies by using 'separable pairs' to complements conventional sequential disassembly planning methods [15]. Another solution is based on a divide-and-conquer disassembly strategy which is essentially a hierarchical rather than a sequential planning method to avoid the interlocking problem.

Section 2 presents the mathematical representation of a product and explains interlocking problems. Section 3 presents the first solution, the use of 'separable pairs' to find subassemblies. Section 4 gives a new mathematical representation of assemblies to be adopted in a divide-and-conquer strategy. Section 5 explains the strategy and depicts the use of it for disassembly sequence planning. A case study is given in Sect. 6 to demonstrate the use of the two proposed solutions.

2 Contact and Relation Matrices: Fundamental Tools

Jin *et al.* [4, 5] demonstrated a method based on the space interference matrix to find removable components to generate feasible disassembly sequences. The key point of this method is to find removable components which have freedom in at least one direction. By taking away removable components iteratively, the product can be disassembled step-by-step in a sequential way.

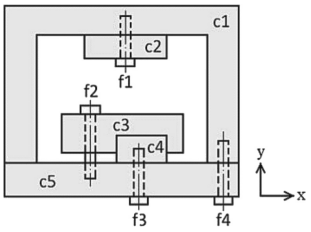

Fig. 1. An example product [10].

Figure 1 shows an example product which contains interlocked components. If the method is used to plan the disassembly sequences for this product, the first step is to remove f_3 and f_4. However, after this step, no component is removable for further disassembly process. Figure 2 gives the space interference matrix of the example product after the removal of f_3 and f_4. This is a typical interlocking structure. An assembly without removable part cannot be disassembled, hence, it requires to be divided into subassemblies to break the interlock.

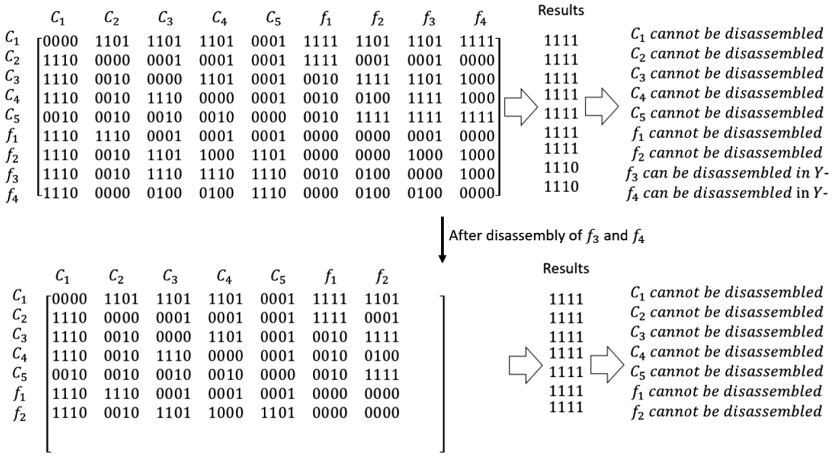

Fig. 2. Sequential disassembly method proposed by Jin *et al.* [4, 5].

The first proposed solution finds subassemblies by using contact matrix and relation matrix as fundamental tools, in which the contact conditions in an assembly in six directions (X+, X−, Y+, Y−, Z+, Z−) can be represented by the number 0 or 1. For the example product, only four directions (X+, X−, Y+, Y−) is required for demonstrations due to its 2D representation, as shown in Eq. 1.

In the matrix, C_n represents components in an assembly. $r_{ij.x+}$, $r_{ij.x-}$, $r_{ij.y+}$, and $r_{ij.y-}$ indicate the contact status of components in the corresponding columns and rows by using two states: 0 for no contact and 1 for contact. For example, Eq. 2 is the mathematical representation of the assembly in Fig. 1. The element in C_1 row and C_2

column, $r_{12.x+}r_{12.x-}r_{12.y+}r_{12.y-}$, is 0001 because C_2 contact C_1 in Y− direction. C_1 can be removed from C_2 in Y+ direction. Similarly, the element in C_2 row and C_1 column, $r_{21.x+}r_{21.x-}r_{21.y+}r_{21.y-}$, is 0010 because C_1 contact C_2 in Y+ direction, and hence C_2 can be removed from C_1 in Y− direction. It is worth noting that symmetry may not be observed in $r_{ab.x+}r_{ab.x-}r_{ab.y+}r_{ab.y-}$ and $r_{ba.x+}r_{ba.x-}r_{ba.y+}r_{ba.y-}$ due to the requirements of proper disassembly operations. For example, $r_{61.x+}r_{61.x-}r_{61.y+}r_{61.y-}$ is 1110 while $r_{16.x+}r_{16.x-}r_{16.y+}r_{16.y-}$ is 1111, because removing f_1 from C_1 is a proper operation but the reverse is not.

The general contact status of the components in an assembly is represented by the relation matrix (Fig. 3). In Sect. 3, the utility of the two matrices to allow a machine to understand subassemblies will be explained.

$$C = \begin{array}{c} \\ C_1 \\ \vdots \\ C_n \end{array} \begin{array}{c} \begin{array}{ccc} C_1 & \cdots & C_n \end{array} \\ \begin{bmatrix} r_{11.x+}r_{11.x-}r_{11.y+}r_{11.y-} & \cdots & r_{1n.x+}r_{1n.x-}r_{1n.y+}r_{1n.y-} \\ \vdots & \ddots & \vdots \\ r_{n1.x+}r_{n1.x-}r_{n1.y+}r_{n1.y-} & \cdots & r_{nn.x+}r_{nn.x-}r_{nn.y+}r_{nn.y-} \end{bmatrix} \end{array} \quad [15] \qquad (1)$$

$$C = \begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \end{array} \begin{array}{c} \begin{array}{ccccccccc} C_1 & C_2 & C_3 & C_4 & C_5 & f_1 & f_2 & f_3 & f_4 \end{array} \\ \begin{bmatrix} 0000 & 0001 & 0000 & 0000 & 0001 & 1111 & 0000 & 0000 & 1111 \\ 0010 & 0000 & 0000 & 0000 & 0000 & 1111 & 0000 & 0000 & 0000 \\ 0000 & 0000 & 0000 & 1101 & 0000 & 0000 & 1111 & 0000 & 0000 \\ 0000 & 0000 & 1110 & 0000 & 0001 & 0000 & 0000 & 1111 & 0000 \\ 0010 & 0000 & 0000 & 0010 & 0000 & 0000 & 1111 & 1111 & 1111 \\ 1110 & 1110 & 0000 & 0000 & 0000 & 0000 & 0000 & 0000 & 0000 \\ 0000 & 0000 & 1101 & 0000 & 1101 & 0000 & 0000 & 0000 & 0000 \\ 0000 & 0000 & 0000 & 1110 & 1110 & 0000 & 0000 & 0000 & 0000 \\ 1110 & 0000 & 0000 & 0000 & 1110 & 0000 & 0000 & 0000 & 0000 \end{bmatrix} \end{array} \quad [15] \qquad (2)$$

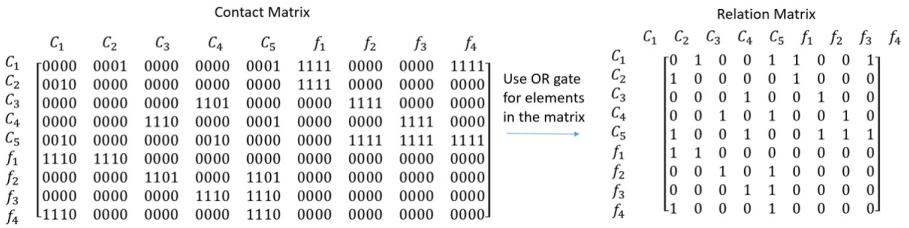

Fig. 3. Derivation of a relation matrix from a contact matrix [15].

3 Separability Check

3.1 Definition of Separability

The separability of an assembly is defined by whether it contains 'separable pairs', the pairs of contacting components that can be separated without effect on other contacting components, which means it can be divided into subassemblies. For example, in Fig. 4

(a), the assembly consists of three components: A1, B1 and C1, and two pairs of contacting components: A1–B1 and B1–C1. If the contact relationships can be represented as a line, and the component is represented by a node, then the model of the assembly in Fig. 4(a) can be abstracted to Fig. 4(b), which can also be represented by its relation matrix (R1), as shown in Fig. 4(c). Both pairs, A1–B1 and B1–C1, are separable, as the separation of either pair would not affect the other.

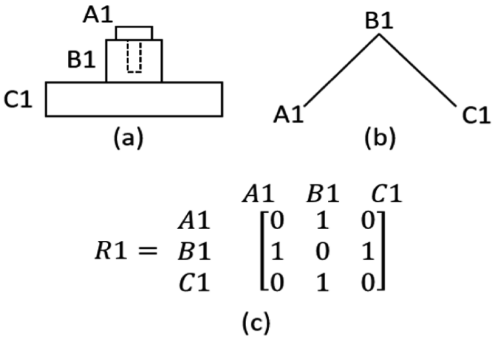

Fig. 4. An example of a product comprising separable pairs [15].

However, the model shown in Fig. 5 has no 'separable pair', as the separation of any pair (A2–B2, B2–C2 and A2–C2) could have effect on other pairs. For example, the separation of A2–B2 pair inevitably causes the detachment of A2 from C2. Thus, it indicates the assembly is interlocked so that it cannot be divided into subassemblies. Comparing Figs. 4(b) to 5(b), apparently, there is only one path between A1 and B1 (A1–B1) in Fig. 4(b), while there are two paths between A2 and B2 (A2–B2, and A2–C2–B2) in Fig. 5(b). If there is only one path between two components, their interaction is not coupled with other components. Hence, they are separable pairs. It is the sufficient condition of separable pairs that there exists only one path between the two components in a pair, as shown in the pairs A1–B1 and B1–C1 in Fig. 4(b).

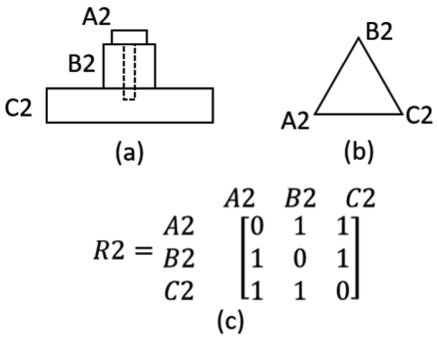

Fig. 5. An example of a product comprising inseparable pairs [15].

3.2 Separable Pairs Search Process

The node vectors are used to identify separable pairs as shown in Eq. 3, so that the nodes which connect to a given node can be calculated by multiplying the relation matrix R1 with its node vector (Eqs. 4 to 6).

$$\mathrm{A1} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, B1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \mathrm{C1} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} [15] \tag{3}$$

$$\mathrm{R1}.A1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = B1, [15] \tag{4}$$

$$\mathrm{R1}.B1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = A1 + C1, [15] \tag{5}$$

$$\mathrm{R1.C1} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = B1\,[15] \tag{6}$$

This method can be used to identify adjacent components pairs. Besides, the path between two nodes can be obtained by traversing of the graph. It is carried out mathematically by recursively multiplying the relation matrix by a node vector and its adjacent node vectors until a destination is reached. Figure 6 shows the procedure of searching for separable pairs using a relation matrix.

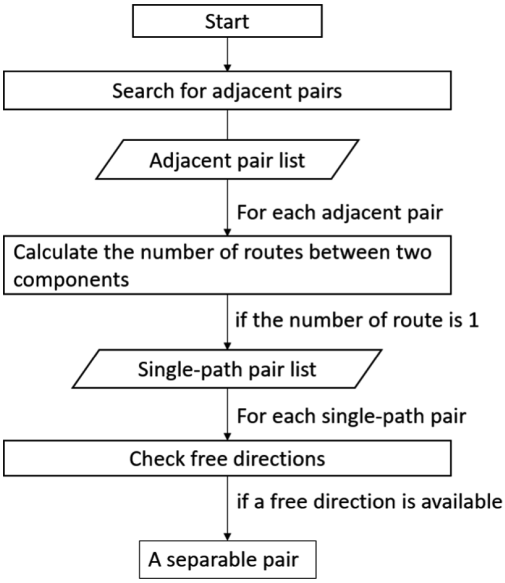

Fig. 6. Separable pairs search process [15].

The first step is to search for adjacent components pairs, two components in contact, using Eqs. 3 to 6.

The second step is to identify the pair in which there is only one route between the two components, as discussed in Sect. 3.1. We propose a recursive strategy by using the pseudo code in Algorithm 1.

Algorithm 1. Generate single-path pair list from adjacent pair list [15].

```
Main function:
Input:  adjacent pair list (APL)
Output: Single-path list (SPL)
1  For every pair {X, Y} Є APL
2   counter = 0
3    searchPath(X,Y) ;
4    If counter = 1
5      add {X, Y} to SPL;
6    End if
7  End for

searchPath(X,Y)
8  Label X as discovered
9  For every component k adjacent to X
10  If k is not labelled as discovered
11   If k = Y
12    counter++;
13    If counter >=2
14     break;
15    End if
16   Else
17    Recursively call searchPath(k,Y)
18   End if
19  End if
19 Return counter
20 End for
```

If all single-path pairs are obtained, their corresponding elements in the contact matrix should be checked. If the elements are not 1111 ($r_{ab.x+}\, r_{ab.x-} r_{ab.y+}\, r_{ab.y-} \neq 1111$ and $r_{ba.x+}\, r_{ba.x-} r_{ba.y+}\, r_{ba.y-} \neq 1111$), it indicates that one component has freedom in at least one direction in relation to the other, and thus the pair is separable. Details are explained using the discussed example in Fig. 1. If f_3 and f_4 are removed from the assembly, the node-line model of the assembly and its relation matrix are presented in Fig. 7 and Eq. 7. By using Eqs. 3 and 4, eight adjacent pairs can be identified: C1–C2, C1–C5, C1–f1, C2–f1, C3–C4, C3–f2, C4–C5, and C5–f2.

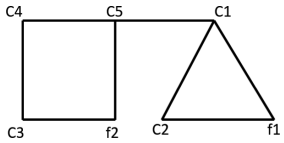

Fig. 7. Model of the assembly after the removal of f3 and f4 [15].

$$\mathrm{R} = \begin{array}{c} \\ C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ f_1 \\ f_2 \end{array} \begin{array}{c} \begin{array}{ccccccc} C_1 & C_2 & C_3 & C_4 & C_5 & f_1 & f_2 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix} \end{array} \quad [15] \tag{7}$$

Algorithm 1 gives the solution to obtain the number of routes between two components in a pair, which starts from the first pair C1–C2 in the pair list. It returns the result that the pair is not separable, as there is more than one route from C1 to C2 (C1→C2 and C1→f1→C2), as depicted in Fig. 8. For the next pair on the adjacent pair list, C1–C5, it returns only one route. Thus, this pair is added to single-path list. The algorithm continues to traverse all pairs in the adjacent pair list. Finally, only C1–C5 is single-path pair. By checking the contact matrix, C1 and C5 have freedom in 3 directions. Hence, the pair C1–C5 is a separable pair.

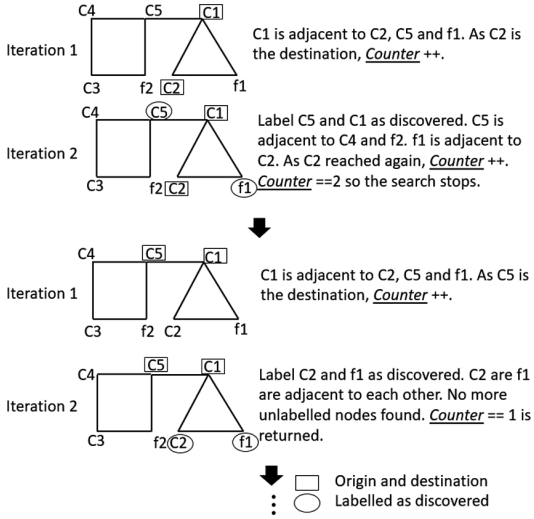

Fig. 8. An example of searching for single path pairs [15].

It indicates that the separation of C1 and C5 would result in two subassemblies: C1–C2–f1 and C3–C4–C5–f2. For each subassembly, f1 and f2 are removable respectively. The disassembly operations can carry on by applying conventional sequential disassembly planning methods.

4 Set and Closure: Mathematical Representation

The section first introduces new mathematical tools, set and unary operator, to represent a product and describe its space interference relations. The next part explains the definitions of closure which is used to detect subassemblies. They build the foundation of divide-and-conquer strategy.

4.1 Set

The Set of an Assembly

An assembly of a product can be represented by a set $\boldsymbol{A}$, which is a collection of the components of the products. Each component $\boldsymbol{E_i}(i = 1, 2, 3, \ldots n)$, the element of the set $\boldsymbol{A}$, cannot be disassembled into smaller components (Eq. 8).

$$\boldsymbol{A} = \{\boldsymbol{E}_1, \boldsymbol{E}_2, \ldots, \boldsymbol{E}_n\} \tag{8}$$

The Relation of Components in a Set

Space interferences of components in a set $\boldsymbol{A}$ can be described by using an interference matrix $\boldsymbol{I}$ [5]. For example, the 2-dimensional space relation of a product in Fig. 9 can be described in Eq. 9.

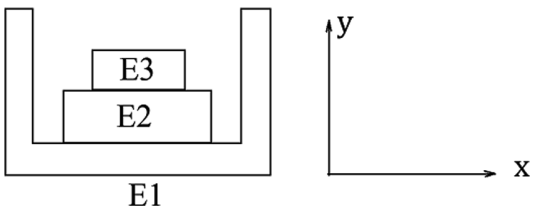

Fig. 9. An example to show the use of an interference matrix to describe the 2-dimensional space relations of a product.

$$\boldsymbol{I} = \begin{array}{c} \\ \boldsymbol{E}1 \\ \boldsymbol{E}2 \\ \boldsymbol{E}3 \end{array} \begin{array}{c} \begin{array}{ccc} \boldsymbol{E}1 & \boldsymbol{E}2 & \boldsymbol{E}3 \end{array} \\ \begin{bmatrix} 0000 & 1110 & 1110 \\ 1101 & 0000 & 0010 \\ 1101 & 0001 & 0000 \end{bmatrix} \end{array} \tag{9}$$

The Interference Operator of a Set

Set $\boldsymbol{A}$ and the interference matrix $\boldsymbol{I}$ describe the attributes of an assembly. For a set $\boldsymbol{A} = \{\boldsymbol{E}_1, \boldsymbol{E}_2, \ldots, \boldsymbol{E}_n\}$ and its interference matrix $\boldsymbol{I}$, a unary operator $\boldsymbol{U_d}(\boldsymbol{E}_i)$ $(\boldsymbol{E}_i \in \boldsymbol{A})$ can obtain the elements which have interference with $\boldsymbol{E}_i$ in direction $\boldsymbol{d}$ $(\boldsymbol{d} \in \{x+, x-, y+, y-, z+, z-\})$.

$$\forall \boldsymbol{E}_i \in \boldsymbol{A}, \boldsymbol{U_d}(\boldsymbol{E}_i) = \{\boldsymbol{E}_u, \boldsymbol{E}_v, \ldots, \boldsymbol{E}_w\} \subseteq \boldsymbol{A}, 1 \leq u, v, w \leq n \tag{10}$$

Referring to Eq. 9 as an example, an unary operator can be used to find the elements which have space interference with $\boldsymbol{E}_2$ along x+ direction. The result (Eq. 11) indicates that $\{\boldsymbol{E}_1\}$ has a space interference with $\boldsymbol{E}_2$ in x+ direction.

$$\boldsymbol{E}_{2x+} = \begin{matrix} \boldsymbol{E}_1 & \boldsymbol{E}_2 & \boldsymbol{E}_3 \\ [1 & 0 & 0] \end{matrix} \tag{11}$$

4.2 Subassembly and Closure

A subassembly is a collection of components that can be disassembled along a certain direction $\boldsymbol{d}$ without affecting other components. For a given direction $\boldsymbol{d}$, elements in a subassembly have interferences with one another, but they have no interferences with elements outside the subassembly. In other words, if an element has a space interference with another element, they both belong to the same subassembly (Eq. 12).

$$\forall \boldsymbol{E}_i \in \boldsymbol{S}, \boldsymbol{U_d}(\boldsymbol{E}_i) = \{\boldsymbol{E}_u, \boldsymbol{E}_v, \ldots, \boldsymbol{E}_w\} \subseteq \boldsymbol{S} \tag{12}$$

Where $\boldsymbol{S} = \{\boldsymbol{E}_u, \boldsymbol{E}_v, \ldots, \boldsymbol{E}_w\} \subseteq \boldsymbol{A}, 1 \leq u, v, w \leq n$. The subset $\boldsymbol{S}$ is defined to be *closed* under the operator $\boldsymbol{U_d}$ in $\boldsymbol{d}$ direction, if the results of the operation on any element in $\boldsymbol{S}$ also belongs to $\boldsymbol{S}$. Such feature is also called a *closure*. A closure indicates that the elements in the subset do not have interference with components outside the subset, which is a sign that the subset builds a subassembly that can be removed in direction $\boldsymbol{d}$.

Also, the subset $\boldsymbol{S}$ in a set $\boldsymbol{A}$ is equivalent to an element in $\boldsymbol{A}$. The space interference operator can be subjected to the subset $\boldsymbol{S}$, too. For example, $\boldsymbol{S} \subseteq \boldsymbol{A}, \boldsymbol{A} = \{\boldsymbol{E}_1, \boldsymbol{E}_2, \boldsymbol{E}_3, \boldsymbol{E}_4, \ldots, \boldsymbol{E}_n\}$, $\boldsymbol{S} = \{\boldsymbol{E}_1, \boldsymbol{E}_2, \boldsymbol{E}_3\}$, then,

$$\boldsymbol{A} = \{\mathbf{S}, \boldsymbol{E}_4, \ldots, \boldsymbol{E}_n\} \tag{13}$$

The divide-and-conquer strategy to be presented in Sect. 5 is based on the detection of subassemblies through the detection of closure in a set.

5 Divide-and-Conquer Disassembly Strategy

The strategy is to divide a product into subassemblies iteratively until all subassemblies become individual components which cannot be divided anymore. This method allows all subassemblies to be identified automatically during a disassembly planning, and hence the interlocking problem does not exist.

5.1 Subassembly Search Process

The search for subassemblies can use the algorithm given in Fig. 10. For example, referring to Fig. 11, after the disassembly of f3 and f4, the assembly is shown as in Eq. 14 and the space interference relation is given in Eq. 15.

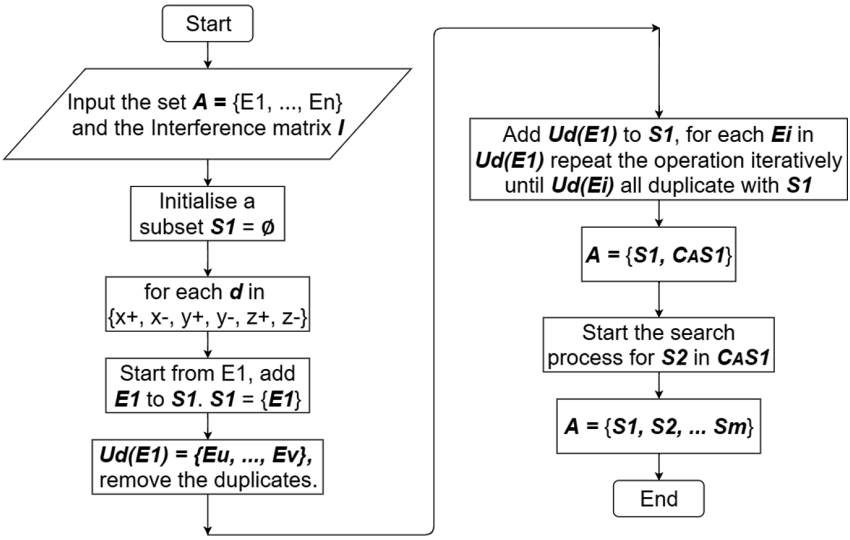

Fig. 10. Subassembly search process

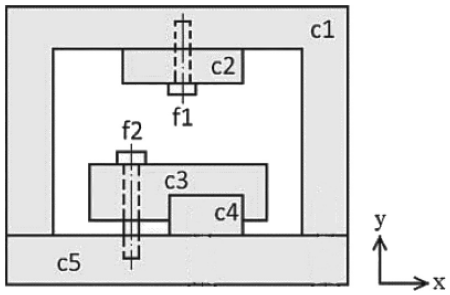

Fig. 11. An assembly example after the disassembly of f3 and f4 (refer to Fig. 1).

$$\boldsymbol{A} = \{\boldsymbol{C}_1, \boldsymbol{C}_2, \boldsymbol{C}_3, \boldsymbol{C}_4, \boldsymbol{C}_5, \boldsymbol{f}_1, \boldsymbol{f}_2\} \tag{14}$$

$$\boldsymbol{I} = \begin{array}{c} \\ \boldsymbol{C}_1 \\ \boldsymbol{C}_2 \\ \boldsymbol{C}_3 \\ \boldsymbol{C}_4 \\ \boldsymbol{C}_5 \\ \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \end{array} \begin{array}{c} \begin{array}{ccccccc} \boldsymbol{C}_1 & \boldsymbol{C}_2 & \boldsymbol{C}_3 & \boldsymbol{C}_4 & \boldsymbol{C}_5 & \boldsymbol{f}_1 & \boldsymbol{f}_2 \end{array} \\ \begin{bmatrix} 0000 & 1101 & 1101 & 1101 & 0001 & 1111 & 1101 \\ 1110 & 0000 & 0001 & 0001 & 0001 & 1111 & 0001 \\ 1110 & 0010 & 0000 & 1101 & 0001 & 0010 & 1111 \\ 1110 & 0000 & 1110 & 0000 & 0001 & 0010 & 0100 \\ 0010 & 0010 & 0010 & 0010 & 0000 & 0010 & 1111 \\ 1110 & 1110 & 0001 & 0001 & 0001 & 0000 & 0000 \\ 1110 & 0010 & 1101 & 1000 & 1101 & 0000 & 0000 \end{bmatrix} \end{array} \tag{15}$$

Identification of a subassembly starts with initialising a target subset $\boldsymbol{S}_1$ to be an empty set ϕ. The subassembly search process adds the first element $\boldsymbol{C}_1$ to the subset $\boldsymbol{S}_1$, and hence $\boldsymbol{S}_1 = \{\boldsymbol{C}_1\}$. Subject the interference operator to $\boldsymbol{C}_1$ along y+ direction to obtain:

$$\boldsymbol{U}_{y+}(\boldsymbol{C}_1) = \{\boldsymbol{f}_1\} \tag{16}$$

The result $\{\boldsymbol{f}_1\}$ does not duplicate with the elements in $\boldsymbol{S}_1$, and thus $\boldsymbol{f}_1$ is added to $\boldsymbol{S}_1$

$$\boldsymbol{S}_1 = \{\boldsymbol{C}_1, \boldsymbol{f}_1\} \tag{17}$$

Subject the interference operator to the new element $\boldsymbol{f}_1$ again and yield:

$$\boldsymbol{U}_{y+}(\boldsymbol{f}_1) = \{\boldsymbol{C}_1, \boldsymbol{C}_2\} \tag{18}$$

The new element $\boldsymbol{C}_2$ is added to $\boldsymbol{S}_1$ as in Eq. 19.

$$\boldsymbol{S}_1 = \{\boldsymbol{C}_1, \boldsymbol{f}_1, \boldsymbol{C}_2\} \tag{19}$$

Subject the operator to $\boldsymbol{C}_2$ and yield

$$\boldsymbol{U}_{y+}(\boldsymbol{C}_2) = \{\boldsymbol{C}_1, \boldsymbol{f}_1\} \subseteq \boldsymbol{S}_1 \tag{20}$$

As $\boldsymbol{C}_1$and$\boldsymbol{f}_1$ are already in $\boldsymbol{S}_1$, the result indicates that the current subset $\boldsymbol{S}_1$ is closed.

$$\boldsymbol{S}_1 \subseteq \boldsymbol{A}, \forall \boldsymbol{E}_i \in \boldsymbol{S}_1, \boldsymbol{U}_{y+}(\boldsymbol{E}_i) \subseteq \boldsymbol{S} \tag{21}$$

Hence, the set $\boldsymbol{A}$ can be re-written and contain two elements: subset $\boldsymbol{S}_1$ and its complement $\boldsymbol{C}_A\boldsymbol{S}_1$.

$$\boldsymbol{A} = \{\boldsymbol{S}_1, \boldsymbol{C}_A\boldsymbol{S}_1\} \tag{22}$$

Where $\boldsymbol{C}_A\boldsymbol{S}_1 = \{\boldsymbol{C}_3, \boldsymbol{C}_4, \boldsymbol{C}_5, \boldsymbol{f}_2\}$. The next step is to identify subassemblies in $\boldsymbol{C}_A\boldsymbol{S}_1$ starting from the first element $\boldsymbol{C}_3$. Because the previous subset $\boldsymbol{S}_1$ can be disassembled along y+ direction, the new search direction is y−.

Similarly, a subset $\boldsymbol{S}_2$ is initialised to be empty: $\boldsymbol{S}_2 = \phi$, and the first element $\boldsymbol{C}_3$ is added to $\boldsymbol{S}_2$. Hence, $\boldsymbol{S}_2 = \{\boldsymbol{C}_3\}$.

$$\boldsymbol{U}_{y-}(\boldsymbol{C}_3) = \{\boldsymbol{C}_4, \boldsymbol{C}_5, \boldsymbol{f}_2\} \tag{23}$$

$\boldsymbol{C}_4, \boldsymbol{C}_5, \boldsymbol{f}_2$ do not exist in $\boldsymbol{S}_2$ and are added.

$$\boldsymbol{S}_2 = \{\boldsymbol{C}_3, \boldsymbol{C}_4, \boldsymbol{C}_5, \boldsymbol{f}_2\} = \boldsymbol{C}_A\boldsymbol{S}_1 \tag{24}$$

The subset $\boldsymbol{S}_2$ is equal to $\boldsymbol{C}_A\boldsymbol{S}_1$, which means the second subset has been found. The space interference of the rest of the elements are checked below:

$$\boldsymbol{U}_{y-}(\boldsymbol{C}_4) = \{\boldsymbol{C}_5\} \subseteq \boldsymbol{S}_2 \tag{25}$$

$$\boldsymbol{U}_{y-}(\boldsymbol{C}_5) = \{\boldsymbol{f}_2\} \subseteq \boldsymbol{S}_2 \tag{26}$$

$$\boldsymbol{U}_{y-}(\boldsymbol{f}_2) = \{\boldsymbol{C}_3, \boldsymbol{C}_5\} \subseteq \boldsymbol{S}_2 \tag{27}$$

Hence, the subset $\boldsymbol{S}_2$ is closed under the interference operator. All the elements have been traversed, which means

$$\boldsymbol{A} = \{\boldsymbol{S}_1, \boldsymbol{S}_2\} \tag{28}$$

5.2 Procedure of Divide-and-Conquer Strategy

This strategy analyses elements in set $\boldsymbol{A}$ in a hierarchical way following a procedure given in Fig. 13. The first operation is to divide $\boldsymbol{A}$ into subassemblies (Eq. 29)

$$\boldsymbol{A} = \{\boldsymbol{S}_1, \boldsymbol{S}_2, \boldsymbol{S}_3, \ldots, \boldsymbol{S}_u\} \tag{29}$$

For each $\boldsymbol{S}_i$ in $\boldsymbol{A} = \{\boldsymbol{S}_1, \boldsymbol{S}_2, \boldsymbol{S}_3, \ldots, \boldsymbol{S}_u\}$, divide the subset $\boldsymbol{S}_i$ into several subassemblies again, yields:

$$\boldsymbol{S}_i = \{\boldsymbol{S}_{i1}, \boldsymbol{S}_{i2}, \ldots, \boldsymbol{S}_{iw}\} \tag{30}$$

Repeat this operation iteratively, until all the subassemblies are components which cannot be divided anymore. A diagram showing the hierarchy of the disassembly sequences is shown below.

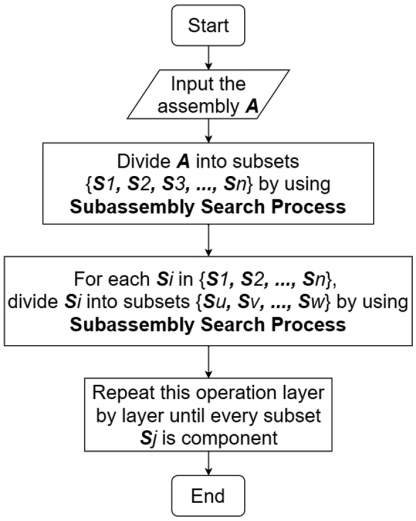

Fig. 12. Procedure of divide and conquer disassembly strategy.

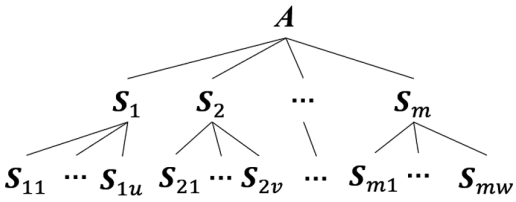

Fig. 13. Hierarchy of disassembly sequences of a product.

For example, if there is a set $\boldsymbol{A}$ (Eq. 31) and its undisclosed hierarchy is in Fig. 14, the first step of using divide-and-conquer disassembly strategy is to divide $\boldsymbol{A}$ into subsets, yielding Eqs. 32 and 33.

$$\boldsymbol{A} = \{\boldsymbol{E}_1, \boldsymbol{E}_2, \ldots, \boldsymbol{E}_{10}\} \tag{31}$$

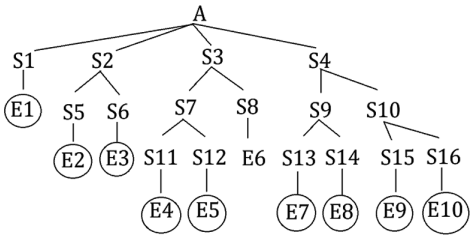

Fig. 14. The hierarchy of the disassembly sequences.

$$\boldsymbol{A} = \{\boldsymbol{S}_1, \boldsymbol{S}_2, \boldsymbol{S}_3, \boldsymbol{S}_4\} \tag{32}$$

$$\begin{cases} \boldsymbol{S}_1 = \{\boldsymbol{E}_1\} \\ \boldsymbol{S}_2 = \{\boldsymbol{E}_2, \boldsymbol{E}_3\} \\ \boldsymbol{S}_3 = \{\boldsymbol{E}_4, \boldsymbol{E}_5, \boldsymbol{E}_6\} \\ \boldsymbol{S}_4 = \{\boldsymbol{E}_7, \boldsymbol{E}_8, \boldsymbol{E}_9, \boldsymbol{E}_{10}\} \end{cases} \tag{33}$$

After generating the first layer of disassembly sequences as in Eq. 33, each subset is checked. $\boldsymbol{S}_1$ has already been an element so the operation stops. $\boldsymbol{S}_2$, $\boldsymbol{S}_3$, and $\boldsymbol{S}_4$ can be further divided.

$\boldsymbol{S}_2$ is divided into two subsets:

$$\boldsymbol{S}_2 = \{\boldsymbol{S}_5, \boldsymbol{S}_6\}, \begin{cases} \boldsymbol{S}_5 = \{\boldsymbol{E}_2\} \\ \boldsymbol{S}_6 = \{\boldsymbol{E}_3\} \end{cases} \tag{34}$$

$\boldsymbol{S}_5$ and $\boldsymbol{S}_6$ cannot longer be divided.

$\boldsymbol{S}_3$ and $\boldsymbol{S}_4$ are divided into subsets below:

$$\boldsymbol{S}_3 = \{\boldsymbol{S}_7, \boldsymbol{S}_8\}, \begin{cases} \boldsymbol{S}_7 = \{\boldsymbol{E}_4, \boldsymbol{E}_5\} \\ \boldsymbol{S}_8 = \{\boldsymbol{E}_6\} \end{cases} \tag{35}$$

$$\boldsymbol{S}_4 = \{\boldsymbol{S}_9, \boldsymbol{S}_{10}\}, \begin{cases} \boldsymbol{S}_9 = \{\boldsymbol{E}_7, \boldsymbol{E}_8\} \\ \boldsymbol{S}_{10} = \{\boldsymbol{E}_9, \boldsymbol{E}_{10}\} \end{cases} \tag{36}$$

$\boldsymbol{S}_7, \boldsymbol{S}_9$ and $\boldsymbol{S}_{10}$ can be divided below:

$$\begin{aligned} \boldsymbol{S}_7 &= \{\boldsymbol{S}_{11}, \boldsymbol{S}_{12}\}, \begin{cases} \boldsymbol{S}_{11} = \{\boldsymbol{E}_4\} \\ \boldsymbol{S}_{12} = \{\boldsymbol{E}_5\} \end{cases} \\ \boldsymbol{S}_9 &= \{\boldsymbol{S}_{13}, \boldsymbol{S}_{14}\}, \begin{cases} \boldsymbol{S}_{13} = \{\boldsymbol{E}_7\} \\ \boldsymbol{S}_{14} = \{\boldsymbol{E}_8\} \end{cases} \\ \boldsymbol{S}_{10} &= \{\boldsymbol{S}_{15}, \boldsymbol{S}_{16}\}, \begin{cases} \boldsymbol{S}_{15} = \{\boldsymbol{E}_9\} \\ \boldsymbol{S}_{16} = \{\boldsymbol{E}_{10}\} \end{cases} \end{aligned} \tag{37}$$

Then, all the subsets contain only an element and the disassembly planning is finished.

6 Case Study

This section demonstrates a case study of the disassembly of a piston used in a 4-stroke engine, as shown in Fig. 15, by using both 'separable pairs' method and divide-and-conquer strategy. B–C1–C2–D is an interlocked structure. Disassembly would require building two subassemblies: B–C1 and C2–D, so that C1 can be removed from B, and C2 can be taken away from D.

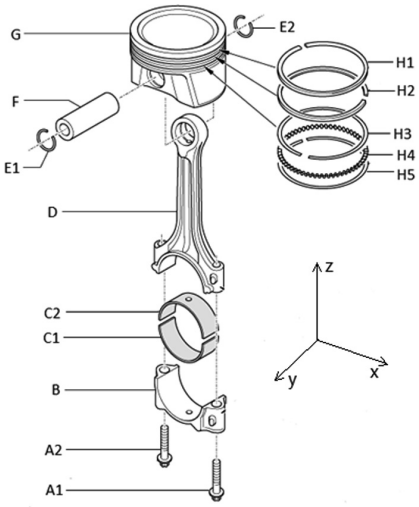

Fig. 15. Parts in a piston [15].

6.1 'Separable Pairs' Method

The sequential disassembly plan as shown in Table 1 is obtained by using the conventional sequential disassembly method [4, 5] based on the space interference matrix (Appendix). As shown in Table 1, there is no removable part identified at iteration 7, and parts B, C1–2 and D are interlocked. The methods presented in Sect. 3 is used to find a separable pair to divide the interlocking structure into subassemblies for normal disassembly operations.

Table 1. Sequential disassembly plan generated using the method by Jin *et al.* [15].

Iteration	Removable parts	Remaining parts
1	A1-111110	B, C1–2, D, F, G, H2–5
	A2-111110	
	E1-110111	
	E2-111011	
	H1-111101	
2	F -110011	B, C1–2, D, G, H3–5
	H2 - 111101	
3	H3 - 111101	B, C1–2, D, G, H4–5
4	H4 - 111101	B, C1–2, D, G, H5
5	H5 - 111101	B, C1–2, D, G
6	G - 111101	B, C1–2, D
7	None	B, C1–2, D

Equations 38 and 39 give the contact matrix and the relation matrix of the interlocked structure B–C1–C2–D. By using the method presented in Eqs. 3 to 6, three adjacent components pairs are identified: B–C1, B–D and C2–D. Then, the number of routes between two components in each adjacent components pair is calculated based on Algorithm 1. The results show that all adjacent components pairs are single-path pairs. However, by checking the contact matrix, only B–D pair is a separable pair as C_{12} and C_{43} in Eq. 38 are 111111, indicating that either B–C1 or C2–D has no freedom to separate. The separation of B and D forms two subassemblies, B–C1 and C2–D. Thus, the conventional sequential disassembly methods can carry on.

$$C = \begin{array}{c} \\ B \\ C1 \\ C2 \\ D \end{array} \begin{array}{c} \begin{array}{cccc} B & C1 & C2 & D \end{array} \\ \begin{bmatrix} 000000 & 111111 & 000000 & 000010 \\ 111101 & 000000 & 000000 & 000000 \\ 000000 & 000000 & 000000 & 111110 \\ 000001 & 000000 & 111111 & 000000 \end{bmatrix} \end{array} \tag{38}$$

$$R = \begin{array}{c} \\ B \\ C1 \\ C2 \\ D \end{array} \begin{array}{c} \begin{array}{cccc} B & C1 & C2 & D \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \end{array} [15] \tag{39}$$

6.2 Divide-and-Conquer Strategy

By using the divide-and-conquer disassembly strategy proposed, the hierarchy of the piston example can be generated as in Fig. 16. The disassembly sequences are generated, given in Tables 2, 3 and 4 by using the space interference matrix (Appendix).

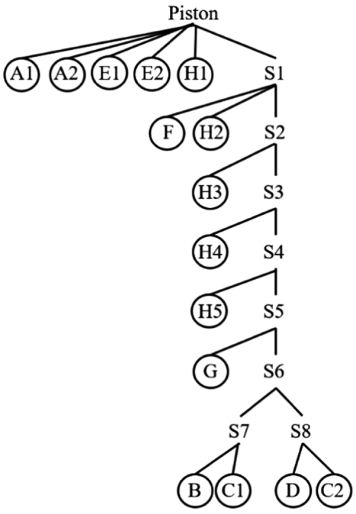

Fig. 16. Hierarchy analysis of piston example.

Table 2. Divide-and-conquer disassembly sequences of the example product.

Iteration	Subassemblies	Minimum units	Remaining subassemblies
1	{A1}, {A2}, {E1}, {E2}, {H1}, {B, C1–2, D, F, G, H2–5}	{A1}, {A2}, {E1}, {E2}, {H1}	{B, C1–2, D, F, G, H2–5}
2	{F}, {H2}, {B, C1–2, D, G, H3–5}	{F}, {H2}	{B, C1–2, D, G, H3–5}
3	{H3}, {B, C1–2, D, G, H4–5}	{H3}	{B, C1–2, D, G, H4–5}
4	{H4}, {B, C1–2, D, G, H5}	{H4}	{B, C1–2, D, G, H5}
5	{H5}, {B, C1–2, D, G}	{H5}	{B, C1–2, D, G}
6	{G}, {B, C1–2, D}	{G}	{B, C1–2, D}
7	{B, C1}, {D, C2}	ϕ	{B, C1}, {D, C2}

When it comes to step 7, the remaining subassemblies is {B, C1} and {D, C2}. Similarly, the disassembly sequences of these two subassemblies generated by divide-and-conquer strategy are shown in Tables 3 and 4. The interlocking problem does not exist using this method.

Table 3. Disassembly sequences of the subassembly {B, C1}.

Iteration	Subassemblies	Minimum units	Remaining subassemblies
8	{B}, {C1}	{B}, {C1}	ϕ

Table 4. Disassembly sequences of the subassembly {D, C2}.

Iteration	Subassemblies	Minimum units	Remaining subassemblies
8	{D}, {C2}	{D}, {C2}	ϕ

7 Conclusion

Machine understanding of the structure of an assembly in three-dimensional space is an enabler of autonomous disassembly planning. Conventionally, due to the complexity of spatial information, many disassembly planning methods are not suitable for those containing interlocked components.

This paper presents two methods to deal with the interlocking problem. They are designed to work for all subassemblies containing interlocking components and their effectiveness was demonstrated with a case study. One work is based on the search for separable pairs, which can be used to identify subassemblies during a sequential disassembly planning. The other solution is based on a divide-and-conquer strategy to replace the conventional sequential disassembly methods. It plans disassembly sequence through the detection of subassemblies instead of removable individual components so that interlocked components can be detected automatically in each interaction, and thus interlocking problems can be avoided.

Acknowledgement. This research was supported by the EPSRC (Grant No. EP/N018524/1) and the National Science Foundation of China (Grant No. 51775399).

Appendix

Space interference matrix 15:

	A1	A2	B	C1	C2	D	E1	E2	F	G	H1	H2	H3	H4	H5
A1	000000	010000	111110	010000	010000	111110	000000	000000	000000	000010	000010	000010	000010	000010	000010
A2	100000	000000	111110	100000	100000	111110	000000	000000	000000	000010	000010	000010	000010	000010	000010
B	111111	111111	000000	111111	000010	000010	000000	000000	000010	000010	000010	000010	000010	000010	000010
C1	100000	010000	111101	000000	000010	000010	000000	000000	000010	000010	000010	000010	000010	000010	000010
C2	100000	010000	000001	000001	000000	111110	000000	000000	000010	000010	000010	000010	000010	000010	000010
D	111111	111111	000001	000001	111111	000000	001000	000100	110011	111110	000010	000010	000010	000010	000010
E1	000000	000000	000000	000000	000000	000100	000000	000100	000100	110111	000000	000000	000000	000000	000000
E2	000000	000000	000001	000001	000001	110011	001000	000000	001000	111011	000000	000000	000000	000000	000000
F	000001	000001	000001	000001	000001	111101	111111	111111	111111	000000	111111	111111	111111	111111	111111
G	000001	000001	000001	000001	000001	111101	111111	111111	111111	000000	111101	000001	000001	000001	000001
H1	000001	000001	000001	000001	000001	111101	111111	111111	111111	111101	000010	000000	000001	000001	000001
H2	000001	000001	000001	000001	000001	111101	111111	111111	111111	111101	000010	000000	000001	000001	000001
H3	000001	000001	000001	000001	000001	111101	111111	111111	111111	111101	000010	000000	000000	000001	000001
H4	000001	000001	000001	000001	000001	111101	111111	111111	111111	111101	000010	000000	000000	000000	000001
H5	000001	000001	000001	000001	000001	111101	111111	111111	111111	111101	000010	000000	000000	000000	000000

References

1. Barwood M, Li J, Pringle T, Rahimifard S (2015) Utilisation of reconfigurable recycling systems for improved material recovery from e-waste. Procedia CIRP 29:746–751. https://doi.org/10.1016/j.procir.2015.02.071
2. Gil P, Pomares J, Puente SVT, Diaz C, Candelas F, Torres F (2007) Flexible multi-sensorial system for automatic disassembly using cooperative robots. Int J Comput Integr Manuf 20 (8):757–772. https://doi.org/10.1080/09511920601143169
3. Ji C, Pham DT, Su S, Huang J, Wang Y (2017) AUTOREMAN – D.1.1 - List of generic disassembly task categories. Technical report, Autonomous Remanufacturing Laboratory, The University of Birmingham
4. Jin G, Li W, Wang S, Gao S (2015) A systematic selective disassembly approach for waste electrical and electronic equipment with case study on liquid crystal display televisions. Proc Inst Mech Eng Part B J Eng Manuf 231:2261–2278. https://doi.org/10.1177/0954405415575476
5. Jin GQ, Li WD, Xia K (2013) Disassembly matrix for liquid crystal displays televisions. Procedia CIRP 11:357–362. https://doi.org/10.1016/j.procir.2013.07.015
6. Johnson MR, McCarthy IP (2014) Product recovery decisions within the context of extended producer responsibility. J Eng Tech Manag 34:9–28. https://doi.org/10.1016/j.jengtecman.2013.11.002
7. Kopacek P, Kronreif G (1996) Semi-automated robotic disassembling of personal computers. In: EFTA 1996 - IEEE conference on emerging technologies and factory automation, vol 2, pp 567–572. IEEE. https://doi.org/10.1109/ETFA.1996.573938
8. Li JR, Khoo LP, Tor SB (2002) A novel representation scheme for disassembly sequence planning. Int J Adv Manuf Technol 20(8):621–630. https://doi.org/10.1007/s001700200199
9. Smith S, Chen W-H (2009) Rule-based recursive selective disassembly sequence planning for green design. Springer, London, pp 291–302. https://doi.org/10.1007/978-1-84882-762-2_27
10. Smith S, Hung P-Y (2015) A novel selective parallel disassembly planning method for green design. J Eng Des 26(10–12):283–301. https://doi.org/10.1080/09544828.2015.1045841
11. Smith S, Smith G, Chen W-H (2012) Disassembly sequence structure graphs: an optimal approach for multiple-target selective disassembly sequence planning. Adv Eng Inform 26 (2):306–316. https://doi.org/10.1016/j.aei.2011.11.003
12. Tao F, Bi L, Zuo Y, Nee AYC (2017) Partial/parallel disassembly sequence planning for complex products. J Manuf Sci Eng 140(1):011016. https://doi.org/10.1115/1.4037608
13. Torres F, Puente ST, Aracil R (2003) Disassembly planning based on precedence relations among assemblies. Int J Adv Manuf Technol 21(5):317–327. https://doi.org/10.1007/s001700300037
14. Vongbunyong S, Chen WH (2015) Disassembly automation. Springer, Cham. https://doi.org/10.1007/978-3-319-15183-0
15. Wang Y, Lan F, Pham D, Liu J, Huang, J, Ji C, Su S, Xu W, Liu Q, Zhou Z (2018) Automatic detection of subassemblies for disassembly sequence planning. In: Proceedings of the 15th international conference on informatics in control, automation and robotics - volume 1: ICINCO, pp 94–100. https://doi.org/10.5220/0006906601040110. ISBN 978-989-758-321-6
16. Zhao S, Li Y (2010) Disassembly sequence decision making for products recycling and remanufacturing systems. In: 2010 International symposium on computational intelligence and design, pp 44–48). IEEE. https://doi.org/10.1109/ISCID.2010.19