

Interlocking problems in disassembly sequence planning

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

DOI:

[10.1080/00207543.2020.1770892](https://doi.org/10.1080/00207543.2020.1770892)

License:

None: All rights reserved

Document Version

Peer reviewed version

Citation for published version (Harvard):

Wang, Y, Lan, F, Liu, J, Huang, J, Su, S, Ji, C, Pham, D, Xu, W, Liu, Q & Zhou, Z 2020, 'Interlocking problems in disassembly sequence planning', *International Journal of Production Research*.
<https://doi.org/10.1080/00207543.2020.1770892>

[Link to publication on Research at Birmingham portal](#)

Publisher Rights Statement:

This is an Accepted Manuscript of an article published by Taylor & Francis in International Journal of Production Research on 11/06/2020, available online: <http://www.tandfonline.com/10.1080/00207543.2020.1770892>

General rights

Unless a licence is specified above, all rights (including copyright and moral rights) in this document are retained by the authors and/or the copyright holders. The express permission of the copyright holder must be obtained for any use of this material other than for purposes permitted by law.

- Users may freely distribute the URL that is used to identify this publication.
- Users may download and/or print one copy of the publication from the University of Birmingham research portal for the purpose of private study or non-commercial research.
- User may use extracts from the document in line with the concept of 'fair dealing' under the Copyright, Designs and Patents Act 1988 (?)
- Users may not further distribute the material nor use it for the purposes of commercial gain.

Where a licence is displayed above, please note the terms and conditions of the licence govern your use of this document.

When citing, please reference the published version.

Take down policy

While the University of Birmingham exercises care and attention in making items available there are rare occasions when an item has been uploaded in error or has been deemed to be commercially or otherwise sensitive.

If you believe that this is the case for this document, please contact UBIRA@lists.bham.ac.uk providing details and we will remove access to the work immediately and investigate.

INTERNATIONAL JOURNAL OF

Production Research

Official Journal of the International Foundation for Production Research

Editor-in-Chief: Alexandre Dolgui



Taylor & Francis
Taylor & Francis Group

Interlocking problems in disassembly sequence planning

Journal:	International Journal of Production Research
Manuscript ID	TPRS-2019-IJPR-2165.R2
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	01-May-2020
Complete List of Authors:	Wang, Yongjing; University of Birmingham, Department of Mechanical Engineering Lan, Feiying; University of Birmingham, Department of Mechanical Engineering Liu, Jiayi; Wuhan University of Technology, Schoole of Information Engineering Huang, Jun; University of Birmingham, Department of Mechanical Engineering Su, Shizhong; University of Birmingham, Department of Mechanical Engineering Ji, Chunqian; University of Birmingham Pham, Duc; Birmingham University, Mechanical Engineering Xu, Wenjun; Wuhan University of Technology Liu, Quan; Wuhan University of Technology, School of Automation Zhou, ZuDe; Wuhan University of Technology, Hubei Digital Manufacturing Key Lab
Keywords:	DISASSEMBLY PLANNING, DISASSEMBLY
Keywords (user):	Remanufacturing, Robotic disassembly



Interlocking problems in disassembly sequence planning

Yongjing Wang^{1*}, Feiying Lan¹, Jiayi Liu^{1,2}, Jun Huang¹, Shizhong Su¹, Chunqian Ji¹, Duc Truong Pham¹,

Wenjun Xu^{2,3}, Quan Liu^{2,4} and Zude Zhou^{4,5}

¹ Department of Mechanical Engineering, University of Birmingham, Birmingham, United Kingdom

² School of Information Engineering, Wuhan University of Technology, Wuhan 430070, China

³ Hubei Key Laboratory of Broadband Wireless Communication and Sensor Networks, Wuhan University of Technology, Wuhan 430070, China

⁴ Key Laboratory of Fibre Optic Sensing Technology and Information Processing (Ministry of Education), Wuhan University of Technology, Wuhan 430070, China

⁵ School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan 430070, China

*email: y.wang@bham.ac.uk

Abstract

Remanufacturing is the rebuilding of a product to specifications of the original manufactured product using a combination of directly reused, repaired and new parts. Disassembly, the first and arguably most important process in remanufacturing, tends to be labour-intensive due to complexities in the conditions of end-of-life products returned for remanufacture. Robotic disassembly is an attractive alternative to manual disassembly but robotic systems cannot plan disassembly sequences automatically and manual planning is still necessary. Planning requires machines to interpret physical space using a suitable representation to reflect physical contacts and constraints as well as rules for deciding the sequences of disassembly operations. This paper proposes a representation to describe physical contacts and constraints, and a new approach allowing machines to plan disassembly using the representation. The approach involves employing an assembly matrix and simple logic gates to generate a contact matrix, a space interference matrix and a relation matrix. Rules and algorithms are discussed to explain the calculation of disassembly sequences through manipulating the three matrices. A key benefit is that the proposed method can deal with interlocked mechanical structures which cannot be handled using conventional methods. The proposed method is also flexible and is suitable for either selective or complete disassembly.

Keywords: remanufacturing; disassembly planning; dismantling; robotic disassembly

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

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’* [1]. It is an important component of the circular economy, saving raw materials and enabling substantial CO2 emission reductions and energy savings in several industry sectors [2]. It is a development and investment focus in the EU [3] and China [4], [5]. An important feature distinguishing remanufacturing from conventional manufacturing is disassembly. Due to the variability in the condition of the returned products, disassembly tends to be manually carried out. It is very labour intensive, given the complexity of the operations involved.

Several topics on disassembly have been addressed in the contexts of sustainable manufacturing [6] and Industry 4.0 [7]. They include disassembly line balancing [8]–[13], human-robot collaboration [14], [15], and the use of augmented reality in disassembly [16]. Romero-Silva and Marsillac have summarised recent research and development trends [17].

Pioneering developments in automated disassembly systems started in the mid-1990s with the robotic disassembly of a PC [18], followed by several successful attempts at dismantling electrical appliances and automotive components [19]–[21]. Those early systems were mostly product-orientated and based on pre-programmed sequences. A key advance from ‘automated’ disassembly to ‘autonomous’ disassembly would be that machines plan disassembly sequences using the structure of the product rather than following a pre-programmed sequence. A fundamental requirement of autonomous disassembly systems is that machines understand spatial information. However, few methods can be used to translate complete spatial information about a structure or assembly to a mathematical language that machines can interpret and use for motion planning.

Various techniques have been reported for representing the spatial information of components in a product. Some of those techniques are covered in Lambert’s survey of the literature on disassembly sequencing up to 2003 [22]. Bourjault formulated the precedence

relationships between components as a tree structure, namely a liaison graph [23]. De Fazio and Whitney extended Bourjault's work and reduced the number of searches by checking the connectivity states of products [24]. Homem de Mello and Sanderson introduced the AND/OR graph as a tool for representing a disassembly sequence. The AND/OR graph uses fewer nodes and thus reduces the complexity of searches [25], [26]. Kanehara *et al.* adopted Petri nets to generate assembly sequences based on AND/OR graphs [27]. There was also other work on disassembly sequencing focused on the economic return of a disassembly task [28]–[31]. Torres *et al.* proposed a new graph-based representation and emphasised the importance of using an appropriate sequencing algorithm [32]. Kim *et al.* adopted graph-based representations in solving selective and parallel disassembly problems [33], [34]. Li *et al.* created another graph-based scheme using a hybrid graph by '*pruning the search space of disassembly sequences, grouping related components into subassemblies, and identifying free components to facilitate disassembly operations*' [35].

Built on the graph approach, many algorithms and rule-based methods have been used to calculate disassembly sequences. An example is the Fuzzy Reasoning Petri Net proposed by Zhao and Li [36]. However, the generation of a graph relies on human understanding instead of machine interpretation. A more advanced method involves using matrices to represent the relationships between components which can be directly recognised and calculated by computer. Smith *et al.* proposed a tool consisting of five matrices to represent an assembly and used several rules to generate disassembly sequences [37], [38]. Tao *et al.* also modified the matrices to enable partial/parallel disassembly [39]. However, this optimisation-focused work did not reduce the complexity of the mathematical representation of an assembly in which distinguishing between fasteners and general parts was needed although their definitions were fuzzy and could cause confusion in many cases. For example, it is not clear whether to categorise objects in press-fit components as fasteners or general parts. Another matrix-based example can be found in the work of Jin *et al.* [40], [41], in which the relationships between components were represented using just a matrix.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

More recently, Liu *et al.* adopted the Bees Algorithm [42], [43] to identify time-efficient disassembly sequences [4]. Laili *et al.* addressed the problem of failed disassembly operations and proposed a re-planning strategy to revise disassembly sequences in the case of a failure [44]. Laili *et al.* also presented a very fast version of the Bees Algorithm to minimise computation time [44]. Gong *et al.* addressed the combination of disassembly scheduling and planning as a multi-objective problem [45]. However, the methods presented so far are essentially sequential disassembly methods and cannot work correctly in dealing with interlocked structures.

This paper presents a new and flexible matrix-based disassembly sequence planning approach. It consists of addressing sequential disassembly operations first and then dealing with special structures by breaking them into subassemblies for which dismantling planning can readily resume. The proposed approach can generate feasible disassembly sequences depending on the objective of disassembly: selective or complete disassembly. Figure 1 shows the key concepts behind the approach and the sections in which they will be discussed. Section 2 describes the mathematical representation of physical assemblies. An assembly matrix is used to derive three matrices to reflect spatial information related to contact and spatial interference. Section 3 presents two key processes in disassembly sequence planning: removability checking and separability checking. Section 4 explains the proposed disassembly sequence planning procedure using the representation described in Section 2 and the two processes discussed in Section 3. Section 5 gives a case study to demonstrate the proposed approach.

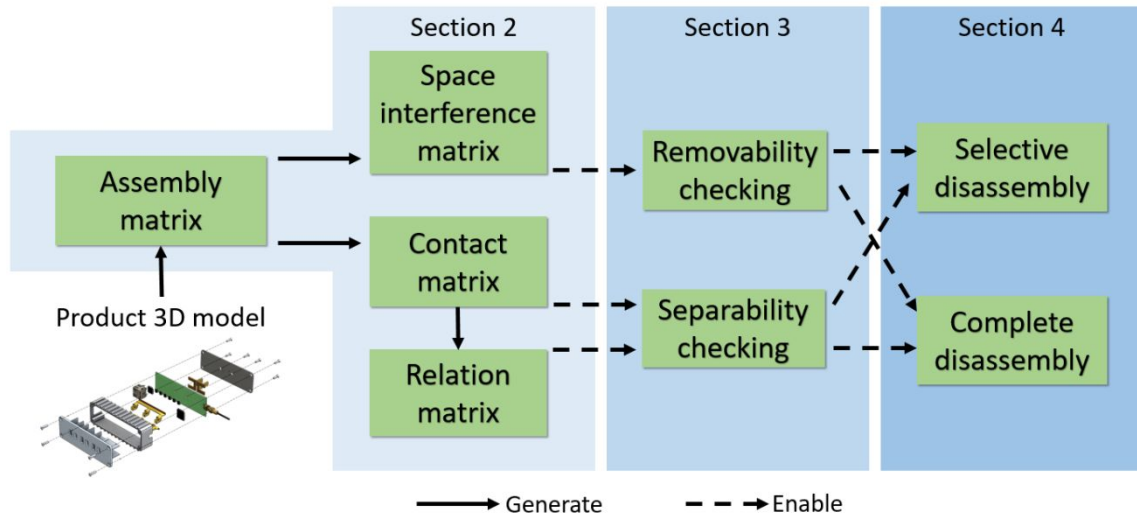


Figure 1. Key concepts discussed in the paper

2. Mathematical representation of physical assemblies

This paper proposes the ‘assembly matrix’ A , as a fundamental tool to interpret a physical assembly. It is an improved approach based on the space interference matrix, used by Jin et al. [40], [41], to represent space interference in an assembly in six directions (X+, X-, Y+, Y-, Z+, Z-). The assembly matrix uses all six directions. In this section, only four directions (X+, X-, Y+, Y-) are employed to illustrate the proposed method in two dimensions, as shown in Eq. 1.

$$[A] = \begin{matrix} & C_1 & \dots & C_n \\ \begin{matrix} C_1 \\ \vdots \\ C_n \end{matrix} & \begin{bmatrix} a_{11.x} + a_{11.x} - a_{11.y} + a_{11.y} - & \dots & a_{1n.x} + a_{1n.x} - a_{1n.y} + a_{1n.y} - \\ \vdots & \ddots & \vdots \\ a_{n1.x} + a_{n1.x} - a_{n1.y} + a_{n1.y} - & \dots & a_{nn.x} + a_{nn.x} - a_{nn.y} + a_{nn.y} - \end{bmatrix} \end{matrix} \quad (1)$$

In Equation 1, C_n represents components in an assembly. $a_{pq.x+}$, $a_{pq.x-}$, $a_{pq.y+}$, and $a_{pq.y-}$ indicate the relationships between the components in the corresponding columns and rows by using three states: 0 for no interference, 1 for contact and 2 for ‘remote interference’. For example, the assembly in Figure 2 [41] can be represented by the assembly matrix in Eq. 2. In the matrix, $a_{12.x} + a_{12.x} - a_{12.y} + a_{12.y} -$ is 2201 because C_2 is a remote obstacle for C_1 in the X+ and X- directions, and a direct contact in the Y- direction. C_1 can be removed from C_2 in the Y+ direction. Similarly, $a_{21.x} + a_{21.x} - a_{21.y} + a_{21.y} -$ is 2210 because C_1 is a remote obstacle for C_2 in the X+ and X-

directions, and a direct contact in Y+ direction. C_2 can be removed from C_1 in the Y- direction. It is worth noting that the matrix may not be symmetrical as $a_{pq.x} + a_{pq.x} - a_{pq.y} + a_{pq.y} -$ may differ from $a_{qp.x} + a_{qp.x} - a_{qp.y} + a_{qp.y} -$. For example, $a_{61.x} + a_{61.x} - a_{61.y} + a_{61.y} -$ is 1110 and $a_{16.x} + a_{16.x} - a_{16.y} + a_{16.y} -$ is 1111, because removing f_1 from C_1 is a legitimate operation but the reverse is not. f_1 can be removed from C_1 by unfastening using a screwdriver to rotate f_1 . Removing C_1 from f_1 , on the other hand, requires rotating C_1 reversely, which is usually not legitimate due to geometrical constraints and lack of tools.

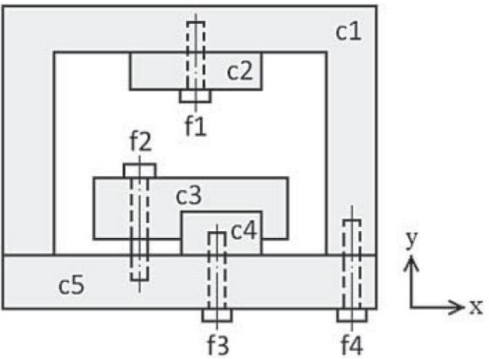


Figure 2. An example product [41]

Assembly Matrix

$$A = \begin{matrix} & C_1 & C_2 & C_3 & C_4 & C_5 & f_1 & f_2 & f_3 & f_4 \\ \begin{matrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \end{matrix} & \begin{bmatrix} 0000 & 2201 & 2202 & 2202 & 0001 & 1111 & 2202 & 2202 & 1111 \\ 2210 & 0000 & 0002 & 0002 & 0002 & 1111 & 0002 & 0002 & 0000 \\ 2220 & 0020 & 0000 & 1101 & 0002 & 0020 & 1111 & 2202 & 2000 \\ 2220 & 0020 & 1110 & 0000 & 0001 & 0020 & 0200 & 1111 & 2000 \\ 0010 & 0020 & 0020 & 0010 & 0000 & 0020 & 1111 & 1111 & 1111 \\ 1110 & 1110 & 0002 & 0002 & 0002 & 0000 & 0000 & 0002 & 0000 \\ 2220 & 0020 & 1101 & 2000 & 1101 & 0000 & 0000 & 2000 & 2000 \\ 2220 & 0020 & 2220 & 1110 & 1110 & 0020 & 0200 & 0000 & 2000 \\ 1110 & 0000 & 0200 & 0200 & 1110 & 0000 & 0200 & 0200 & 0000 \end{bmatrix} \end{matrix}$$

(2)

Contact Matrix

$$\begin{array}{c}
 C_1 \\
 C_2 \\
 C_3 \\
 C_4 \\
 C_5 \\
 f_1 \\
 f_2 \\
 f_3 \\
 f_4
 \end{array}
 \mathbf{C} =
 \begin{array}{c}
 C_1 \\
 C_2 \\
 C_3 \\
 C_4 \\
 C_5 \\
 f_1 \\
 f_2 \\
 f_3 \\
 f_4
 \end{array}
 \begin{bmatrix}
 0000 & 0001 & 0000 & 0000 & 0001 & 1111 & 0000 & 0000 & 1111 \\
 0010 & 0000 & 0000 & 0000 & 0000 & 1111 & 0000 & 0000 & 0000 \\
 0000 & 0000 & 0000 & 1101 & 0000 & 0000 & 1111 & 0000 & 0000 \\
 0000 & 0000 & 1110 & 0000 & 0001 & 0000 & 0000 & 1111 & 0000 \\
 0010 & 0000 & 0000 & 0010 & 0000 & 0000 & 1111 & 1111 & 1111 \\
 1110 & 1110 & 0000 & 0000 & 0000 & 0000 & 0000 & 0000 & 0000 \\
 0000 & 0000 & 1101 & 0000 & 1101 & 0000 & 0000 & 0000 & 0000 \\
 0000 & 0000 & 0000 & 1110 & 1110 & 0000 & 0000 & 0000 & 0000 \\
 1110 & 0000 & 0000 & 0000 & 1110 & 0000 & 0000 & 0000 & 0000
 \end{bmatrix}
 \quad (3)$$

Space Interference Matrix

$$\begin{array}{c}
 C_1 \\
 C_2 \\
 C_3 \\
 C_4 \\
 C_5 \\
 f_1 \\
 f_2 \\
 f_3 \\
 f_4
 \end{array}
 \mathbf{I} =
 \begin{array}{c}
 C_1 \\
 C_2 \\
 C_3 \\
 C_4 \\
 C_5 \\
 f_1 \\
 f_2 \\
 f_3 \\
 f_4
 \end{array}
 \begin{bmatrix}
 0000 & 1101 & 1101 & 1101 & 0001 & 1111 & 1101 & 1101 & 1111 \\
 1110 & 0000 & 0001 & 0001 & 0001 & 1111 & 0001 & 0001 & 0000 \\
 1110 & 0010 & 0000 & 1101 & 0001 & 0010 & 1111 & 1101 & 1000 \\
 1110 & 0010 & 1110 & 0000 & 0001 & 0010 & 0100 & 1111 & 1000 \\
 0010 & 0010 & 0010 & 0010 & 0000 & 0010 & 1111 & 1111 & 1111 \\
 1110 & 1110 & 0001 & 0001 & 0001 & 0000 & 0000 & 0001 & 0000 \\
 1110 & 0010 & 1101 & 1000 & 1101 & 0000 & 0000 & 1000 & 1000 \\
 1110 & 0010 & 1110 & 1110 & 1110 & 0010 & 0100 & 0000 & 1000 \\
 1110 & 0000 & 0100 & 0100 & 1110 & 0000 & 0100 & 0100 & 0000
 \end{bmatrix}
 \quad (4)$$

Relation Matrix

$$\begin{array}{c}
 C_1 \\
 C_2 \\
 C_3 \\
 C_4 \\
 C_5 \\
 f_1 \\
 f_2 \\
 f_3 \\
 f_4
 \end{array}
 \mathbf{R} =
 \begin{array}{c}
 C_1 \\
 C_2 \\
 C_3 \\
 C_4 \\
 C_5 \\
 f_1 \\
 f_2 \\
 f_3 \\
 f_4
 \end{array}
 \begin{bmatrix}
 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\
 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\
 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0
 \end{bmatrix}
 \quad (5)$$

Based on the assembly matrix, three new matrices, a contact matrix \mathbf{C} , a space interference matrix \mathbf{I} , and a relation matrix \mathbf{R} , can be derived, as shown in Figure 3. \mathbf{C} and \mathbf{I} indicate the existence or absence of contact and interference, respectively, between components along different directions. \mathbf{R} reveals the general contact status of components considering all directions. For example, using Equation 2, \mathbf{C} , \mathbf{I} and \mathbf{R} for the example in Figure 2 are given in Equations 3, 4 and 5 respectively.

Element C_{21} and C_{12} are 0010 and 0001, respectively, because components C_1 and C_2 are in contact along the y direction (+y for C_2 and -y for C_1). The contact condition determines that both R_{21} and R_{12} are 1. In addition to the y direction, the two components are also in interference along the x direction, which explains that I_{21} is 1110 and I_{12} 1101.

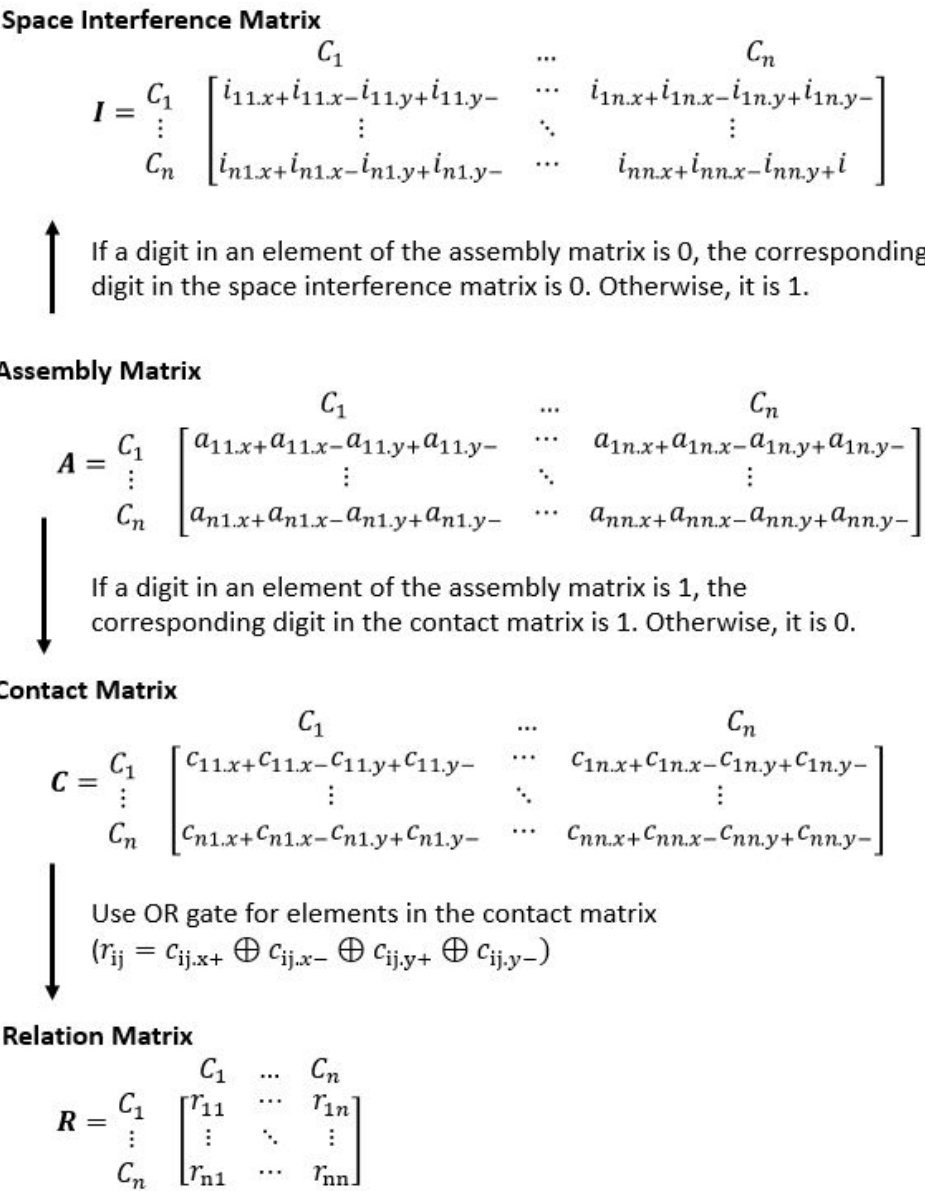


Figure 3. Derivation of a contact matrix, a space interference matrix and a relation matrix from an assembly matrix

3. Disassembly Model: two key processes

Two key processes are needed in the generation of disassembly sequences:

- (1) Removability checking. This process identifies individual components that have the freedom to be taken away.
- (2) Separability checking. This process enables the building of subassemblies.

3.1 Removability checking

Jin et al. [40], [41] proposed a method to identify removable components to generate feasible disassembly sequences using the space interference matrix. The essence of their approach is to find components that have freedom in at least one direction, indicating that the components are removable. A product can be disassembled after multiple cycles of sequentially taking away removable components one by one.

However, if the above method is adopted for the product depicted in Figure 2, after the removal of f_3 and f_4 in the first step, no components can be further disassembled, as shown in Figure 4. This is a typical interlocked structure. An assembly cannot be disassembled as no parts are removable or reachable until the whole structure is broken into smaller subassemblies. Although detection of subassemblies has been proposed in the literature [44], [46]–[48], the focus was not on this interlocking problem.

Studies on subassemblies (i.e. part segregations) have been driven by design optimisation and minimising production time. For example, Maiyar et al. adopted a part segregation optimisation algorithm in a product design, so that the time of production (which incorporates additive manufacturing and manual assembly) can be minimised [49]. Although there could be a degree of similarity between segmentation problems and interlocking problems, the methods proposed for the former has yet to be applied to the latter.

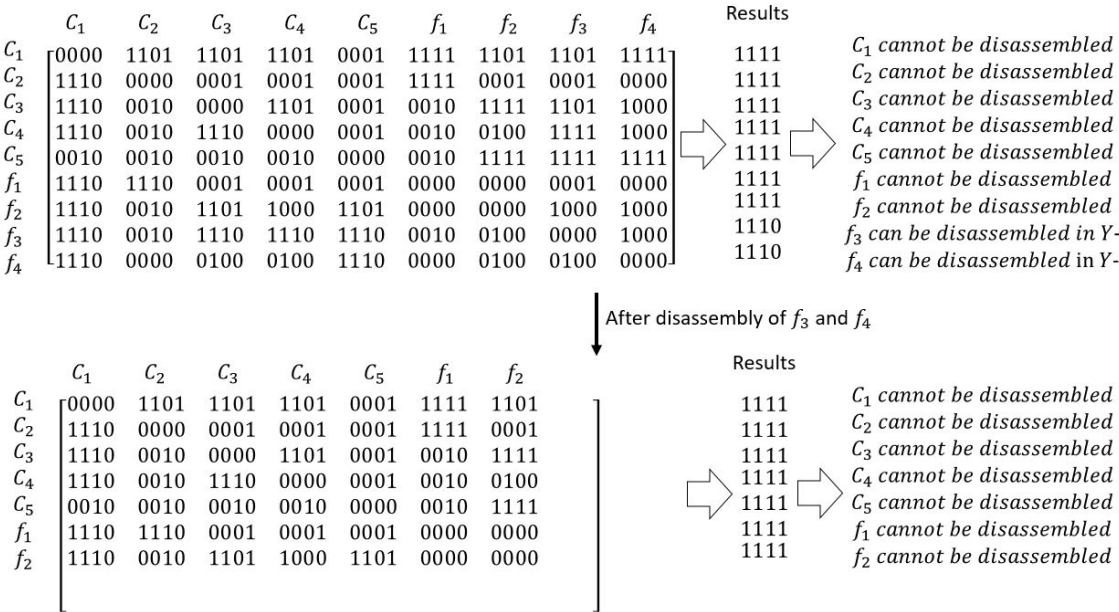


Figure 4. Sequential disassembly method proposed by Jin et al. [40], [41] which explain the calculation methods to obtain 'Results'. For each row, the result is obtained by performing OR gate operations. For example, in the row of C_1 , $0000 + 1101 + 1101 + 1101 + 0001 + 1111 + 1101 = 1111$, where '+' is an OR gate operator. The result '1111' indicates that the component C_1 is unable to be disassembled from any direction (i.e. X+, X-, Y+ and Y-).

3.2 Separability Check

Based on an analysis of over 239 mechanical products by the authors' team, some 23% contain interlocked structures which cannot be correctly dealt with using sequential disassembly methods [50]. In the given example, the interlocked structure would require separating C_1 and C_5 to divide the assembly into two sub-assemblies. This can be performed by considering the separability of components in a product which indicates whether it can be broken into subassemblies. The separability of an assembly is determined by whether it contains 'separable pairs', pairs of contacting components that can be separated without affecting other contacting components.

For example, the assembly in Figure 5a has three components: A1, B1 and C1, and two pairs of contacting components: A1-B1 and B1-C1. If a contact between a pair can be represented as a line, then the physical model in Figure 5a can be simplified to Figure 5b, which can also be represented by its relation matrix ($R1$), as shown in Figure 5c. Both pairs, A1-B1 and B1-C1, are separable, as the separation of either pair would not affect the other.

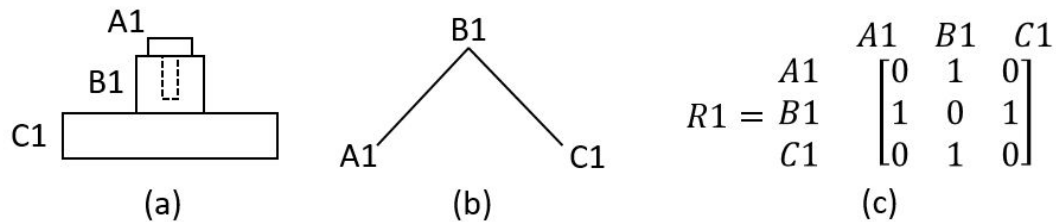


Figure 5. An example of a product comprising separable pairs.

However, in a similar model shown in Figure 6, the result would be different. None of the three pairs, A2-B2, B2-C2 and A2-C2, are separable, as the separation of a pair could affect other pairs. For example, the separation of A2-B2 inevitably causes the detachment of A2 from C2. When comparing Figure 6b to Figure 5b, it is obvious that there is only one path between A1 and B1 (A1 - B1) in Figure 5b, but there are two paths between A2 and B2 (A2-B2, and A2-C2-B2) in Figure 6b. If there is only one path between two components, it means the interaction between them is not coupled with those between other components. A sufficient condition for a pair to be separable is that there is only one path between two nodes in a pair, as is the case with pairs A1-B1 and B1-C1 in Figure 5b.

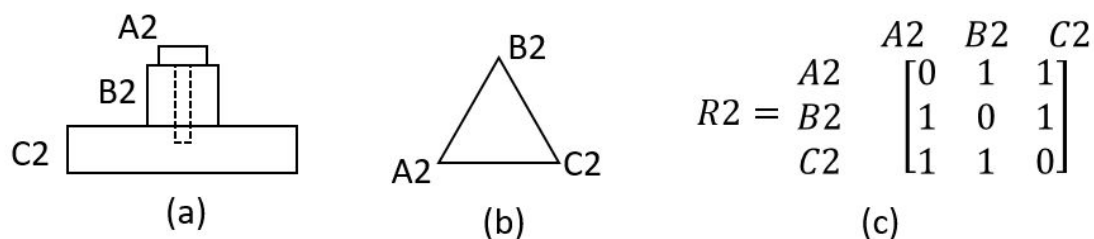


Figure 6. An examples of a product comprising inseparable pairs

Therefore, the key to overcoming interlocked structures is to identify separable pairs in relation matrices. Searching for separable pairs in the relation matrix can follow three steps (Figure 7).

The first step is to search for adjacent pairs, i.e. two components in contact, which can be identified directly in the relation matrix. If an element in the matrix is 1, the corresponding components in the column and row are in contact, and therefore they constitute an adjacent pair.

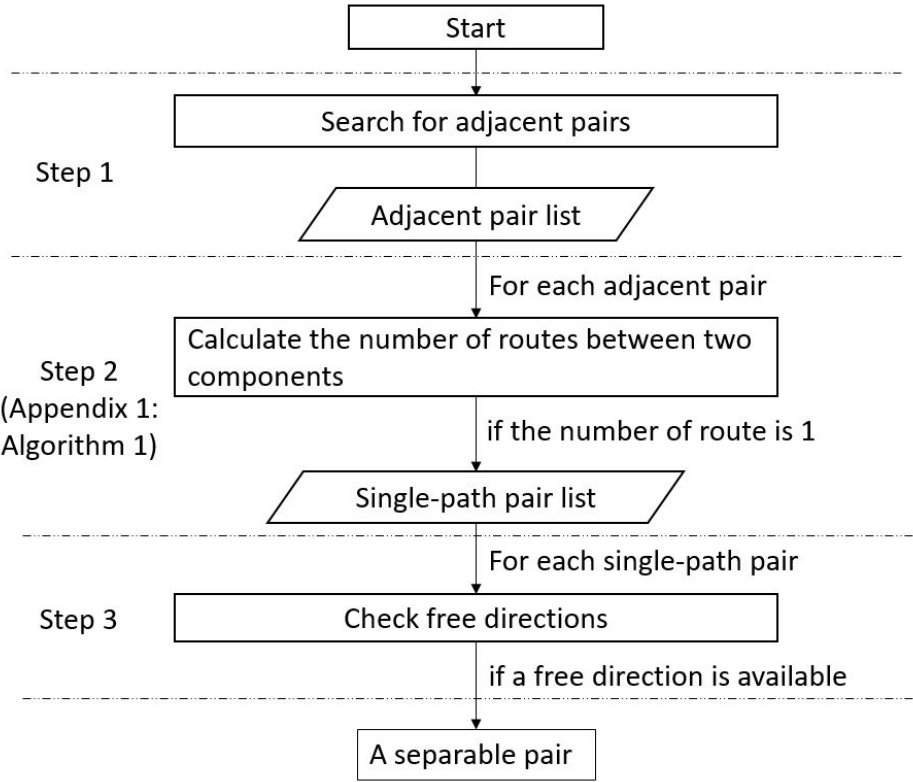


Figure 7. Steps to identify a separable pair

The second step is to identify the pair for which there is only one route between the two components, a sufficient condition for a pair to be separable, as discussed earlier. We propose using a recursive strategy summarised by the pseudo code in Algorithm 1 (Appendix 1).

After all single-path pairs are identified, their corresponding elements in the contact matrix **C** (step 3 in Figure 7) should be checked. If the element is not 1111 ($c_{pq.x} + c_{pq.x} - c_{pq.y} + c_{pq.y} \neq 1111$),

one component has freedom in at least one direction in relation to the other, and thus the pair is separable.

For example, after the removal of f_3 and f_4 , given in Figure 8, the relation matrix of the assembly is given in Equation 6. In Step 1, eight adjacent pairs (Table 1) can be found through checking the value of the elements in R . Algorithm 1 is used to calculate the number of routes between two components in a pair. It starts with the pair $\{C1, C2\}$, in which C1 is the origin and C2 is the destination. The result indicates that the pair is not separable, as there are two routes from C1 to C2 ($C1 \rightarrow C2$ and $C1 \rightarrow f1 \rightarrow C2$), as depicted in Figure 9. For the next member on the adjacent pair list, $\{C1, C5\}$, only one route is found, and thus the pair is added to single-path pair list. The calculation continues for all pairs on the adjacent pair list, and $\{C1, C5\}$ is the only single-path pair. As C1 and C5 have freedom in 3 directions, the pair is a separable pair.

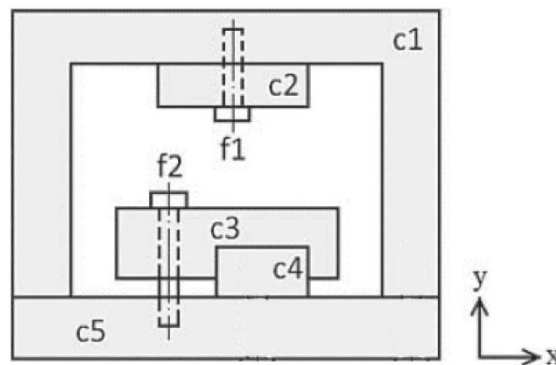


Figure 8. The example assembly in Fig. 2 after the removal of f_3 and f_4 .

step 1	Adjacent pair list	<div> {C1, C2} {C2, f1} </div> <div> {C1, C5} {C3, C4} </div> <div> {C1, f1} {C4, C5} </div> <div> {C2, C3} {C5, f2} </div>
step 2	Single-path list	{C1, C5}
step 3	Separable pair	{C1, C5}

Table 1. An example of identifying separable pairs

Step 1	Adjacent pair list	$\{C1, C2\}$ $\{C2, f1\}$ $\{C1, C5\}$ $\{C3, C4\}$ $\{C1, f1\}$ $\{C4, C5\}$ $\{C2, C3\}$ $\{C5, f2\}$
Step 2	Single-path list	$\{C1, C5\}$
Step 3	Separable pair	$\{C1, C5\}$

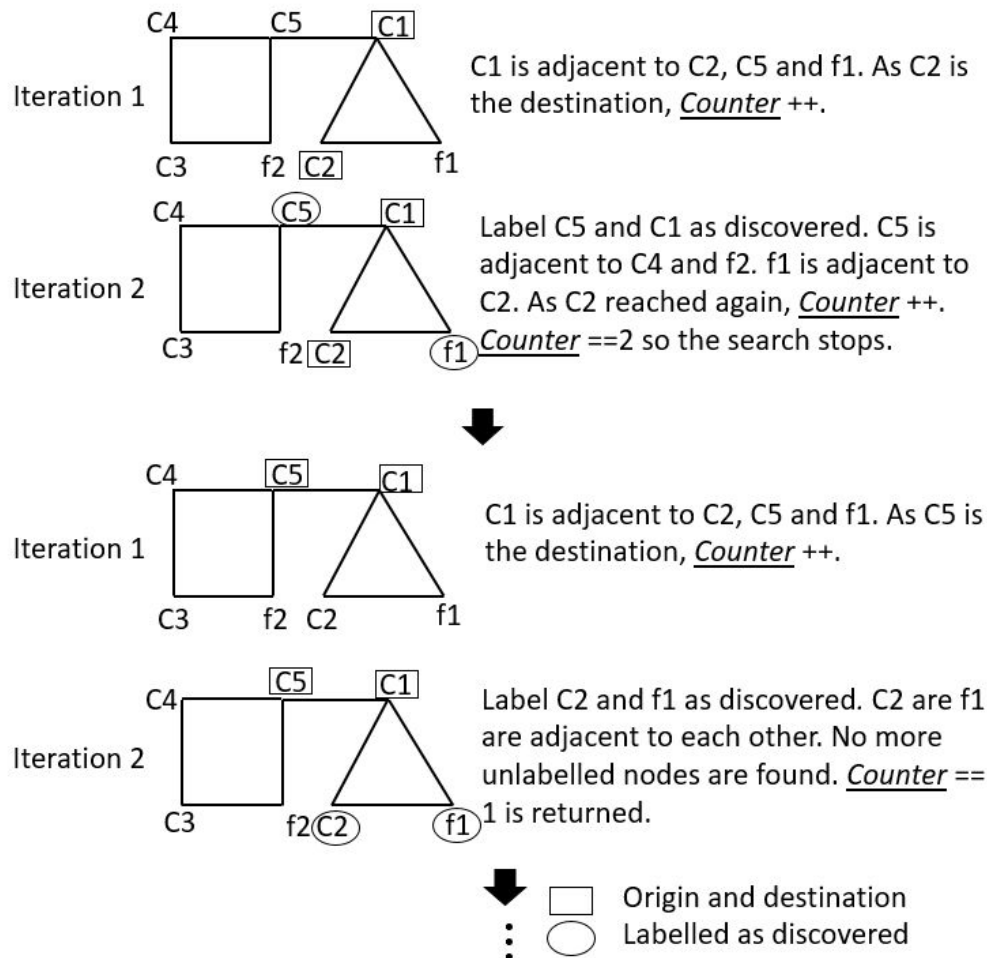


Figure 9. An example of searching for single-path pairs

Furthermore, the separation of C1 and C5 results in two subassemblies: C1-C2-f1 and C3-C4-C5-f2. Then, f2 and f1 become removable and disassembly using sequential disassembly methods could carry on.

4. Disassembly Model: procedures

The proposed model can generate disassembly sequences for both selective disassembly and complete disassembly using the two processes (Removability and Separability) discussed in the last section. For selective disassembly, the procedure for generating the disassembly plan is shown in Figure 10. It contains four important processes:

- (1). Search for adjacent components

- (2). Check removability
- (3). Check separability
- (4). Build subassemblies

It is a process of iteratively checking for and removing the components on the to-be-disassembled list consisting of the selected component and other interfering components. It has the selected component only in the first iteration, but will expand in later iterations if interfering components are identified. In each iteration, the removability and separability of the components on the list are checked to identify removable components or build subassemblies. If all components are fixed, adjacent components are added to the to-be-disassembled list and to be checked in the next iteration. Processes (1), (3) and (4) were presented in Section 3.2, and Process (2), in Section 3.1. If a component or a separable pair is identified, related components will be added to the disassembly list and dismantled.

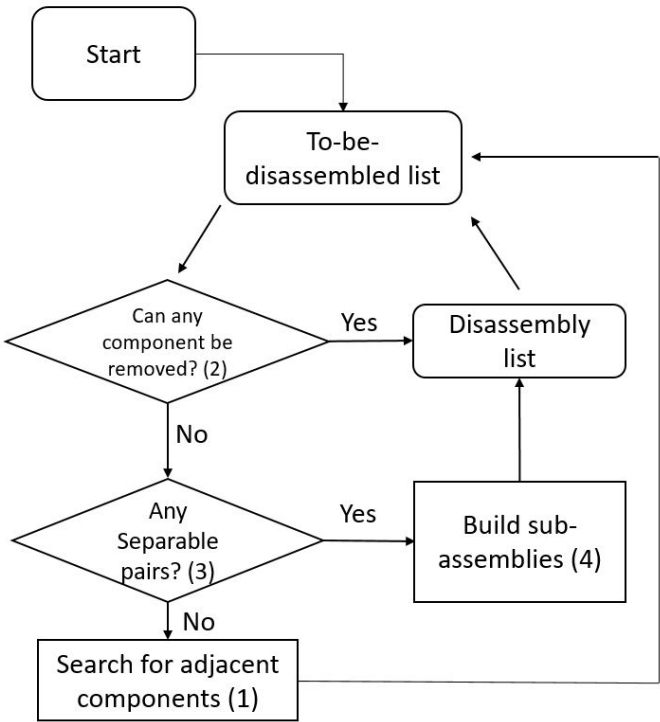


Figure 10. Planning for selective disassembly

For example, assuming C4 is the target component in the assembly (Figure 2), the disassembly process in each iteration is shown in Table 2. In the first iteration, the final target C4 is placed on the

to-be-disassembled list. As it cannot be removed (C4: 1111 in Figure 4) and there are no separable pairs (one part cannot form a pair), searching for adjacent components is triggered. C3, C5 and f3 are found and added to the to-be-disassembled list. Although f3 is removed in iteration 2, no removable components are found in iteration 3, triggering another cycle of adjacent parts search which indicates two more components: f2 and f4. After removing f4 in iteration 4, there are again no removable parts, but a separable pair C1 and C5 is found in iteration 5. Therefore, the assembly is broken into two subassemblies: f2-C3-C4-C5 and f1-C1-C2. As the target part, C4, is in the former subassembly, all components in the latter are removed including C1 which is already on the to-be-disassembled list. In iteration 6, f2 is now found to be removable. C3 and C5 are removed in iteration 7 to allow the final release of C4.

Table 2. Selective disassembly example

Iteration	To-be-disassembled list	Can any component be removed?	Any separable pairs?	Adjacent parts	Subassembly	Disassembly list
1	C4	No	No	C3, C5, f3	-	-
2	f3, C3, C5; C4	f3 - 1110	-	-	-	f3;
3	C3, C5; C4	No	No	C1, f2, f4	-	f3;
4	C1, f2, f4; C3, C5; C4	f4 - 1110	-	-	-	f3; f4;
5	C1, f2; C3, C5; C4	No	Yes, C1-C5	-	f2-C3-C4-C5 and C1-C2-f1	f3; f4; C1;
6	f2; C3, C5; C4	f2 - 1101	-	-	-	f3; f4; C1; f2;
7	C3, C5; C4	C3 - 1101 C5 - 0010	-	-	-	f3; f4; C1; f2; C3, C5;
8	C4	C4 - 0000	-	-	-	f3; f4; C1; f2; C3, C5; C4;

Planning for a complete disassembly is similar to that for selective disassembly and uses processes (2), (3) and (4), as shown in Figure 11. It also employs an iterative approach to identify all

removable components and disassemble them in multiple iterations. If removable components are not found, a separability check is carried out to identify separable pairs.

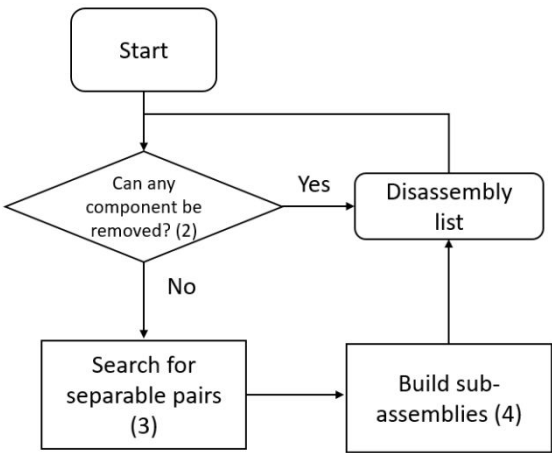


Figure 11. Planning for complete disassembly

For the example in Figure 2, the complete disassembly procedure in different iterations is shown in Table 3. After removing f3 and f4 in iteration 1, there were no removable components in iteration 2, as indicated by using the space interference matrix. C1-C5 was found to be a separable pair, resulting in breaking of the assembly into two subassemblies: f2-C3-C4-C5 and f1-C2-C1. Afterwards, all components are removable and can be disassembled in a sequential way.

Table 3. Complete disassembly example

Iteration	Can be removed?	Separable pairs	Subassembly	Disassembly list
1	f3 - 1110 f4 - 1110	-	-	f3, f4;
2	None	C1-C5	f2-C3-C4-C5 and f1-C2-C1	f3, f4;
3	f1 - 1110 f2 - 1101	-	-	f3, f4; f1,f2;
4	C1 - 1101 C2 – 1110 C3 - 1101 C5 - 0010	-	-	f3, f4; f1,f2; C1, C2, C3, C5;
5	C4 - 0000	-	-	f3, f4; f1,f2; C2, C1, C3, C5; C4

5. Case Study

This section discusses the disassembly of a piston in a 4-stroke engine, as shown in Figure 12. The components of the piston are listed in Table 4 and its assembly matrix is given in Appendix 2. Using the method depicted in Figure 3, the contact matrix (Appendix 3), space interference matrix (Appendix 4), relation matrix (Appendix 5) can be generated automatically.

The purpose of the case study is to demonstrate that the model is able to generate disassembly sequences for either complete or selective disassembly using the assembly matrix of the piston only. In the case of complete disassembly, the disassembly of the interlocked structure B-C1-C2-D requires building subassemblies B-C1 and C2-D so that C1 and C2 can be removed from B and D respectively.

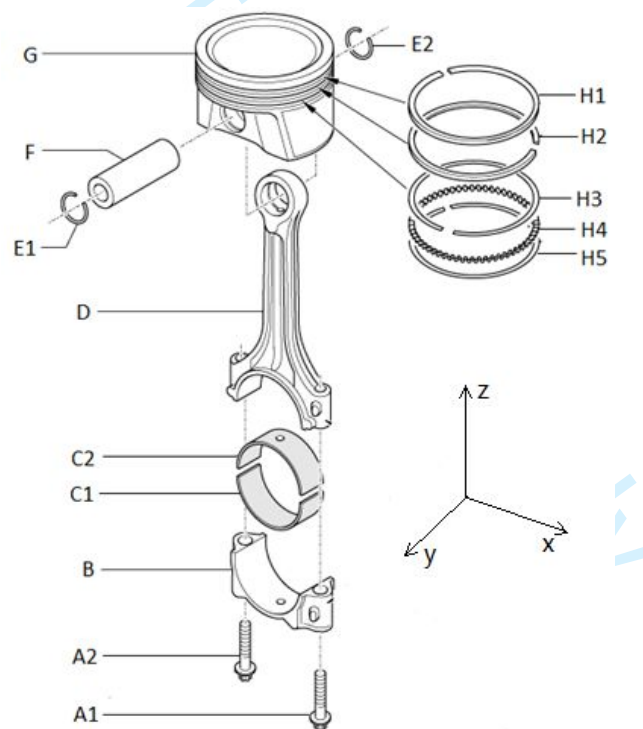


Figure 12. Parts in a piston

The piston head (G) is likely to be the only part to be reused in remanufacturing. It was chosen to demonstrate the use of the model for selective disassembly planning. A disassembly sequence, shown in Table 5, can be generated using the method in Figure 9. All parts contacting G

can be disassembled in a sequence and searching for separable pairs was not needed. After the space interference matrix was generated using the assembly matrix, the procedure was similar to that explained in [40].

Table 4. Part information

Code	Components
A1, A2	Bolts
B	Connecting rod bearing cap
C1, C2	Bearing shells
D	Connecting rod
E1, E2	Circular clips
F	Piston pin
G	Piston head
H1, H2, H3, H4, H5	Piston rings

Table 5. Selective disassembly of a piston head

Iteration	To-be-disassembled list	Removable components	Separable pairs	Adjacent parts	Sub-assembly	Disassembly list
1	G	None	None	D, E1, E2, F, H1, H2, H3, H4, H5	-	-
2	D, E1, E2, F, H1, H2, H3, H4, H5; G	E1 - 110111 E2 - 111011 H1 - 111101	-	-	-	E1, E2, H1
3	D, F, H2, H3, H4, H5; G	F - 110011 H2 - 111101	-	-	-	E1, E2, H1; F, H2
4	D, H3, H4, H5; G	H3 - 111101	-	-	-	E1, E2, H1; F, H2; H3;
5	D, H4, H5; G	H4 - 111101	-	-	-	E1, E2, H1; F, H2; H3; H4;
6	D, H5; G	H5 - 111101	-	-	-	E1, E2, H1; F, H2; H3; H4; H5;
7	D, G	G - 111101	-	-	-	E1, E2, H1; F, H2; H3; H4; H5; G

However, when complete disassembly was required, breaking of the assembly into subassemblies was needed, as shown in iteration 7 in Table 6. A separable-pair search process in

iteration 7 was triggered to find B-D to create two subassemblies B-C1 and C2-D. In the next step, sequential disassembly carried on.

For Peer Review Only

Table 6. Complete disassembly of a piston

Iteration	Removable components	Separable pairs	Subassembly	Disassembly list
1	A1 - 111110 A2 - 111110 E1 - 110111 E2 - 111011 H1 - 111101	-	-	A1, A2, E1, E2, H1;
2	F - 110011 H2 - 111101	-	-	A1, A2, E1, E2, H1; F, H2
3	H3 - 111101	-	-	A1, A2, E1, E2, H1; F, H2; H3
4	H4 - 111101	-	-	A1, A2, E1, E2, H1; F, H2; H3; H4
5	H5 - 111101	-	-	A1, A2, E1, E2, H1; F, H2; H3; H4; H5
6	G - 111101	-	-	A1, A2, E1, E2, H1; F, H2; H3; H4; H5; G;
7	None	B-D	B-C1 and C2-D	A1, A2, E1, E2, H1; F, H2; H3; H4; H5; G;
8	C1 - 111101 C2 - 111110	-	-	A1, A2, E1, E2, H1; F, H2; H3; H4; H5; G; C1, C2;
9	B - 000000 D - 000000	-	-	A1, A2, E1, E2, H1; F, H2; H3; H4; H5; G; C1, C2; B, D;

The model provides a new approach for generating feasible disassembly sequences with the following benefits:

- Reduction in difficulty of transforming 3D data into mathematical representations
- The presented method only uses the assembly matrix, which contains complete information about disassembly in three dimensions, rather than multiple models as with existing methods.
- The simple model reduces the difficulty of information extraction (to generate the assembly matrix) which would be needed in building a fully autonomous disassembly planning/re-planning system.

➤ Flexible use

The model based on the assembly matrix can be used for both selective and complete disassembly. Two paths can be chosen to manipulate the matrix to reach either result. They share modules, which simplifies programming.

➤ An efficient and simple solution for products with interlocked structures

Information about interlocked structures that cannot be dismantled is contained in the new model using the new concept of separable pairs. The new model can automatically break an assembly into subassemblies if it finds sequential disassembly is no longer possible. Previous efforts involved using a modular approach to group parts together so the new groups can be treated as ordinary parts [51]. However, that approach would require high computation capabilities if the number of parts is large.

6. Conclusion

Machine understanding of the structure of an assembly in three-dimensional space is required for autonomous disassembly planning. A key step is to create a mathematical representation (or model) of physical contacts, constraints and interferences that is readable to machines. Another key step is to develop suitable algorithms to calculate feasible disassembly sequences using the new representation. Conventionally, because of the complexity of spatial information, models tended to be complex and normally not suitable for all products, in particular, those containing interlocked structures. This paper presents a new mathematical representation of an assembly, the assembly matrix, which is needed only for generating feasible disassembly sequences. It can trigger the breaking of an assembly into subassemblies when sequential disassembly of components one at a time is not possible. A case study was used to demonstrate its function based on different scenarios.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

In future work, a suitable CAD extraction method will be developed to build assembly matrices automatically. Rules and algorithms that can change the assembly matrix online will also be created. It would be useful if machines could realise when a part supposed to be removable is no longer so, which could happen in remanufacturing due to rust and deformation. By combining the two functions, a true autonomous disassembly planning system could be achieved.

Competing interests. *We declare we have no competing interests.*

Acknowledgements. *This paper is an extended version of the article ‘Automatic Detection of Subassemblies for Disassembly Sequence Planning’ published In Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics (1), pp. 104-110, 2018. The authors acknowledge Dr Yuanjun Laili, Beihang University, for her comments on the research. This research is supported by the EPSRC (Grant No. EP/N018524/1), the Royal Society (Grant No. (IEC\NSFC\181018) and the National Science Foundation of China (Grant No. 51775399).*

References

- [1] M. R. Johnson and I. P. McCarthy, "Product recovery decisions within the context of Extended Producer Responsibility," *J. Eng. Technol. Manag.*, vol. 34, pp. 9–28, 2014.
- [2] M. Matsumoto and W. Ijomah, "Remanufacturing," in *Handbook of Sustainable Engineering*, Dordrecht: Springer Netherlands, 2013, pp. 389–408.
- [3] EC, *Manifesto for a resource-efficient Europe*. 2012.
- [4] J. Liu, Z. Zhou, D. T. Pham, W. Xu, C. Ji, and Q. Liu, "Robotic disassembly sequence planning using enhanced discrete bees algorithm in remanufacturing," *Int. J. Prod. Res.*, vol. 56, no. 9, pp. 3134–3151, May 2018, doi: 10.1080/00207543.2017.1412527.
- [5] Z. Yuan, J. Bi, and Y. Moriguchi, "The Circular Economy: A New Development Strategy in China," *J. Ind. Ecol.*, vol. 10, no. 1–2, pp. 4–8, Feb. 2006, doi: 10.1162/108819806775545321.
- [6] L. Wang, X. Xu, R. Gao, and A. Y. C. Nee, "Sustainable cybernetic manufacturing." Taylor & Francis, 2019.
- [7] P. Rosa, C. Sassanelli, A. Urbinati, D. Chiaroni, and S. Terzi, "Assessing relations between Circular Economy and Industry 4.0: a systematic literature review," *Int. J. Prod. Res.*, vol. 58, no. 6, pp. 1662–1687, 2020.
- [8] L. Zhu, Z. Zhang, and Y. Wang, "A Pareto firefly algorithm for multi-objective disassembly line balancing problems with hazard evaluation," *Int. J. Prod. Res.*, vol. 56, no. 24, pp. 7354–7374, 2018.
- [9] M. L. Bentaha, A. Dolgui, O. Battaia, R. J. Riggs, and J. Hu, "Profit-oriented partial disassembly line design: dealing with hazardous parts and task processing times uncertainty," *Int. J. Prod. Res.*, vol. 56, no. 24, pp. 7220–7242, 2018.
- [10] M. Liu, X. Liu, F. Chu, F. Zheng, and C. Chu, "Robust disassembly line balancing with ambiguous task processing times," *Int. J. Prod. Res.*, pp. 1–30, 2019.
- [11] E. Özceylan, C. B. Kalayci, A. Güngör, and S. M. Gupta, "Disassembly line balancing problem: a review of the state of the art and future directions," *Int. J. Prod. Res.*, vol. 57, no. 15–16, pp. 4805–4827, 2019.
- [12] F. Zheng, J. He, F. Chu, and M. Liu, "A new distribution-free model for disassembly line balancing problem with stochastic task processing times," *Int. J. Prod. Res.*, vol. 56, no. 24, pp. 7341–7353, 2018.
- [13] Y. Fang, H. Ming, M. Li, Q. Liu, and D. T. Pham, "Multi-objective evolutionary simulated annealing optimisation for mixed-model multi-robotic disassembly line balancing with interval processing time," *Int. J. Prod. Res.*, vol. 58, no. 3, pp. 846–862, 2020.
- [14] Q. Liu, Z. Liu, W. Xu, Q. Tang, Z. Zhou, and D. T. Pham, "Human-robot collaboration in disassembly for sustainable manufacturing," *Int. J. Prod. Res.*, vol. 57, no. 12, pp. 4027–4044, Jun. 2019, doi: 10.1080/00207543.2019.1578906.
- [15] R. Li *et al.*, "Unfastening of Hexagonal Headed Screws by a Collaborative Robot," *IEEE Trans. Autom. Sci. Eng.*, 2020.
- [16] M. M. L. Chang, A. Y. C. Nee, and S. K. Ong, "Interactive AR-assisted product disassembly sequence planning (ARDIS)," *Int. J. Prod. Res.*, pp. 1–16, 2020.
- [17] R. Romero-Silva and E. Marsillac, "Trends and topics in IJPR from 1961 to 2017: a statistical history," *Int. J. Prod. Res.*, vol. 57, no. 15–16, pp. 4692–4718, 2019.
- [18] P. Kopacek and G. Kronreif, "Semi-automated robotic disassembling of personal

- computers," in *Proceedings 1996 IEEE Conference on Emerging Technologies and Factory Automation. ETFA'96*, 1996, vol. 2, pp. 567–572.
- [19] S. Vongbunyong and W. H. Chen, *Disassembly Automation*. 2015.
- [20] P. Gil, J. Pomares, S. V. T. Puente, C. Diaz, F. Candelas, and F. Torres, "Flexible multi-sensorial system for automatic disassembly using cooperative robots," *Int. J. Comput. Integr. Manuf.*, vol. 20, no. 8, pp. 757–772, Dec. 2007, doi: 10.1080/09511920601143169.
- [21] M. Barwood, J. Li, T. Pringle, and S. Rahimifard, "Utilisation of reconfigurable recycling systems for improved material recovery from e-waste," in *Procedia CIRP*, 2015, vol. 29, pp. 746–751, doi: 10.1016/j.procir.2015.02.071.
- [22] A. J. D. Lambert, "Disassembly sequencing: a survey," *Int. J. Prod. Res.*, vol. 41, no. 16, pp. 3721–3759, 2003.
- [23] A. Bourjault, "Contribution à une approche méthodologique de l'assemblage automatisé: élaboration automatique des séquences opératoires," *Thèse d'Etat, Univ. Fr.*, 1984.
- [24] T. De Fazio and D. Whitney, "Simplified generation of all mechanical assembly sequences," *IEEE J. Robot. Autom.*, vol. 3, no. 6, pp. 640–658, 1987.
- [25] L. S. H. De Mello and A. C. Sanderson, "AND/OR graph representation of assembly plans," *IEEE Trans. Robot. Autom.*, vol. 6, no. 2, pp. 188–199, 1990.
- [26] L. S. H. De Mello and A. C. Sanderson, "A correct and complete algorithm for the generation of mechanical assembly sequences," *IEEE Trans. Robot. Autom.*, vol. 7, no. 2, pp. 228–240, 1991.
- [27] T. Kanehara, T. Suzuki, A. Inaba, and S. Okuma, "On algebraic and graph structural properties of assembly Petri net-searching by linear programming," in *Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93)*, 1993, vol. 3, pp. 2286–2293.
- [28] D. NAVTN-CHANDRA, "The recovery problem in product design," *J. Engineering Des.*, vol. 5, no. 1, pp. 65–86, 1994.
- [29] A. D. De Ron and K. Penev, "Disassembly and recycling of electronic consumer products: an overview," *Technovation*, vol. 15, no. 6, pp. 363–374, 1995.
- [30] Y. Pnueli and E. Zussman, "Evaluating the end-of-life value of a product and improving it by redesign," *Int. J. Prod. Res.*, vol. 35, no. 4, pp. 921–942, Apr. 1997, doi: 10.1080/002075497195452.
- [31] H. R. Krikke, A. van Harten, and P. C. Schuur, "On a medium term product recovery and disposal strategy for durable assembly products," *Int. J. Prod. Res.*, vol. 36, no. 1, pp. 111–140, 1998.
- [32] F. Torres, S. T. Puente, and R. Aracil, "Disassembly planning based on precedence relations among assemblies," *Int. J. Adv. Manuf. Technol.*, vol. 21, no. 5, pp. 317–327, 2003.
- [33] H.-W. Kim and D.-H. Lee, "An optimal algorithm for selective disassembly sequencing with sequence-dependent set-ups in parallel disassembly environment," *Int. J. Prod. Res.*, vol. 55, no. 24, pp. 7317–7333, 2017.
- [34] H.-W. Kim, C. Park, and D.-H. Lee, "Selective disassembly sequencing with random operation times in parallel disassembly environment," *Int. J. Prod. Res.*, vol. 56, no. 24, pp. 7243–7257, 2018.
- [35] J. R. Li, L. P. Khoo, and S. B. Tor, "A novel representation scheme for disassembly sequence planning," *Int. J. Adv. Manuf. Technol.*, vol. 20, no. 8, pp. 621–630, 2002.

- 1
 - 2
 - 3
 - 4
 - 5
 - 6
 - 7
 - 8
 - 9
 - 10
 - 11
 - 12
 - 13
 - 14
 - 15
 - 16
 - 17
 - 18
 - 19
 - 20
 - 21
 - 22
 - 23
 - 24
 - 25
 - 26
 - 27
 - 28
 - 29
 - 30
 - 31
 - 32
 - 33
 - 34
 - 35
 - 36
 - 37
 - 38
 - 39
 - 40
 - 41
 - 42
 - 43
 - 44
 - 45
 - 46
 - 47
 - 48
 - 49
 - 50
 - 51
 - 52
 - 53
 - 54
 - 55
 - 56
 - 57
 - 58
 - 59
 - 60
- [36] S. Zhao and Y. Li, "Disassembly Sequence Decision Making for Products Recycling and Remanufacturing Systems," in *2010 International Symposium on Computational Intelligence and Design*, 2010, pp. 44–48, doi: 10.1109/ISCID.2010.19.
 - [37] S. Smith and W.-H. Chen, "Rule-Based Recursive Selective Disassembly Sequence Planning for Green Design," Springer, London, 2009, pp. 291–302.
 - [38] S. Smith, G. Smith, and W.-H. Chen, "Disassembly sequence structure graphs: An optimal approach for multiple-target selective disassembly sequence planning," *Adv. Eng. Informatics*, vol. 26, no. 2, pp. 306–316, Apr. 2012, doi: 10.1016/j.aei.2011.11.003.
 - [39] F. Tao, L. Bi, Y. Zuo, and A. Y. C. Nee, "Partial/parallel disassembly sequence planning for complex products," *J. Manuf. Sci. Eng.*, vol. 140, no. 1, 2018.
 - [40] G. Q. Jin, W. D. Li, and K. Xia, "Disassembly Matrix for Liquid Crystal Displays Televisions," *Procedia CIRP*, vol. 11, pp. 357–362, 2013, doi: 10.1016/j.procir.2013.07.015.
 - [41] G. Jin, W. Li, S. Wang, and S. Gao, "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.*, Mar. 2015, doi: 10.1177/0954405415575476.
 - [42] D. T. Pham, A. Ghanbarzadeh, E. Koç, S. Otri, S. Rahim, and M. Zaidi, "The bees algorithm—a novel tool for complex optimisation problems," in *Intelligent Production Machines and Systems*, Elsevier, 2006, pp. 454–459.
 - [43] D. T. Pham and M. Castellani, "The bees algorithm: modelling foraging behaviour to solve continuous optimization problems," *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.*, vol. 223, no. 12, pp. 2919–2938, 2009.
 - [44] Y. Laili, F. Tao, D. T. Pham, Y. Wang, and L. Zhang, "Robotic disassembly re-planning using a two-pointer detection strategy and a super-fast bees algorithm," *Robot. Comput. Integr. Manuf.*, vol. 59, pp. 130–142, Oct. 2019, doi: 10.1016/j.rcim.2019.04.003.
 - [45] G. Gong, Q. Deng, R. Chiong, X. Gong, H. Huang, and W. Han, "Remanufacturing-oriented process planning and scheduling: mathematical modelling and evolutionary optimisation," *Int. J. Prod. Res.*, pp. 1–19, 2019.
 - [46] H. Cao, R. Mo, N. Wan, F. Shang, C. Li, and D. Zhang, "A subassembly identification method for truss structures manufacturing based on community detection," *Assem. Autom.*, vol. 35, no. 3, pp. 249–258, Aug. 2015, doi: 10.1108/AA-02-2015-011.
 - [47] Y. Wang and J. Liu, "Subassembly identification for assembly sequence planning," *Int. J. Adv. Manuf. Technol.*, vol. 68, no. 1–4, pp. 781–793, Sep. 2013, doi: 10.1007/s00170-013-4799-y.
 - [48] N. S. Ong and Y. C. Wong, "Automatic Subassembly Detection from a Product Model for Disassembly Sequence Generation," *Int. J. Adv. Manuf. Technol.*, vol. 15, no. 6, pp. 425–431, Jun. 1999, doi: 10.1007/s001700050086.
 - [49] L. M. Maiyar, S. Singh, V. Prabhu, and M. K. Tiwari, "Part segregation based on particle swarm optimisation for assembly design in additive manufacturing," *Int. J. Comput. Integr. Manuf.*, vol. 32, no. 7, pp. 705–722, Jul. 2019, doi: 10.1080/0951192X.2019.1610577.
 - [50] C. Ji, D. T. Pham, S. Su, J. Huang, and Y. Wang, "AUTOREMAN--D. 1.1-List of generic disassembly task categories," 2017.
 - [51] S. Smith and P.-Y. Hung, "A novel selective parallel disassembly planning method for

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

green design,” *J. Eng. Des.*, vol. 26, no. 10–12, pp. 283–301, Dec. 2015, doi:
10.1080/09544828.2015.1045841.

For Peer Review Only

Appendix 1 –Algorithm 1. Generate single-path pair list from adjacent pair list

Main function:

Input: adjacent pair list (APL)

Output: Single-path list (SPL)

1 **For** every pair $\{X, Y\} \in \text{APL}$

2 $\text{counter} = 0$

3 $\text{searchPath}(X, Y)$;

4 **If** $\text{counter} = 1$

5 add $\{X, Y\}$ to SPL;

6 **End if**

7 **End for**

$\text{searchPath}(X, Y)$

8 Label X as discovered

9 **For** every component k adjacent to X

10 **If** k is not labeled as discovered

11 **If** $k = Y$

12 $\text{counter}++$;

13 **If** $\text{counter} \geq 2$

14 **break**;

15 **End if**

16 **Else**

17 **Recursively call** $\text{searchPath}(k, Y)$

18 **End if**

19 **End if**

19 Return counter

20 **End for**

This algorithm first selects a pair $\{X, Y\}$, defines a counter, and calls a function $\text{searchPath}(X, Y)$ which recursively calculates the number of routes (Line 1 to 3). If the number of routes is 1, a counter equal to 1 is returned and the pair $\{X, Y\}$ is added to the single-path pair list (Line 4 to 6). The function $\text{searchPath}(X, Y)$ starts with labelling X as the origin of a route and identifying the components adjacent to X (Line 8). If Y is found to be an adjacent component, a route between X and Y is established and thus the counter increases (Line 11 to 12). The counter over 1 indicates that more than one route has been found between X and Y , and thus the search can stop as X and Y is not a separable pair (Line 13 to 15). Otherwise, the search should continue by recursively calling $\text{searchPath}(k, Y)$ in which an adjacent component k becomes the new origin in the next recursion (Line 17), until the destination Y is found.

Appendix 2 – Assembly matrix for the piston

	A1	A2	B	C1	C2	D	E1	E2	F	G	H1	H2	H3	H4	H5
A1	000000	020000	111110	020000	020000	111110	000000	000000	000000	000020	000020	000020	000020	000020	000020
A2	200000	000000	111110	200000	200000	111110	000000	000000	000000	000020	000020	000020	000020	000020	000020
B	111111	111111	000000	111111	000020	000010	000000	000000	000020	000020	000020	000020	000020	000020	000020
C1	200000	020000	111101	000000	000020	000020	000000	000000	000020	000020	000020	000020	000020	000020	000020
C2	200000	020000	000002	000002	000000	111111	000000	000000	000020	000020	000020	000020	000020	000020	000020
D	111111	111111	000001	000002	111111	000000	002000	000200	000100	221120	000020	000020	000020	000020	000020
E1	000000	000000	000000	000000	000000	000200	000000	000200	000100	110111	000000	000000	000000	000000	000000
E2	000000	000000	000000	000000	000000	002000	002000	000000	001000	110111	000000	000000	000000	000000	000000
F	000000	000000	000002	000002	000002	110011	001000	000100	000000	110011	000000	000000	000000	000000	000000
G	000002	000002	000002	000002	000002	221102	111111	111111	111111	000000	111102	000001	000001	000001	111111
H1	000002	000002	000002	000002	000002	000002	000000	000000	000000	111102	000000	000001	000001	000002	000002
H2	000002	000002	000002	000002	000002	000002	000000	000000	000000	111102	000010	000000	000000	000002	000002
H3	000002	000002	000002	000002	000002	000002	000000	000000	000000	111102	000020	000010	000000	000001	000002
H4	000002	000002	000002	000002	000002	000002	000000	000000	000000	111101	000020	000020	000010	000000	000001
H5	000002	000002	000002	000002	000002	000002	000000	000000	000000	111101	000020	000020	000010	000000	000000

Appendix 3– Contact Matrix

	A1	A2	B	C1	C2	D	E1	E2	F	G	H1	H2	H3	H4	H5
A1	000000	000000	111110	000000	000000	111110	000000	000000	000000	000000	000000	000000	000000	000000	000000
A2	000000	000000	111110	000000	000000	111110	000000	000000	000000	000000	000000	000000	000000	000000	000000
B	111111	111111	000000	111111	000000	000010	000000	000000	000000	000000	000000	000000	000000	000000	000000
C1	000000	000000	111101	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000
C2	000000	000000	000000	000000	000000	111110	000000	000000	000000	000000	000000	000000	000000	000000	000000
D	111111	111111	000001	000000	111111	000000	000000	000000	110011	001100	000000	000000	000000	000000	000000
E1	000000	000000	000000	000000	000000	000000	000000	000000	000000	111111	000000	000000	000000	000000	000000
E2	000000	000000	000000	000000	000000	000000	000000	000000	000000	111111	000000	000000	000000	000000	000000
F	000000	000000	000000	000000	000000	110011	001000	000100	000000	111111	000000	000000	000000	000000	000000
G	000000	000000	000000	000000	000000	001100	111111	111111	111111	000000	111100	000000	111100	000000	111101
H1	000000	000000	000000	000000	000000	000000	000000	000000	000000	111100	000000	000000	000000	000000	000000
H2	000000	000000	000000	000000	000000	000000	000000	000000	000000	111100	000010	000000	000000	000000	000000
H3	000000	000000	000000	000000	000000	000000	000000	000000	000000	111100	000000	000010	000000	000000	000000
H4	000000	000000	000000	000000	000000	000000	000000	000000	000000	111100	000000	000010	000000	000000	000000
H5	000000	000000	000000	000000	000000	000000	000000	000000	000000	111101	000000	000010	000000	000000	000000

Appendix 4 – Space Interference Matrix

	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	001000	110011	000010	000010	000010	000010	000010
E1	000000	000000	000000	000000	000000	000100	000000	000000	001000	110111	000000	000000	000000	000000	000000
E2	000000	000000	000000	000000	000000	001000	001000	000000	001000	111011	000000	000000	000000	000000	000000
F	000000	000000	000001	000001	000001	110011	001000	000100	000000	110011	000000	000000	000000	000000	000000
G	000001	000001	000001	000001	000001	111101	111111	111111	111111	000000	111101	111111	111111	111111	111111
H1	000001	000001	000001	000001	000001	000001	000000	000000	000000	111101	000000	000001	000001	000001	000001
H2	000001	000001	000001	000001	000001	000001	000000	000000	000000	111101	000010	000000	000001	000001	000001
H3	000001	000001	000001	000001	000001	000001	000000	000000	000000	111101	000010	000010	000000	000001	000001
H4	000001	000001	000001	000001	000001	000001	000000	000000	000000	111101	000010	000010	000010	000000	000001
H5	000001	000001	000001	000001	000001	000001	000000	000000	000000	111101	000010	000010	000010	000010	000000

Appendix 5 – Relation Matrix

	A1	A2
A1	0	0
A2	0	0
B	1	1
C1	0	0
C2	0	0
D	1	1
E1	0	0
E2	0	0
F	0	0
G	0	0
H1	0	0
H2	0	0
H3	0	0
H4	0	0
H5	0	0