Disassembly Sequence Planning for End-of-Life Products

By

Hengyu Wang

A thesis submitted to the Faculty of Graduate Studies of

University of Manitoba

in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

Department of Mechanical Engineering

University of Manitoba

Winnipeg

Abstract

Nowadays, manufactures are under tremendous pressure to dispose product in an environmentally responsible way to pursue a sustainable development. Disassembly operations are required in the product recycling and maintenance period. An optimal disassembly sequence can reduce the disassembly cost and time. This thesis proposes an efficient method for selective disassembly sequence planning (DSP). The proposed method includes two main aspects: product representation and sequence searching. Multi-level constraint matrices based on product's bill of material (BOM) are constructed. This representation approach can identify the product's hierarchical structure to reduce the searching size of the sequence plans. Traversal algorithm and genetic algorithm are used to search the desired disassembly sequence. A disassembly feasibility check is integrated in the genetic algorithm to generate a better disassembly sequence with a less searching time. Several case studies are used to verify the proposed algorithms. In addition, destructive disassembly operations are considered to remove those constraints that cannot be removed by the non-destructive disassembly. Disassembly cost comparison is made between the destructive disassembly and non-destructive disassembly. The solution with the less cost is selected.

Acknowledgements

I would like to take this opportunity to thank my supervisor, Dr. Qingjin Peng. Without his constant support and great patience, I would not be able to complete this research thesis. His guidance and encouragement is very important for me to achieve this master degree.

I would also like to give my thanks to colleagues in the Virtual Manufacturing Lab and friends who helped me for my research and thesis.

Finally, I want to thank my parents, my wife and my sister. They gave me endless support and strength to chase my dream.

Contents

Chapter 1 Introduction....1

1.1 Background....1

1.2 Disassembly Classification....3

1.3 Constraint categories....5

1.4 Research Objectives....8

1.5 Thesis Outline....9

Chapter 2 Literature Review....10

2.1 Product Representation....12

2.1.1 Graph-based Representation....12

2.1.2 Constraint Matrix....14

2.2 Sequence Planning Approach....16

2.2.1 DSP based on the graph representation....16

2.2.2 DSP methods based on the matrix representation....18

2.2.3 Methods comparison and problems conclusion....20

2.3 Bill of Material....22

2.4 Summary....24

Chapter 3 DSP based on multi-layer constraint matrices....25

3.1 Introduction....25

3.2 Multi-layer constraint matrix....26

3.3 Traversal algorithm for selective disassembly....32

3.3.1 Constraint matrix analysis....32

3.3.2 Implementation of the traversal algorithm....33

3.4 Conclusions....42

Chapter 4 DSP using genetic algorithm..43

4.1 Introduction 43

4.2 Parameters in the genetic algorithm for DSP..45

4.3 Fitness evaluation 47

4.4 Implementation of the genetic algorithm...51

4.5 Case Study for disassembly feasibility check 54

4.6 Summary...60

Chapter 5 Destructive Method for DSP..61

5.1 Introduction 61

5.2 Literature review and research objective 62

5.3 UTD problems in the non-destructive disassembly 63

5.4 Product presentation and component constraints...65

5.4.1 Constraint information for the destructive disassembly.................................65

5.4.2 Multi-level constraint matrices and fastener matrix......................................67

5.5 Disassembly sequence planning 67

5.5.1 Disassembly cost criteria..67

5.5.2 Flowchart of proposed disassembly sequence planning.................................68

5.6 Case studies for the destructive DSP 72

5.6.1 Mechanical arm...72

5.6.2 MP4 player...81

5.7 Summary...85

Chapter 6 Conclusion and Future work..86

6.1 Research contribution 86

6.2 Future work..87

References: 88

Appendix 1...99

Appendix 2...101

List of Tables

Table 3-1 Parts of the Gear Box……………………………………………………...………28

Table 3-2 Comparison between representation methods…………………………….............32

Table 3-3 Disassembly time for components…………………………………………………...40

Table 3-4 Comparison between Searching methods……………………………………………42

Table 4-1 Disassembly time criteria…………………………………………………………….48

Table 4-2 Comparison between traversal algorithm and GA………………………………...53

Table 4-3 Component list and information…………………………………………….........55

Table 4-4 Initial population of GA……………………………………………………………..56

Table 4-5 Disassembly time (in second) for components in the camshaft assembly…………..57

Table 4-6 Near-optimal disassembly sequences of camshaft assembly………………………..59

Table 4-7 Comparison for Gas with and without feasibility check………………………….....60

Table 5-1 Constraints type between component *i* and *j*………………………...……………66

Table 5-2 Disassembly time using tools for different constraints……………………...……68

Table 5-3 Components' information of the Mechanical arm………………………...………74

Table 5-4 Components' information of the MP4…………………………………….……..82

List of Figures

Figure 1-1 Sustainable product developments in Green Manufacturing………………..………2

Figure 1-2 Constraint categories…………………………………………………….………...7

Figure 2-1 Flowchart of DSP……………………………………….……………………..10

Figure 2-2 AND/OR graph……………………………………….………………………..12

Figure 2-3 Adjacent Graphs…………………………………………………….…………13

Figure 2-4 Disassembly directions in the Cartesian coordinate system………………………..15

Figure 2-5 Example of BOM……………………………………………………………..23

Figure 3-1 Example of constraint Matrix……………………………………..……......26

Figure 3-2 A simplified Gear box model……………………………………………………27

Figure 3-3 BOM of the Gear Box……………………………………………….……..29

Figure 3-4 Four subassemblies…………………………………………………………...29

Figure 3-5 Constraint matrix for the first-level……………………………………………..30

Figure 3-6 Constraint matrices of second level……………………………………………..31

Figure 3-7 Updated constraint matrix of gear box…………………………………………….33

Figure 3-8 Flowchart of improved traversal algorithm……………………………………..34

Figure 3-9 Search subassembly FS2………………………………………………………..36

Figure 3-10 Rotation in a Cartesian Coordinates System…………………………………..37

Figure 3-11 User interface of the search program…………………………………………......38

Figure 3-12 User interface for inputting constraint matrix………………………………...39

Figure 3-13 Optimal disassembly sequence search for subassembly FS3……………………40

Figure 3-14 Output optimal disassembly sequences…………………………………………41

Figure 4-1 Flowchart of the genetic algorithm for DSP……………………………………...44

Figure 4-2 Updated flow chart of GA for DSP……………………………………………….52

Figure 4-3 Explosion graph of the camshaft assembly……………………………………….54

Figure 4-4 Constraint matrix of the camshaft………………………………………………...58

Figure 4-5 GA with disassembly feasibility check…………………………………………...58

Figure 4-6 GA without disassembly feasibility check………………………………………..59

Figure 5-1 Riveting and welding products…………………………………………………...61

Figure 5-2 Graph representation example……………………………………………………63

Figure 5-3 Search subassemblies with the target component………………………………...69

Figure 5-4 An example of the tree structure………………………………………………….69

Figure 5-5 disassembly sequence search for the target component…………….………….71

Figure 5-6 3D model of the mechanical arm…………………………………………………73

Figure 5-7 BOM of the mechanical arm………………………….……………….........74

Figure 5-8 (a) Subassembly 1, (b) Subassembly 2……………………………………75

Figure 5-9 Graph representation of the mechanical arm……………………………………..76

Figure 5-10 Constraint matrices……………………………………………………………...77

Figure 5-11 Fastener-component matrix of the mechanical arm……………………………78

Figure 5-12 An MP4 Player of the EOL stage……………………………………………….81

Figure 5-13 Graph representation of the MP4 player……………………………………….82

Figure 5-14 Constraint matrix of the MP4 player…………………………………………...83

Figure 5-15 Fastener-component matrix of MP4 player……………………………………....83

Chapter 1

Introduction

1.1 Background

During the past few decades, with the development of modern technologies and human aesthetic evolutions, products' life has been shortened rapidly (Xie, 2007). As a consequence, lots of products are discarded in the landfill with a huge waste of resources. Moreover, many electronic products are detrimental to the environment because of containing toxic materials. Therefore, manufacturers are under tremendous pressure to dispose products in an environmentally responsible way and pursue a sustainable development (Mamadou, 2014; Rusinko, 2007).

In 2011, the total scrap cars in China reached to 2.7 million and this scrap rate kept rising by 6% per year (Van, 2011). Also, there are approximately 7 million tons of wastes in electrical and electronic equipment (WEEE) generated in Europe per year (Walther, 2010). All of those products have negative impacts on environments. Especially for those electronic products with plastic components and heavy metals, the long degradation time and severe pollution have become an increasingly important environmental problem (Ongondo, 2010).

In order to reduce the pollution and resource wastes essentially, a conception of green manufacturing (GM) was proposed by many researchers (Bhattacharya, 2015; Rusinko, 2007). Based on the theory of green manufacturing, manufacturers are expected to consider eliminating resource wastes and environmental pollutions from the beginning of product development. A sustainable development flowchart of prospective green manufacturing is shown in Figure 1-1.

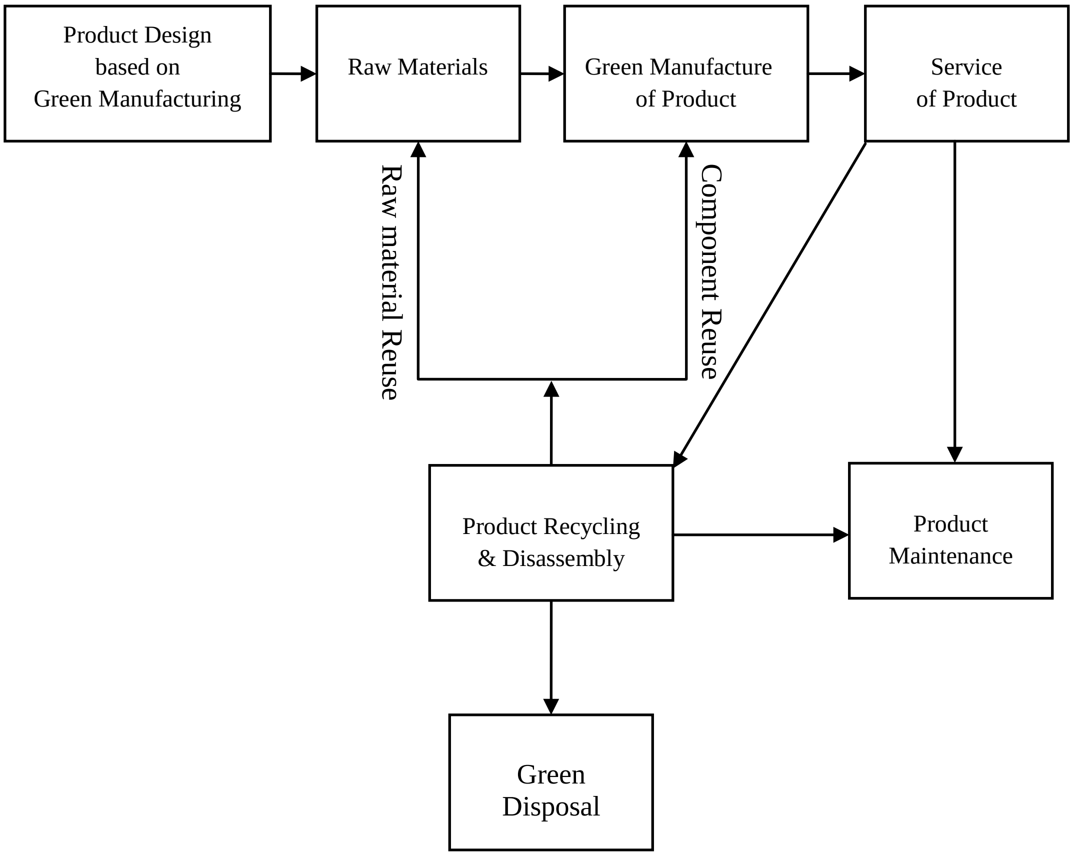


Figure 1-1 Sustainable product developments in Green Manufacturing

The expectation of Green Manufacturing is to reduce environmental pollution and resource wastes throughout the entire lifecycle of a product. In other words, it considers product design, manufacturing and maintenance to guarantee a minimum influence on environment and a maximum utilization of resources (Govindan, 2014). In conclusion, Green Manufacturing can help increasing manufacturers’ benefit without sacrificing the social benefits at the same time. Among those steps, product recycling and maintenance are regarded as two major measures of green manufacturing which can decrease the environment pollution significantly (Xie, 2007).

Disassembly is a basic and important operation in product recycling and maintenance stage. The purpose of disassembly for an End-of-Life (EOL) product is to collect reusable components or valuable materials. While in the maintenance process, a disassembly operation is to replace the malfunction components and maintain the product in the original state (Shana, 2010).

Usually, individual components can be removed out of a product in various sequences. In this study, the combination sequence with all disassembled components is called disassembly sequence (Basdere, 2003; Giri, 2015). For example, in order to recycle the CUP in a desktop computer, a disassembly worker could start from either the top case or side case. In this situation, two options in the disassembly process will have two different disassembly sequences. When the components to be removed increase, the number of disassembly sequences will be increased exponentially (Lambert, 2003). In order to evaluate those disassembly sequences and generate the “best one”, a conception of the disassembly cost (Van, 2011) was used to evaluate disassembly sequences. The disassembly cost includes the disassembly time cost, labor cost, and tooling cost.

In order to perform the disassembly operation in an effective and economical way, an optimal or near-optimal disassembly sequence is required. Studies on disassembly sequence planning (DSP) aim to generate a desired disassembly sequence based on specific criteria such as the shortest disassembly time or the minimum disassembly cost.

1.2 Disassembly Classification

Based on different disassembly requirements in a product life time, disassembly operations can be classified into two main categories: complete disassembly and incomplete disassembly (Viganò, 2013).

A complete disassembly operation could be regarded as the reverse processing of an assembly operation. It separates the entire product into detail components completely. In a real manufacturing operation, the complete disassembly is not frequently used because, no matter for the maintenance or recycle purpose, it is not necessary to disassemble a product into components completely. For example, in the maintenance operation, a disassembly operation only needs to replace the failure component (Walther, 2010). Although the complete disassembly is not commonly used in recycling and maintenance operations, it is still very important because the complete disassembly can be used to verify the feasibility of an assembly sequence and to train novices (Zhang, 2015). Therefore, complete disassembly sequence planning also plays an important role in DSP research and has been studied by many researchers (Bo, 2013).

The incomplete disassembly, on the contrary, only removes some components out of the product, which is frequently used for the maintenance and recycling purpose. Selective disassembly (SD) is one of the incomplete disassemblies (Kai, 2013). It is defined as the disassembly of selected components in a product. Selective disassembly separates selected components for recycling and maintenance. The target component or subassembly is known in the selective disassembly (Kara, 2006). Owing to the importance of the selective disassembly in product recycling and maintenance, this thesis will search the optimal or near-optimal disassembly sequence for the selective disassembly.

Disassembly operations also can be classified into destructive disassembly and non-destructive disassembly based on whether there is damage to components during the disassembly process (Park, 2002; Smith 2012). For materials recycling, a complete destructive disassembly is regarded as a commonly used approach, which includes drilling, shredding and other destructive methods (Lambert, 2003). In a real recycling operation, those methods could provide an effective and economic solution for materials recycling. By partially destroying constraints of components, a short disassembly path may appear to reduce the disassembly cost in a selective disassembly. The target component can be disassembled with less complexity (Bras, 2004; Yasushi, 2015).

Non-destructive disassembly is often required for activities such as maintenance or remanufacturing. It will disassemble a product mainly by removing constraints of components then reaching constraint-free components. Without damaging any component, the non-destructive disassembly aims to quickly isolate target components for a specific purpose. Nowadays, most studies on disassembly sequence planning focuses on this non-destructive disassembly method (Mitrouchev, 2015; Xie, 2007, Zhu, 2013). However, for an End-of-Life product, many components are useless and can be destroyed. In this case, partially destructive disassembly can help to find a short path to the target component (Song, 2013). In addition, when it comes to those constraints such as riveting and welding, non-destructive disassembly operations have to bypass them but destructive disassembly operations could remove them (Chen, 2014). In this thesis, the destructive disassembly will be considered in a selective disassembly and a comparison will be made between the non-destructive manner and destructive manner based on their disassembly costs.

1.3 Constraint categories

Constraint relations are used to maintain components in a specific position during the service cycle of a product (Sun, 2010). Therefore, in a disassembly operation, constraints should be removed preferentially in order to achieve constraint-free components. In this thesis, constraint relations are classified into three categories as follows:

(1) Contact Relations (CRs)

Contact relations are the most common constraint between components in a product. Any pair of components has a physical contact and constrained by their own geometric shapes that can be called the contact relation. In a real disassembly operation, the contact relation could be removed with hands or simple tools. For example, a back cover of Samsung cell phone is fixed on the phone body by its own geometric shape, and we can remove the back cover by hands (Pornsing, 2014).

(2) Fastening relations (FRs)

Fastening relations indicate the fastener constraints which can be removed by non-destructive operations. Fasteners are used to link parts together by the fastening force or the friction force such as thread connections, screw connections, interference fits, etc (Satou, 2009). They can be removed by common-used disassembly tools such as hammer, wrench, screwdriver, etc (Hoang, 2011).

(3) Destructive constraints (DCs)

Destructive methods have to be performed where a non-detachable connection should be removed. For example, if two components are connected by rivets, the rivets are deformed during the installation process hence they can’t be removed by non-destructive disassembly operations. Destructive constraints include riveting, welding and gluing etc. (Venkatesh, 2009).

Figure 1-2 shows the constraints relations introduced above.

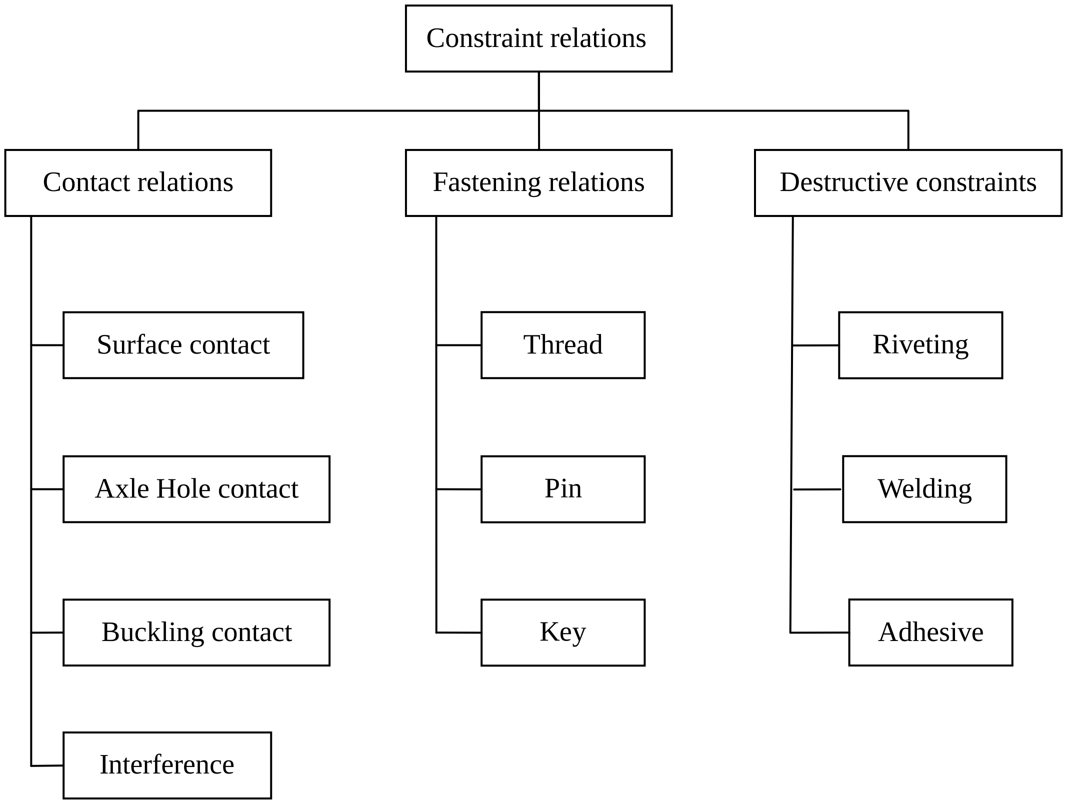


Figure 1-2 Constraint categories

In conclusion, in order to generate an optimal or near-optimal disassembly sequence, constraints of a product should be identified before planning the disassembly sequence. In a complete disassembly operation, all constraints should be removed including non-detachable constraints. In a selective disassembly, for the purpose to minimize the disassembly effort and cost, some non-detachable constraints may be bypassed. However, it is also very common that some constraints are removed by destructive approaches to reduce the total disassembly effort and cost (Kara, 2006).

1.4 Research Objectives

Disassembly sequence planning has been a popular research topic in recent years. Many researchers try to generate the optimal or near-optimal disassembly sequence using the existing or improved methods. However, due to different products with different geometric shapes and hierarchical relations, many of these methods are only effective to a certain kind of products or models. Moreover, there is a lack of considering the bill of material (BOM) of a product in the research. Many unnecessary components may be disassembled, and hence the disassembly workload and cost may be increased.

In this thesis, an efficient product representation approach and an optimized algorithm is proposed to generate an optimal or near-optimal disassembly sequence, especially for selective disassembly sequence planning. Product's BOM will be incorporated in the product representation to locate the target component and to reduce the number of components to be removed in a selective disassembly operation.

Destructive disassembly method is considered for disassembling an EOL product, and the disassembly cost is compared to the non-destructive disassembly cost. Moreover, when it comes to a certain constraint where a non-destructive disassembly operation could not work, a destructive disassembly operation is performed to minimize the disassembly cost. Finally, an improved genetic algorithm is proposed to solve DSP problems.

1.5 Thesis Outline

Chapter two will discuss three commonly-used product representation approaches and four types of sequence planning algorithms. They all worked well and proved to be feasible in their own cases.

In Chapter three, multi-level constraint matrices are constructed based on the product BOM. Traversal algorithm is proposed to search the multi-layer constraint matrices. They will work together to generate an optimal disassembly sequence in the selective disassembly operation.

In Chapter four, genetic algorithm is used to generate a near-optimal disassembly sequence. Disassembly feasibility check is performed during the crossover and mutation operations to skip infeasible disassembly sequences. In comparison, the GA with disassembly feasibility check can achieve a better solution in a shorter calculation time.

In Chapter five, the destructive disassembly operation is considered to get the target component in an EOL product. An optimal disassembly sequence is generated by comparing the non-destructive method with the destructive method. Two case studies are used to verify the proposed method.

In Chapter six, contributions of this thesis are concluded and the future work is discussed.

Chapter 2

Literature Review

Disassembly sequence planning (DSP) aims to find an optimal or near-optimal disassembly sequence. A general flowchart of disassembly sequence planning process is shown in Figure 2-1.

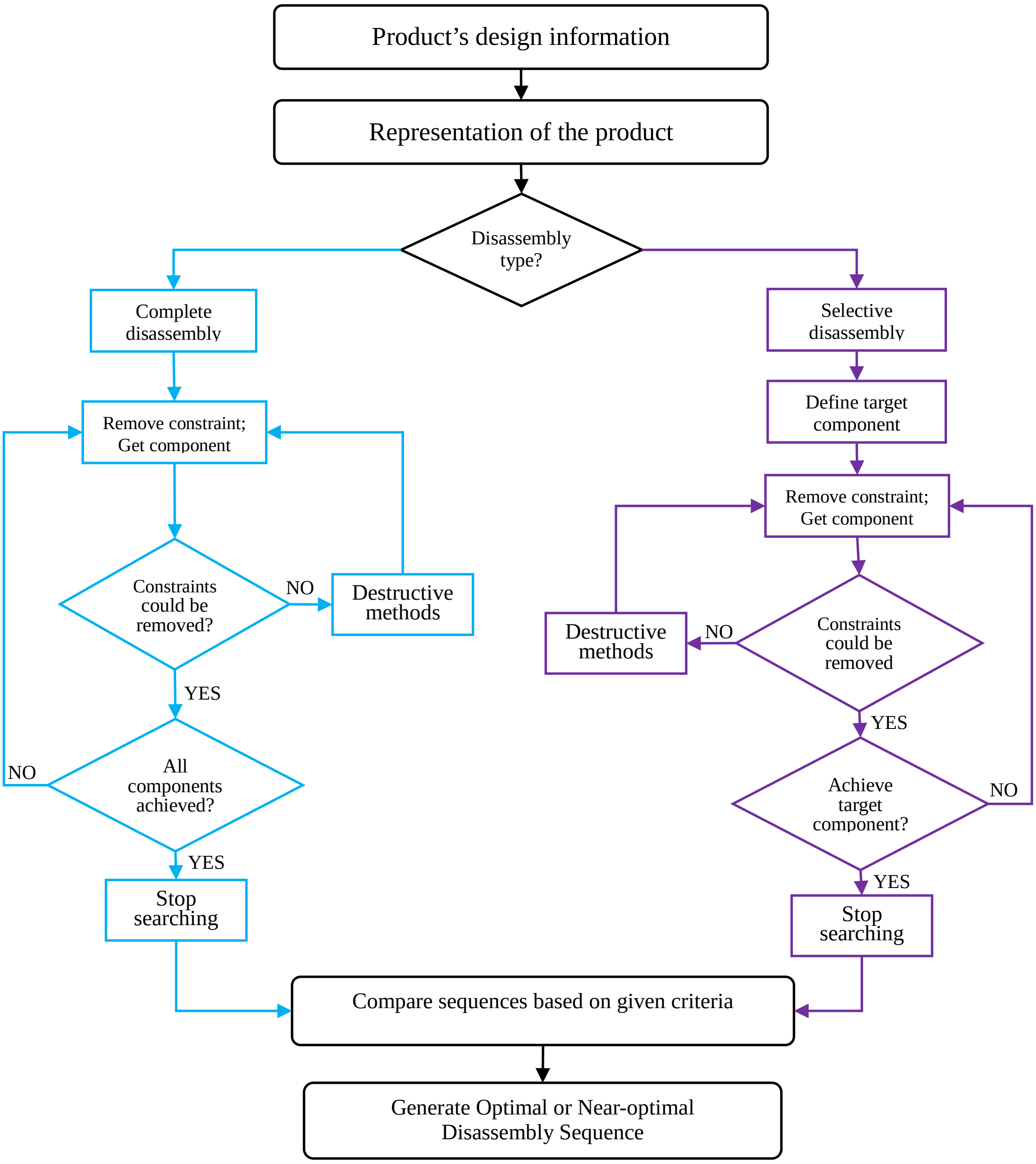


Figure 2-1 Flowchart of DSP

During a product design stage, the product details such as product's geometric structure, bill of material (BOM), components' relationships and constraints' types and quantity etc. are decided (Shana, 2011). In the disassembly sequence planning, the representation of a product is always the first step of the process (Berg, 2015). It selects useful design information and combines them together. In order to facilitate the sequence planning process, a product model has to be represented in the computer based on the design information (Walther, 2010).

Disassembly sequence searching process is the second step of DSP. Planning disassembly sequence is a mathematical problem essentially (Viganò, 2013). For a complete disassembly operation, all components are required to be disassembled and an optimal disassembly sequence could be generated by the exhaustion search methods or traversal algorithms (Giri, 2015). However, for a selective disassembly operation, the disassembly operation will stop whenever the target component is disassembled (Riggs, 2015). For those products with the complex structure and a large number of components, disassembly sequence planning for a target component is an NP-hard problem essentially and the optimal solution is not likely to be achieved (Pomares, 2004). In this case, a heuristic algorithm has to be employed to find a near-optimal solution, namely the near-optimal disassembly sequence (Zhou, 2015). During the disassembly operation, constraint types between components are decided and all constraints are removed by non-destructive operations preferably. However, when it comes to certain constraints such as riveting constraint, welding constraint etc, a destructive disassembly method has to be used (Chen, 2014).

In this chapter, several commonly-used representation methods and corresponding sequence planning algorithms will be reviewed.

2.1 Product Representation

2.1.1 Graph-based Representation

AND/OR graph is a commonly used graph-based representation approach for product assembly and disassembly studies (Tseng, 2010). The graph consists of two parts: nodes and hyper-arcs. The nodes stand for subassemblies or components in a product, while the hyper-arcs stand for disassembly operations. An AND/OR graph employs a top-down approach in modeling disassembly processes. Theoretically, an AND/OR graph can represent all disassembly sequences (Song, 2010). However, for a product possessing a large amount of components, a combination explosion is more likely to happen and jeopardize the accuracy and efficiency of disassembly sequence planning. Figure 2-2 is an AND/OR graph for a product with 4 components (Li, 2014).

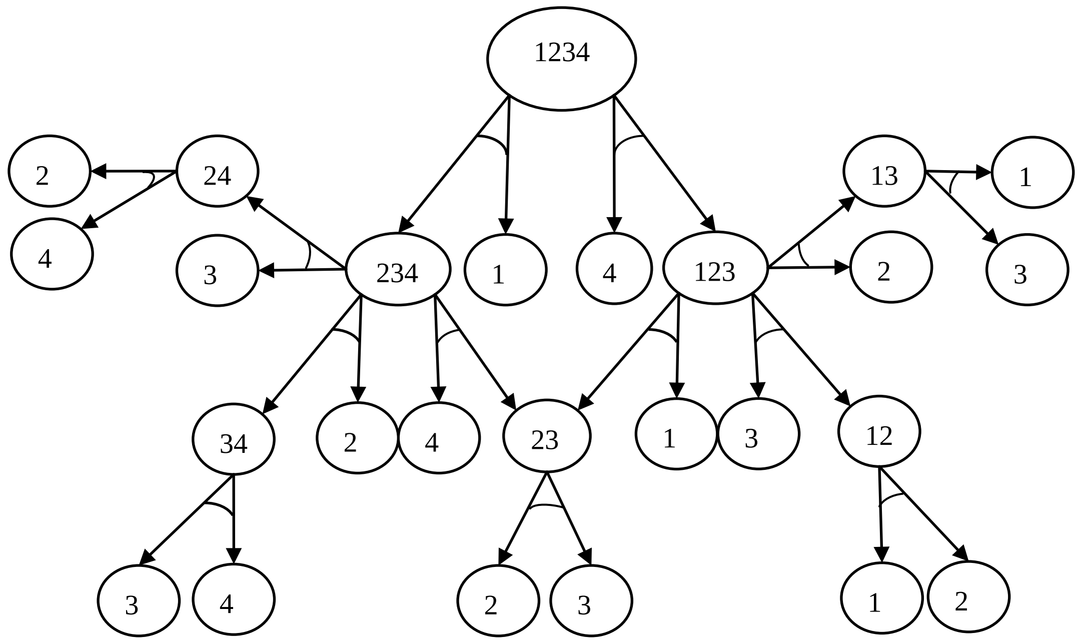
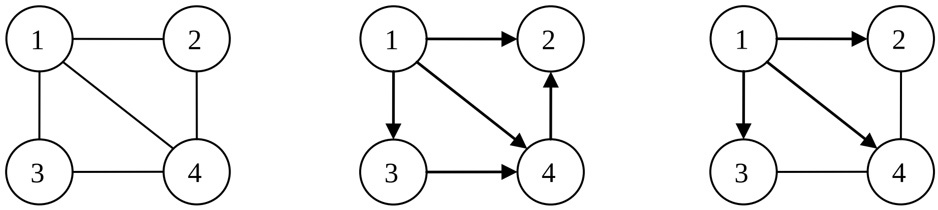


Figure 2-2 AND/OR graph

Zhou *et al.* constructed an AND/OR graph to identify the components stability information during disassembly sequence planning (Zhou, 2008). Koc *et al.* used the AND/OR graph to ensure the feasibility of the precedence relations among disassembly operations; hence the useable components were disassembled in a cost-effective manner (Koc, 2009). Min *et al*. used the AND/OR graph to solve maintenance and repairs problems of robots in hazardous environments such as outer space, under sea and nuclear power plants. With a clear view of the target component and toxic components, a selective disassembly sequence was generated without endangering operators' health (Min, 2010). Shana *et al.* proved the AND/OR graph was feasible and efficient in her assembly/disassembly analysis for green manufacturing (Shana, 2011). Furthermore, a weighted AND/OR graph is also proposed to better evaluate each disassembly sequence. The disassembly sequences with a high weight value are more likely to be performed than those disassembly sequences possessing the low weight value. With a distinct weight value attached to each task, the weighted AND/OR graph is more efficient in mechanical products' automatic disassembly decisions (Han, 2013).

Adjacent Graph is another visualized representation approach for disassembly sequence planning. It represents connection relations between adjacent components (Kai, 2014). The adjacent graph could be classified into three categories: undirected adjacent graph, directed adjacent graph and hybrid adjacent graph. Figure 2-3 shows these three adjacent graphs (Zhang, 2011).



(1)Undirected adjacent graph (2) Directed adjacent graph (3) Hybrid adjacent graph

Figure 2-3 Adjacent Graphs

As shown in Figure 2-3, nodes with numbers inside stand for components, segments/arrows between components stand for connection relations. For example, in the undirected adjacent graph (Figure 2-3 (1)), components 1&2, 1&3, 1&4, 2&4, and 3&4 are connected with segments. It means those pairs of components are physically connected to each other. In the directed adjacent graph (Figure 2-3 (2)), those segments are all substituted by arrows. Direction of an arrow is decided by the disassembly sequence precedence, disassembly sequences between each pair of components is pre-defined compulsively, which is from a start point to the end point of the arrow (Rickli, 2014). For example, in the directed adjacent graph Figure 2-3 (2), component 1 has to be removed before components 2, 3 and 4. Hybrid adjacent graph (Figure 2-3 (3)) is the combination of undirected and directed adjacent graphs (Liu, 2014). In a hybrid adjacent graph, only when two components has a distinctive disassembly precedence relationship will be connected with arrow, otherwise, adjacent components are only jointed with the directionless segment (Chen, 2014).

2.1.2 Constraint Matrix

Constraint matrix is a frequently-used representation approach for DSP. It can store products' structure information, components' connections as well as constraint relations in a matrix-based way. Based on the information included, a constraint matrix has different types (Elsayed, 2012). For example, Kalayci *et al.* constructed a global constraint matrix to show constraints relations between any pair of components, while Tian *et al.* only consider constraints' type & quantity between adjacent components (Kalayci, 2013; Tian, 2013). In addition to constraint relations between components, Behdad *et al.* recorded disassembly directions and tools in an expanded matrix (Behdad, 2012).

In a real disassembly operation, components are mainly removed by translation movements while some components, such as screws and bolts, are removed by rotation movements along with their central axes (Mascel, 2003; Pornsing, 2014). In disassembly sequence planning, all movements are regarded as the translation movement (Walther, 2010). Therefore, in a Cartesian coordinate system, normally only three DOFs are considered in six disassembly directions: $\pm X, \pm Y, \pm Z$ as shown in Figure 2-4 (Zhu, 2013).

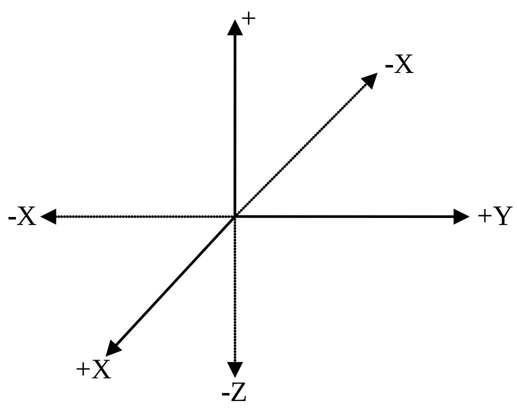


Figure 2-4 Disassembly directions in the Cartesian coordinate system

The number of disassembly directions included in a constraint matrix is determined by disassembly requirements and product's geometrical shape (Smith, 2012). For example, a bearing can only move along the axis of a shaft. If it is assumed that the axis of the shaft is X axis, there will be only two directions to be taken into consideration: +X and –X (Liu, 2014). As indicated by its name, the constraint matrix stores all data in the format of a matrix. Therefore, a comprehensive constraint matrix with all disassembly directions considered may have the problem of redundant storage space and long searching time (Park, 2003). Therefore, a well-constructed constraint matrix with the proper information included is the premise of generating an effective and economical disassembly sequence (Jiang, 2013).

Li et al. used three matrices to represent the constraint relations in +x, +y, +z and all elements in matrices are either 0 or 1. According to this method, 1 means that there is a constraint relation between two components and 0 means no constraint relations. If all elements in one row are 0s, the component is constraint free in the corresponding direction and could be removed. Otherwise, the component is constrained by other components and could not be disassembled. In addition, all elements are checked in a column to identify the constraint relations in -x, -y, -z directions (Li, 2011).

Liu et al. also used three matrices to represent constraint relations between each pair of components. But constraint relations are classified into three types and recorded into matrices. 0 means that there is no constraint relation; 1 represent that the two components are only adjacent and they are not connected by any fasteners; 2 means that the two components are connected by fasteners or similar forces and they should be disassembled by tools (Liu, 2012).

2.2 Sequence Planning Approach

2.2.1 DSP based on the graph representation

As introduced before, AND/OR graphs can represent all disassembly sequences for a product DSP. It starts from a complete product and ends up with individual components by decomposing the assembly using AND/ OR relations defined (Cappelli, 2007). In general, building an AND/OR graph needs to understand the inter-relations among components (Li, 2006). Although an AND/OR graph has the sound theoretical foundation, it is not practical to handle complex products. For example, a simple assembly with 14 components possesses 16383 nodes in an AND/OR graph (Homem de Mello, 1990). Obviously, this is inflexible and time-costly in terms of the computation. Disassembly Petri Net (DPN) is a variant of AND/OR graphs. It takes into account the dynamic aspect of disassembly operations using a token approach. However, it still results in the huge computation and can only solve complete disassembly problems (Li, 2006).

Adjacent graphs can generate disassembly sequences by dismantling free components from constraint relations (Zhang, 2015). These free components are accessible in the direction of contacts linked with other components and can be removed successively until the target component becomes constraint-free. Although adjacent graphs represent the entire constraint of a product, its search strategy can be used to plan selective disassembly sequences for a target component (Zhu, 2013).

Kuo proposed a non-directed graph-based heuristic approach to generate the disassembly sequence for recycling. A product is modeled by a component-fastener graph. By identifying the "cut-vertices", the searching operation decomposes the graph into sub-graphs until a disassembly tree is formed. Based on the disassembly tree, disassembly sequences can be generated (Kuo, 2000). Although disassembly sequences could be generated by Kuo's method, computation efforts are still very heavy because all possible disassembly sequences are represented by a disassembly tree. In order to reduce computation efforts and avoid generating all disassembly sequences, Murayama presented a search procedure to generate an AND/OR graph representation for the disassembly of a target component (Murayama, 2011). A product was represented as a liaison graph with nodes and arcs representing components and the connective relations between pairs of components respectively. Information entropy in relation to the disassembly of the target component was used to evaluate the feasibility of each disassembly step.

In conclusion, even though the graph-based approaches are straight forward and explicit for products structure and connections, the searching processes are computationally complex and very likely to cause combination explosion problems (Mitrouchev, 2015). Therefore, they are rarely used to handle disassembly sequence planning problems solely. They are more frequently employed together with other approaches (Smith, 2012).

2.2.2 DSP methods based on the matrix representation

Matrix can be used to represent products' geometrical connections and constraints in a mathematical-friendly way. Compared to the graph-based representations, matrices are more convenient and flexible in terms of computation (Zhang, 2006).

Using the matrix representation of a product, for a complete disassembly, the traversal algorithm can search all complete disassembly sequence and an optimal disassembly sequence could be generated by pre-defined evaluation criteria such as disassembly time and cost (Federico, 2008). For selective disassembly sequence planning, the traversal algorithm also can generate all disassembly sequence ended by the target component and an optimal disassembly sequence can then be searched based on the matrix representation (Han, 2013).

In DSP, an optimal disassembly sequence is an ideal solution for the minimum disassembly cost or the shortest disassembly time with the highest disassembly efficiency (Rickli, 2014). However, when it comes to a product with a large number of components and complex structure, it will take a large amount of calculation time and computation efforts to search the matrix and generate the sequences, especially for the selective disassembly operation (Smith, 2011). In addition, the combination explosion is more likely to happen in this situation therefore a sequence planner will not be able to achieve the optimal disassembly sequence (Elsayed, 2012). In this situation, a heuristic algorithm has to be employed to search the matrix and generate a near-optimal solution instead of the optimal one (Yeh, 2013).

A heuristic algorithm usually employs the disassembly cost and time etc. as criteria to select the near-optimal disassembly sequence (Giudice, 2010). It can reduce the searching efforts of the matrix. Genetic algorithm (GA), simulated annealing algorithm (SAA) and ant colony algorithm (ACA) are all commonly used approaches to generate near-optimal disassembly sequences (Go, 2012; Kalayci, 2013).

In the field of artificial intelligence, GA is a heuristic search algorithm that mimics the process of the natural selection which is chromosome with a lower fitness replaced by the chromosome with the higher fitness. The whole population evolves towards a better adaption to the natural selection (Sharma, 2015). In DSP research, each disassembly sequence should be coded as one chromosome and evaluated by the pre-defined fitness objective function (Lu, 2006). After generations of evolutions, the disassembly sequences will undergo the natural selection, crossover and mutation procedures. Finally, the disassembly sequence with the highest fitness is defined as the near-optimal disassembly sequence. Li successfully used GA for disassembly sequence planning of a traditional model: the electric torch. The result was proved to be feasible and also in accordance with manual disassembly planning (Li, 2002). Maroua *et al.* also used GA to generate a near-optimal disassembly sequence for the complete disassembly operation. The method takes into account several criteria such as maintainability of components, part volume, tools change and the change of disassembly directions (Maroua, 2014).

The concept of simulated annealing comes from an analogy with metallurgical annealing in which a piece of metal is initially heated to a high temperature, then cools down slowly to a low temperature (Hassanzadeh, 2012). Xie et al. develop a new algorithm by making a combination of the simulated annealing algorithm (SAA) and genetic algorithm (GA). This simulated annealing and genetic algorithm (SAGA) solved the common combination explosion problem in GA and premature phenomena problem in SAA (Xie, 2007).

Ant colony algorithm was initially proposed by Doctor Marco Dorigo in 1992. This algorithm mimics the behavior of ants seeking paths between their habitats and food (Wang, 2014). In the beginning, all ants head to different direction randomly and when one ant finds food it will release pheromone to attract other ants. Finally, more and more ants will find the short path to the food. In disassembly sequence planning research, those paths represent disassembly sequences and a short distance of path means the disassembly cost is low. Seamus has successfully used this method to generate the disassembly sequence with multiple target components (Seamus, 2004).

2.2.3 Methods comparison and problems conclusion

In conclusion, all disassembly sequence planning methods discussed above have their own advantages. Graph-based DSP methods are straight forward and explicit visually (Zhang, 2011). Especially in the selective disassembly, it can show the hierarchical position of the target component and corresponding components with constraint relations (Rickli, 2014). Traversal algorithms could provide an optimal disassembly sequence with the minimum disassembly cost for both selective disassembly and complete disassembly operations (Zhang, 2013). Heuristic algorithms such as GA and ACA are all helped to generate the near-optimal disassembly sequences based their own objective functions (Go, 2011; Xing, 2012).

However, those disassembly sequence planning methods all considered how to remove constraints and get components in a non-destructive manner and few of them took the destructive disassembly into consideration. During the disassembly operation of an End-of-Life product, many components are useless and could be destroyed. In this case, destructive disassembly operations can help to find a short path to the target component (Song, 2013). In addition, when it comes to those constraints such as riveting and welding, non-destructive disassembly operations have to bypass them but destructive disassembly operations could remove them (Chen, 2014). However, if the selective disassembly operation is performed for maintenance, all disassembly operations should be non-destructive manner (Pomares, 2004). In this thesis, the destructive disassembly will be considered in selective disassembly for EOL products and a comparison will be made between non-destructive manner and destructive manner based on their disassembly costs.

Moreover, the existing representation methods fail to incorporate products' bill of material (BOM) information. One constraint matrix with all components has the problem of redundant information and could not show the hierarchical relations (Pornsing, 2014). In this thesis, product models will be represented based on their BOMs. In the selective disassembly operation, this representation method can help to find the target component's hierarchical position and only its father subassemblies will be disassembled. Therefore, compared to one constraint matrix, its searching size is smaller and hence the efficiency is improved.

Furthermore, when using GA for disassembly sequence planning, researchers use part numbers as the code number for chromosomes instead of using binary string (Li, 2011). For example, chromosome $4-5-2-1-3$ means the corresponding disassembly sequence is $part4-part5-part2-part1-part3$ (Zhang, 2011). This coding method is simple and straight forward. But the disadvantage of this method is obvious. Those part number strings (disassembly sequences) will go through crossover and mutation operations, newly-generated disassembly sequences may not be feasible. In this thesis, all newly-generated disassembly sequences will go through a disassembly feasibility check, only those feasible sequences will go to the next generation.

In conclusion, based on the existing methods reviewed, three problems are proposed for the thesis work to solve:

(1) Integration of BOM in product's representation;

(2) Apply disassembly feasibility check in the GA;

(3) Consideration of destructive disassembly for EOL products.

2.3 Bill of Material

As mentioned before, without considering product's BOM, components' hierarchical relations are unidentified. Hence the disassembly complexity and workload will be increased significantly. In product design and manufacturing stages, the bill of material can be used to represent product's hierarchical structure (Shih, 2014). In addition, it is a list of raw materials, subassemblies, intermediate assemblies, subcomponents, parts and the quantities of each part needed to form an end-product (Luca, 2010).

In most cases, BOM is hierarchical with the top level representing an end-product and the low levels representing subassemblies or components. The structure of BOM fully depends on product design information and design requirements (Kashkoush, 2014).

In general, during the product assembly process, individual components are firstly grouped into small subassemblies. These small subassemblies with the similar assembly property or assembly requirements are then combined into higher level subassemblies. At last, according to BOM, an end product is formed by several large subassemblies which are also called first-level subassemblies. An example of BOM is shown in Figure 2-5 (Min, 2013).

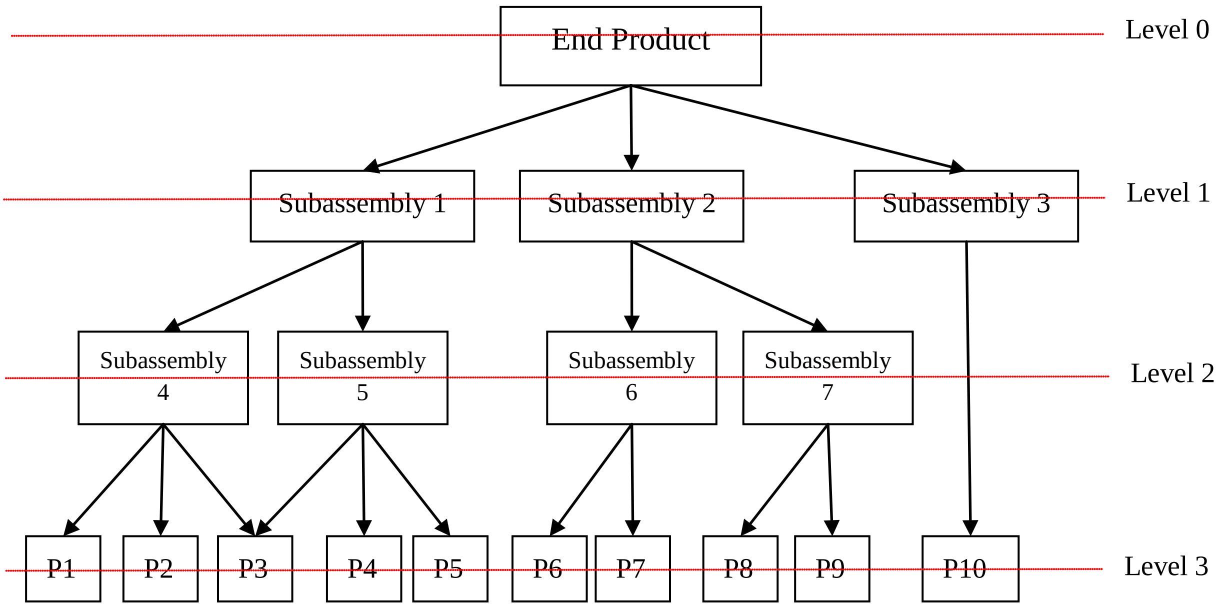


Figure 2-5 Example of BOM

For example in Figure 2-5, this end product (level 0) has three first-level (FS) subassemblies (level 1), namely subassembly 1, subassembly 2 and subassembly 3. Similarly, in the level 2, subassembly 4 and subassembly 5 are grouped into their father subassembly 1 while subassembly 6 and subassembly 7 are combined together to form their father subassembly 2. In the last level, level 3, individual components are all basic elements.

Therefore, for this end product, we can conclude that there are four layers of subassemblies/components (level 0, level 1, level 2 and level 3) and three layers composition relations (level 0-level 1, level 1-level 2 and level 2-level 3) in its bill of material.

2.4 Summary

This chapter discusses three commonly-used product representation approaches as well as four types of sequence planning algorithms. They all worked well and proved to be feasible in their own cases. However, the destructive disassembly is rarely considered in selective disassembly and BOM information is missed in products' representation stage. This thesis will solve above problems.

Chapter 3

DSP based on multi-layer constraint matrices

This chapter introduces a disassembly sequence planning (DSP) method based on an efficient product representation. An improved traversal algorithm is used to search all disassembly sequences. Total disassembly time is the evaluation criterion to generate an optimal disassembly sequence.

3.1 Introduction

General procedures of the proposed method are similar to what is shown in Figure 2-1. In this chapter, the target component is removed by a non-destructive disassembly apporach. The method consists of three main parts, namely product representation, disassembly sequence searching, and disassembly sequence evaluation.

As mentioned in Chapter two, in order to improve the DSP efficiency, the size of a product's representation model has to be manageable. For this intention, multi-layer constraint matrices are constructed to represent product's constraints based on the product bill of material (BOM). In addition, an improved traversal algorithm is proposed to find an optimal disassembly sequence with the minimum complexity. Searching size will also be reduced along with the searching process.

All feasible disassembly sequences will be evaluated using the total disassembly time. The optimal disassembly sequence is the sequence with the least disassembly time. In this chapter, the total disassembly time includes the time used to remove components out of a product, and the time spent in the disassembly direction re-orientation (Cao, 2007).

3.2 Multi-layer constraint matrix

Constraint stops components removing out of a product. Previous studies on representations of products either construct only one-dimension constraint matrix or simply consider the constraint relations between adjacent components (Luo, 2014; Li, 2008). Those types of constraint matrices may solve problems in certain types of product models, but they are not versatile. For example, one-dimension constraint matrix cannot show products with complicated geometric structure. In order to solve the above-mentioned problems, a multi-layer constraint matrix method is proposed based on product's BOM.

A multi-layer constraint matrix represents constraints between one component and other components along $\pm X, \pm Y, \pm Z$ directions in Cartesian Coordinates of a product 3D model. For an assembly A= {A_1, A_2… A_n} with n components, the corresponding constraint matrix is shown in Figure 3-1.

$$
\begin{array}{cccc}
a_{11x}a_{11-x}a_{11y}a_{11-y}a_{11z}a_{11-z} & a_{12x}a_{12-x}a_{12y}a_{12-y}a_{12z}a_{12-z} & \cdots & a_{1nx}a_{1n-x}a_{1ny}a_{1n-y}a_{1nz}a_{1n-z} \\
a_{21x}a_{21-x}a_{21y}a_{21-y}a_{21z}a_{21-z} & a_{22x}a_{22-x}a_{22y}a_{22-y}a_{22z}a_{22-z} & \cdots & a_{2nx}a_{2n-x}a_{2ny}a_{2n-y}a_{2nz}a_{2n-z} \\
. & . & . & . \\
. & . & . & . \\
. & . & . & . \\
a_{n1x}a_{n1-x}a_{n1y}a_{n1-y}a_{n1z}a_{n1-z} & a_{n2x}a_{n2-x}a_{n2y}a_{n2-y}a_{n2z}a_{n2-z} & \cdots & a_{nnx}a_{nn-x}a_{nny}a_{nn-y}a_{nnz}a_{nn-z}
\end{array}
$$

Figure 3-1 Example of constraint Matrix

This is a $n \times n$ matrix and each element in this constraint matrix is a 6-digital array. Element a_{ijd} represents constraint relations between component *i* and component *j* along direction *d*.

$i, j \in (1, 2, ... n); d \in (\pm X, \pm Y, \pm Z)$

Where a_{ijd} =1 means component *j* stops component *i* moving out of the product along direction *d*.

a_{ijd} =0 means component *j* does not stop component *i* moving out of the product along direction *d*.

When *i*=*j*, the value of a_{ijd} equals to 0 regardless any direction. Therefore all elements along diagonal of the constraint matrix are 000000.

A simplified gear box is shown in Figure 3-2. It is used to illustrate this representation method.

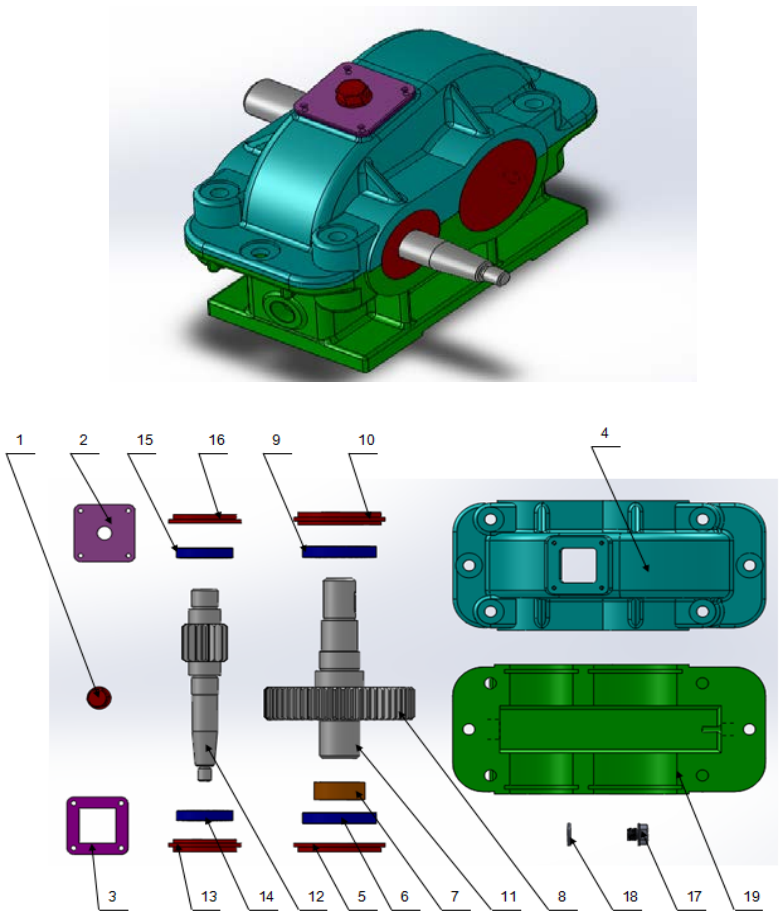


Figure 3-2 A simplified Gear box model

As shown in Figure 3-2, this simplified gear box contains 19 components. Detail information of each component is shown in Table 3-1.

Part No.	Part name	Subassembly No.	Part No.	Part name	Subassembly No.
1	Vent Hood	FS1	11	Transmission Axis	FS2
2	Upper Cover	FS1	12	Gear Axis	FS3
3	Upper Shim	FS1	13	Cover 3	FS3
4	Upper Body	FS1	14	Bearing 3	FS3
5	Cover 1	FS2	15	Bearing 4	FS3
6	Bearing 1	FS2	16	Cover 4	FS3
7	Retainer Ring	FS2	17	Oil-level Pointer	FS4
8	Gear 1	FS2	18	Pointer shim	FS4
9	Bearing 2	FS2	19	Lower Body	FS4
10	Cover 2	FS2			

Table 3-1 Parts of the Gear Box

The bill of material (BOM) of this gear box is shown in Figure 3-3. Based on the BOM, four subassemblies can be formed, namely upper body (FS1), first transmission axis (FS2), second transmission axis (FS3), and lower body (FS4) on Level 1. Their details are shown in Figure 3-4.

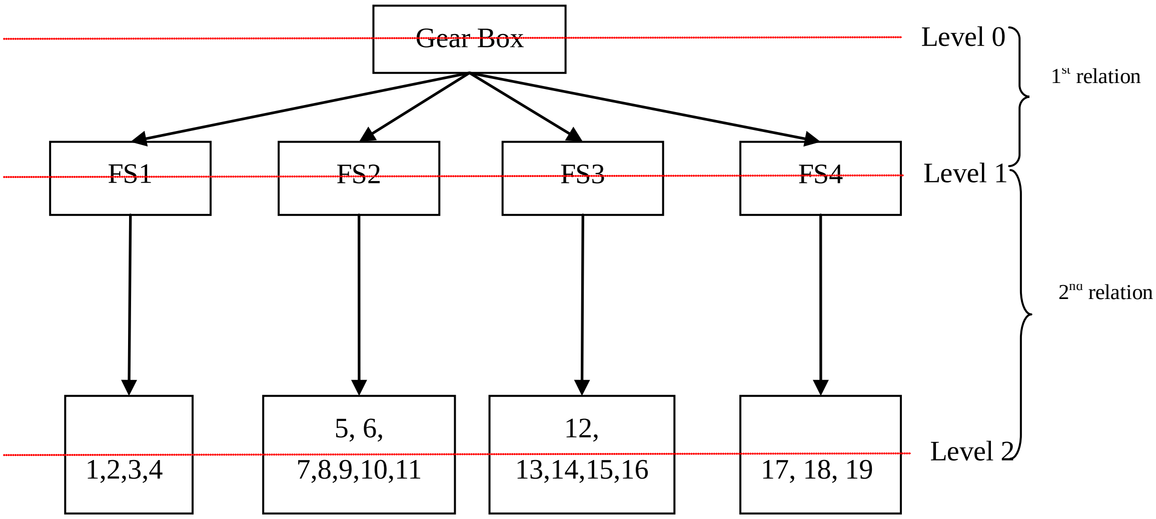


Figure 3-3 BOM of the Gear Box

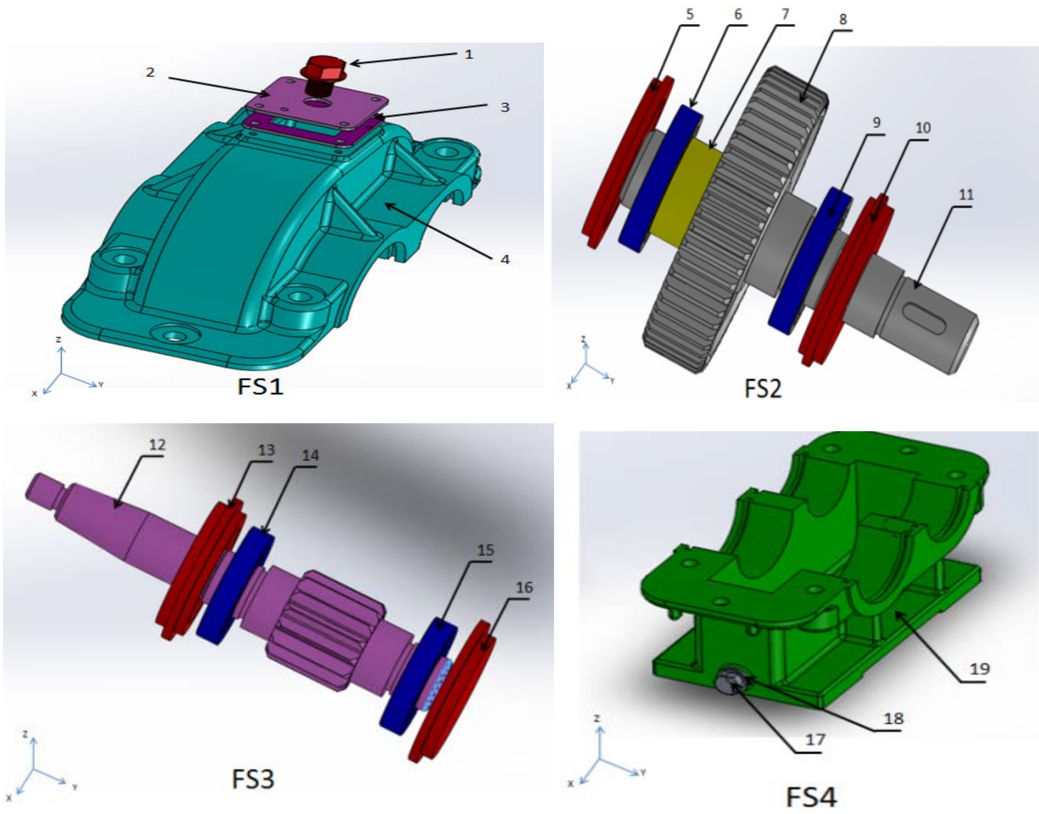


Figure 3-4 Four subassemblies

According to the bill of material (BOM) in Figure 3-3, the gear box possesses three layers. The first and second layers are the gear box and subassemblies (FS1, FS2, FS3 and FS4), respectively.

The third layer is constructed by components. In other words, the gear box consists of four subassemblies in the first composition relation, namely upper body (FS1), first transmission axis (FS2), second transmission axis (FS3), and lower body (FS4). In the second composition relation, there are 4, 7, 5, and 3 components in subassembly FS1, FS2, FS3 and FS4, respectively. Therefore, totally 5 (5=1+4) constraint matrices are used to represent the gear box.

The first-level constraint matrix with four subassemblies is built as shown in Figure 3-5.

$$
\begin{array}{c@{\;}c}
 & \begin{array}{cccc} FS1 & FS2 & FS3 & FS4 \end{array} \\
\begin{array}{c} FS1 \\ FS2 \\ FS3 \\ FS4 \end{array} &
\begin{pmatrix}
000000 & 111101 & 111101 & 000001 \\
111110 & 000000 & 011100 & 111101 \\
111110 & 101100 & 000000 & 111101 \\
000010 & 111110 & 111110 & 000000
\end{pmatrix}
\end{array}
$$

Figure 3-5 Constraint matrix for the first-level

Similarly, four second-level constraint matrices with components are shown in Figure 3-6.

$$
\begin{array}{c@{\;}c}
 & \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\
\begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} &
\begin{pmatrix}
000000 & 111101 & 111101 & 111101 \\
111110 & 000000 & 000001 & 000001 \\
111110 & 000010 & 000000 & 000001 \\
111110 & 000010 & 000010 & 000000
\end{pmatrix}
\end{array}
$$

a. Constraint matrix of FS1

$$
\begin{array}{c}
\begin{array}{cccccccc}
 & 5 & 6 & 7 & 8 & 9 & 10 & 11
\end{array}\\
\begin{array}{c}
5\\6\\7\\8\\9\\10\\11
\end{array}
\begin{pmatrix}
000000 & 001000 & 001000 & 001000 & 001000 & 001000 & 111011\\
000100 & 000000 & 001000 & 001000 & 001000 & 001000 & 110011\\
000100 & 000100 & 000000 & 001000 & 001000 & 000100 & 110011\\
000100 & 000100 & 000100 & 000000 & 001000 & 000100 & 110011\\
000100 & 000100 & 000100 & 000100 & 000000 & 001000 & 110011\\
000100 & 000100 & 000100 & 000100 & 000100 & 000000 & 110011\\
110111 & 110011 & 110011 & 110011 & 110011 & 110011 & 000000
\end{pmatrix}
\end{array}
$$

b. Constraint matrix of FS2

$$
\begin{array}{c}
\begin{array}{cccccc}
 & 12 & 13 & 14 & 15 & 16
\end{array}\\
\begin{array}{c}
12\\13\\14\\15\\16
\end{array}
\begin{pmatrix}
000000 & 110111 & 110111 & 111011 & 111011\\
111011 & 000000 & 001000 & 001000 & 001000\\
111011 & 000100 & 000000 & 001000 & 001000\\
110111 & 000100 & 000100 & 000000 & 001000\\
110111 & 000100 & 000100 & 000100 & 000000
\end{pmatrix}
\end{array}
$$

c. Constraint matrix of FS3

$$
\begin{array}{c}
\begin{array}{cccc}
 & 17 & 18 & 19
\end{array}\\
\begin{array}{c}
17\\18\\19
\end{array}
\begin{pmatrix}
000000 & 011111 & 011111\\
101111 & 000000 & 010000\\
101111 & 100000 & 000000
\end{pmatrix}
\end{array}
$$

d. Constraint Matrix of FS4

Figure 3-6 Constraint matrices of second level

In conclusion, using a one-layer constraint matrix, there are $19\times19=361$ elements to be recorded for the gear box. However, by using multiple-layer constraint matrices based on BOM, the gear box could be represented by one first-level constraint matrix and four second-level matrices. There are 4, 7, 5, and 3 components in subassembly FS1, FS2, FS3 and FS4, respectively. This gear box can be represented by five matrices (one 4×4 matrix, one 4×4 matrix, one 7×7 matrix, one 5×5 matrix and one 3×3 matrix). The total number of elements in those five matrices is $4\times4+4\times4+7\times7+5\times5+3\times3=115$. It is quite obvious that by adopting the multiple-layer constraint matrix, the total number of elements to be recorded is reduced by 68.1% ((361-115)/361).

Representation method	Number of elements
One-layer constraint matrix	$19\times19=361$
Multi-layer constraint matrix	$4\times4+4\times4+7\times7+5\times5+3\times3=115$

Table 3-2 Comparison between representation methods

3.3 Traversal algorithm for selective disassembly

3.3.1 Constraint matrix analysis

In the first step of disassembly operations, the first-level subassemblies will be removed as a whole. In order to remove one subassembly/component out of a product, at least one feasible disassembly direction is required. In other words, the subassembly/component must be disassembled along at least one direction without being blocked by other components. The formula to evaluate component *i* along direction $d\in(\pm X,\pm Y,\pm Z)$ is shown as follows: (n is the number of components in this subassembly)

$$a_{id}=a_{i1d}+a_{i2d}+\ldots+a_{ind} \tag{3.1}$$

Only when $a_{i1d}=a_{i2d}=\ldots=a_{ind}=0$, the value of a_{id} will be 0 and component *i* could be removed along direction *d*. Otherwise, we can say component *i* is temporarily fixed along direction *d*.

Take the constraint matrix for first-level subassemblies in Figure 3-5 as an example:

$$a_{1z} = a_{11z} + a_{12z} + a_{13z} + a_{14z} = 0 + 0 + 0 + 0 = 0 \qquad (3.2)$$

It means that subassembly FS1 is not constrained from moving along +Z direction, hence FS1 could be disassembled along +Z in this step. In addition, when subassembly FS1 is removed out of the gear box, it will not have constraints to the rest three subassemblies any more. Therefore, for the global constraint, elements in the first column are all updated to 000000. Since all elements in the first column are all 000000 and have no constraint to the following search process, they will be deleted. The first row is the constraint information for subassembly FS1, and it can also be deleted after FS1 is disassembled. Therefore, this 4×4 constraint matrix will be reduced into a 3×3 matrix. This step will reduce the searching size from 16 elements to 9 elements. Using a smaller matrix, the calculation complexity is reduced and the disassembly sequence planning efficiency can be improved. The original matrix and updated matrix is shown in Figure 3-7.

$$\begin{array}{ccccc} & \cancel{FS1} & FS2 & FS3 & FS4 \\ \cancel{FS1} & \cancel{000000} & \cancel{111101} & \cancel{111101} & \cancel{000001} \\ FS2 & \cancel{000000} & 000000 & 011100 & 111101 \\ FS3 & \cancel{000000} & 101100 & 000000 & 111101 \\ FS4 & \cancel{000000} & 111110 & 111110 & 000000 \end{array} \qquad \begin{array}{cccc} & FS2 & FS3 & FS4 \\ FS2 & 000000 & 011100 & 111101 \\ FS3 & 101100 & 000000 & 111101 \\ FS4 & 111110 & 111110 & 000000 \end{array}$$

Figure 3-7 Updated constraint matrix of gear box

3.3.2 Implementation of the traversal algorithm

The traversal algorithm can search all disassembly sequences and each sequence must be checked for its disassembly feasibility. Unfeasible disassembly sequences will be skipped and only feasible sequences are recorded. For example, if there are n components in a constraint matrix, the total number of sequences to be checked is $n!$. A general flowchart of the traversal searching algorithm is shown in Figure 3-8.

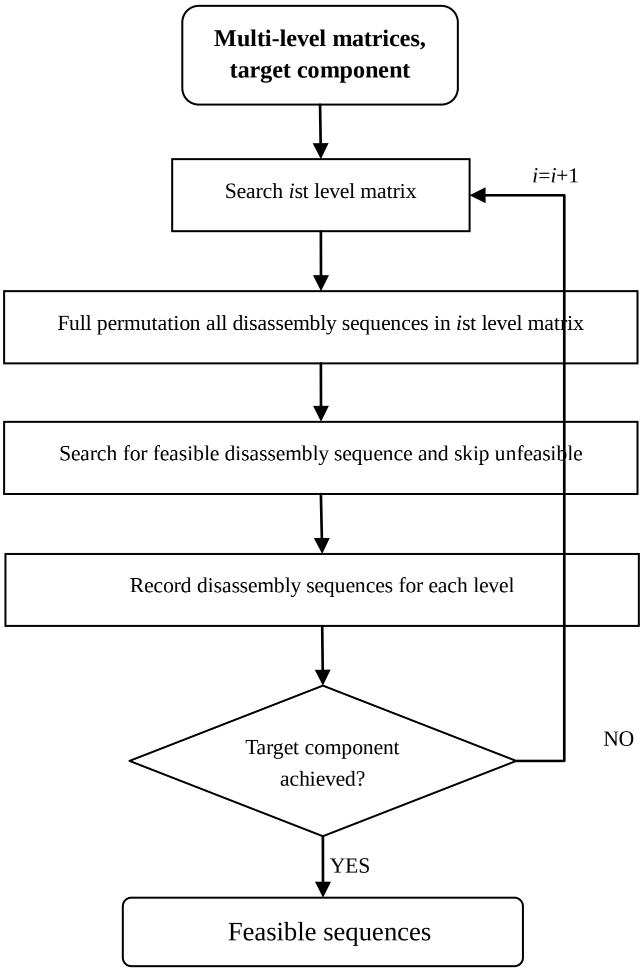


Figure 3-8 Flowchart of improved traversal algorithm

A product model is represented by multi-layer constraint matrices. The first level matrix represents the product and the second level matrices represent the first level subassemblies of the product. The other level matrices are constructed with the same procedure. If a target component is confirmed, the hierarchical positions of the target component in each level matrix can be identified.

As the product is disassembled into several first-level subassemblies first, the search process also starts from the first-level constraints matrix. Constraint matrix is constructed as shown in Figure 3-1. According to Formula 3.1, if $a_{id} = a_{i1d} + a_{i2d} + ... + a_{ind} = 0$, $d \in (\pm X, \pm Y, \pm Z)$, then the subassembly i will be able to move out of the product. After the subassembly is removed, the corresponding row and column of the matrix are deleted, and the matrix is simplified into a smaller size. If $a_{id} = a_{i1d} + a_{i2d} + ... + a_{ind} \neq 0$, it means the subassembly i cannot be removed temporarily. All sequences started with i will be skipped and they will not check for disassembly feasibility. When a disassembly sequence of first-layer subassembly is generated, the same method will search the lower-level matrices with the target component while the other matrices will be skipped. After the searching operation of each level matrices are finished, sequences of disassembling each level subassemblies will be combined together. Then all feasible disassembly sequences are generated. By evaluating all feasible disassembly sequences, an optimal sequence can be selected. A case study of the gear box in Figure 3-2 is used to verify this method.

For the gear box, if component 12 (gear axis) is to be replaced for maintenance. This is a selective disassembly problem in a non-destructive manner. Therefore, component 12 in the gearbox is selected as the target component. Based on its BOM, component 12 belongs to the first-level subassembly FS3. Therefore, subassembly FS3 has to be disassembled in the first step. After subassembly FS3 is removed from the gear box, a following disassembly operation should be performed to get the target component 12 in the lower-level matrix.

Detailed procedures of each step to disassemble the target component 12 are explained as follows:

1: Search feasible disassembly sequences in first-layer constraint matrix

The gear box shown in Figure 3-2 consists of 4 subassemblies in the first layer and there are total $4! = 4 \times 3 \times 2 \times 1 = 24$ permutations. However, some of these permutations are not feasible disassembly sequences, which will not be searched by the improved traversal algorithm. As subassembly FS3 contains the target component, the sequence searching process stops at FS3 in the first layer. In order to remove a subassembly out of the gear box, the subassembly has to be disassembled along one direction without any constraint at least. Formula 3-1 is used to decide whether a subassembly could be removed. Feasible sequences will be recorded and infeasible sequences will be skipped.

For example, subassembly FS2 is under searching for a feasible disassembly direction first. Based on Formula 3-1:

$$
\begin{aligned}
a_{2x} &= 1+0+0+1=2\neq 0 \\
a_{2-x} &= 1+0+1+1=3\neq 0 \\
a_{2y} &= 1+0+1+1=3\neq 0 \\
a_{2-y} &= 1+0+1+1=3\neq 0 \\
a_{2z} &= 1+0+0+0=1\neq 0 \\
a_{2-z} &= 0+0+0+1=1\neq 0
\end{aligned}
$$

Figure 3-9 Search subassembly FS2

Because all values in six directions are not equal to 0, the subassembly FS2 is constrained in the six disassembly directions. Under this circumstance, all sequences started from FS2 are regarded as infeasible sequences. In this way, many infeasible disassembly sequences will be skipped.

2. Searching feasible disassembly sequences in the next level constraint matrix

After searching the first level constraint, a similar searching process will be performed on the next level of the matrix. In the case of the gearbox, there are total two level constraint matrices and therefore subassembly FS2 will be searched for the target component in this step. The search process will stop when the target component is approached. The other three subassemblies will be skipped.

3. Evaluation of feasible disassembly sequences for the optimal one

The evaluation criterion is the total disassembly time including disassembly time and re-orientation time. Disassembly time for components is decided by their weight and geometric shape, etc. Re-orientation time is decided by the disassembly direction change. In disassembly operations, frequently changing disassembly directions is time-costing and energy-consuming. Therefore, it is preferably to disassemble as many components as possible in one disassembly direction before changing the disassembly direction.

The total disassembly time is defined as follows:

$$T_{total} = \sum_{i=1}^{m} T_{Components} + \sum_{j=1}^{n} T_{Rotation} \quad (3\text{-}3)$$

Where m is the total number of components removed; n is the total number of disassembly direction changes.

For example, if a disassembly direction for the first component is +X and the disassembly direction for the second component is –Z, the change of two disassembly directions is 90° in the Cartesian Coordinates System. A rotation table of direction changes for six axial directions is shown in Figure 3-10: (unit: °)

	$+X$	$-X$	$+Y$	$-Y$	$+Z$	$-Z$
$+X$	0	180	90	90	90	90
$-X$	180	0	90	90	90	90
$+Y$	90	90	0	180	90	90
$-Y$	90	90	180	0	90	90
$+Z$	90	90	90	90	0	180
$-Z$	90	90	90	90	180	0

Figure 3-10 Rotation in a Cartesian Coordinates System

Based on the multi-layer constraint matrices, a disassembly direction of each subassembly/component will be recorded along with the disassembly sequence search. In this chapter, it is assumed that the disassembly time for FS1, FS2, FS3 and FS4 is 10s, 20s, 30s and 40s, respectively. It is also assumed that in the example of the gearbox, each rotation of 90° will cost 2 seconds and 4 seconds for a rotation of 180° (Min, 2010).

Above search process is implemented using MATLAB software. Figure 3-11 shows the user interface of this searching program.

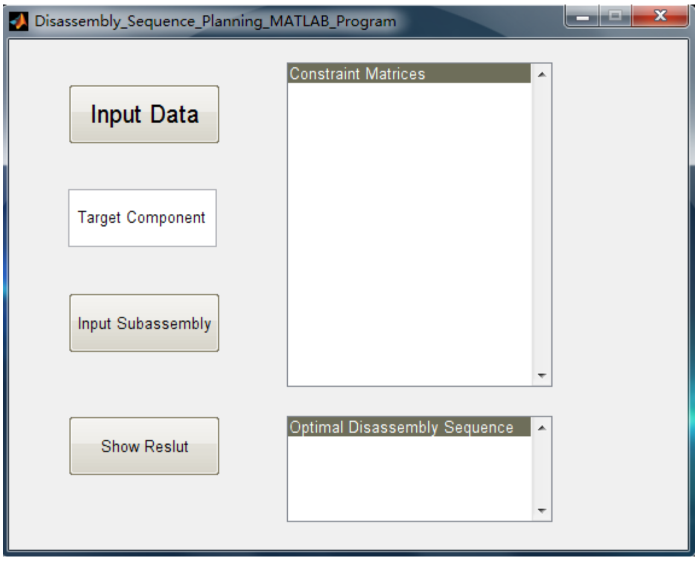


Figure 3-11 User interface of the search program

In the first step, click on the “Input Data” button and the constraint matrix of the gear box in Figure 3-5 is input into the MATLAB as shown in Figure 3-12.

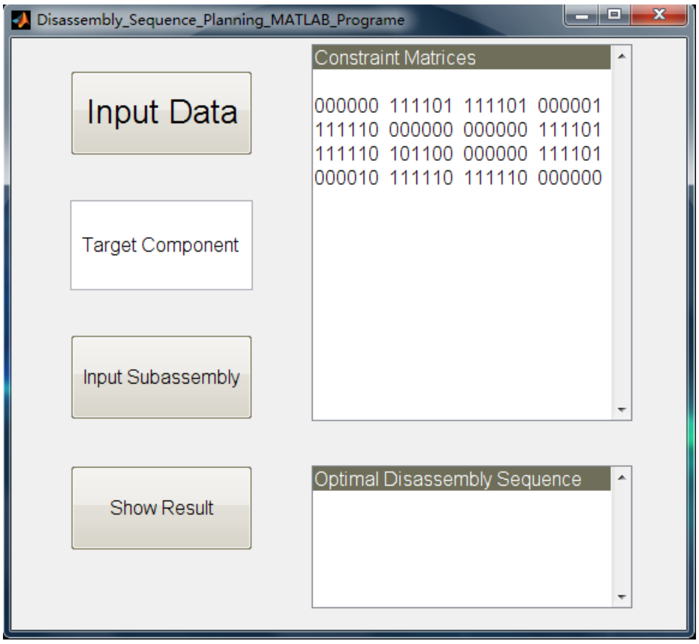


Figure 3-12 User interface for inputting constraint matrix

After inputting component 12 in the "Target Component" box and clicking on the "Show Result" button, the program will identify subassembly FS3 containing target component 12 and the optimal disassembly sequence to FS3 is generated based on searching process and evaluation criteria introduced above. Disassembly operation will stop when the target subassembly FS3 is achieved. The optimal sequence is: FS1 (+Z) —FS3 (+Z) in this layer. As the disassembly direction is not changed, the re-orientation time is 0 second. The total disassembly time is 10s+30s+0s=40s in this layer.

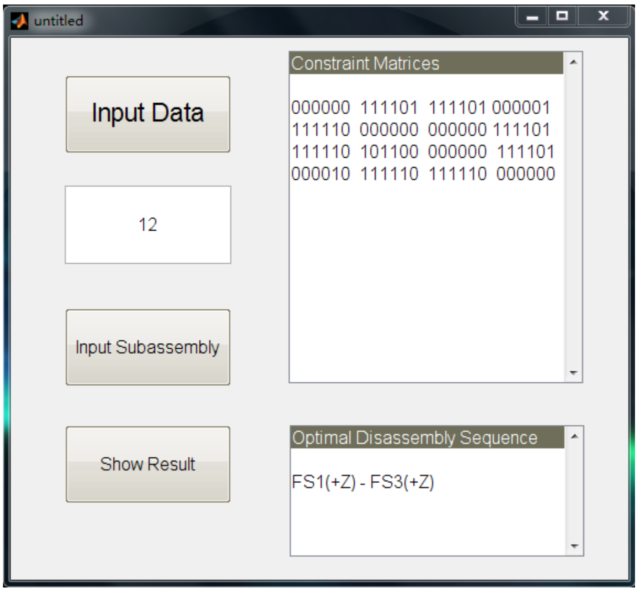


Figure 3-13 Optimal disassembly sequence search for subassembly FS3

After the first-level subassembly FS3 is disassembled, following disassembly operations will be performed only within subassembly FS3. Other three first-level subassemblies (FS1, FS2 and FS4) are skipped. Therefore, there are 5 components in the subassembly FS3 to be considered

Disassembly time required for each component in FS3 is given as Table 3-3. Criteria of disassembly time for different components will be introduced in Chapter 4.

Component No.	**Disassembly Time**
12	13s
13	9s
14	16s
15	7s
16	10s

Table 3-3 Disassembly time for components

Click on the “Input Subassembly” button and constraint matrix of FS3 is input into MATLAB.

Using the same traversal algorithm and evaluation criterion, it is found that the target component 12 is constrained by all components in this subassembly FS3. In other words, components 13, 14, 15 and 16 have to be removed before reaching the target component 12. Finally, two optimal disassembly sequences to disassemble component 12 are generated as follows.

(1) 13(-Y)-14(-Y)-16(+Y)-15(+Y)-12

(2) 16(+Y)-15(+Y)-13(-Y)-14(-Y)-12

In the first disassembly sequence, components 13 and 14 are removed along –Y direction and components 16 and 15 are removed along +Y direction. After those four components are disassembled, component 12 can be reached. Therefore, the total rotation direction is only 180° which is from –Y direction to +Y direction. For the second disassembly sequence, the disassembly direction re-orientation is also 180° which is from +Y to –Y.

Therefore, the optimal disassembly sequences for the target component 12 are generated by combining the two disassembly operations. Click on the "Show Result" button, the final optimal disassembly sequences are shown as Figure 3-14. The total disassembly time is 40s+55s+4s=99s.

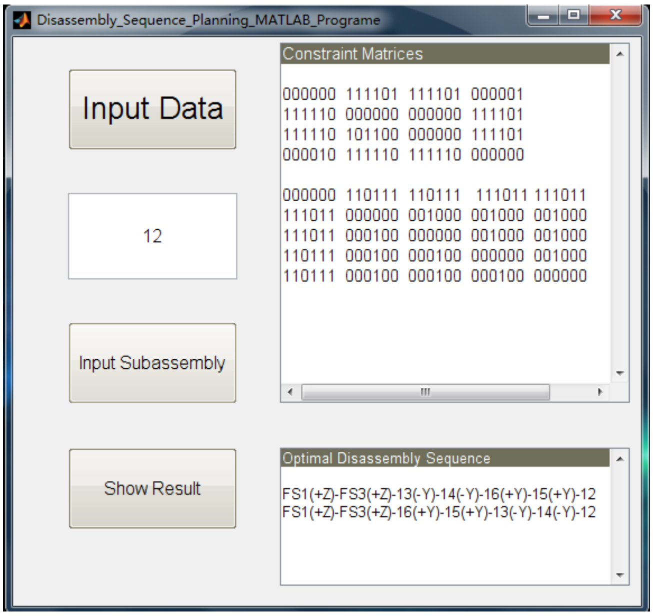


Figure 3-14 Output optimal disassembly sequences

The traversal algorithm skips infeasible sequences to minimize the searching size. A comparison between previous traversal algorithm and the improved traversal algorithm is shown in Table 3-4:

Traversal algorithms	Number of sequences searched
Traversal algorithm with one-layer matrix	$19! = 121645100408832000$
Traversal algorithm with multi-layer matrix	$4!+5!=144$

Table 3-4 Comparison between Searching methods

3.4 Conclusions

In this chapter, based on product's bill of material (BOM), multiple-level constraint matrices are built and a traversal algorithm is proposed to search disassembly sequences. The case study of the gear box in this chapter is a selective disassembly problem for product maintenance, all disassembly operations are performed in a non-destructive manner. Disassembly time for components and re-orientation ensures that the generated sequence is optimal and economical. The search size and calculation complexity of planning is reduced by using multi-layer constraint matrices and the proposed traversal algorithm.

However, the traversing algorithm is not efficient enough. Many infeasible disassembly sequences may be searched and generated. In Chapter 4, the genetic algorithm will be introduced to increase the searching efficiency.

Chapter 4

DSP using genetic algorithm

As introduced in Chapter 2, selective disassembly operations aim to get the target component with a reduced number of components to be disassembled. In Chapter 3, the traversal algorithm is used to find an optimal disassembly sequence. However, many infeasible disassembly sequences are searched and generated. In this chapter, a genetic algorithm (GA) is used to generate a near-optimal disassembly sequence with more efficiency by skipping many infeasible sequences. Furthermore, disassembly feasibility check is implemented after the crossover and mutation operation, it can generate a better solution with a shorter time. An example of camshaft will be used to compare the GA with and without the disassembly feasibility check.

4.1 Introduction

Genetic algorithm was proposed based on Darwin's theory of the evolution, which can be briefly described as follows: the algorithm starts with a set of randomly selected solutions called population. Each member of the population is encoded as an artificial chromosome. And each chromosome will be assigned a fitness score based on a predefined fitness function (Haichao, 2015). During the evolution process, a new population of chromosome is created iteratively for finding a chromosome with a better fitness, namely a high fitness score (Pachauri, 2015). At each generation of the evolution process, a mutation may occur in a chromosome, or two chromosomes may mate to produce a child which is known as crossover. Those chromosomes with higher scores have more chance to be selected (Saranya, 2015). The process is iterated until some predetermined objectives are achieved (Zhang, 2015).

In this thesis, each disassembly sequence is regarded as a chromosome and a disassembly sequence with less disassembly cost means a higher possibility to be selected. A general flowchart of this process is shown in Figure 4-1.

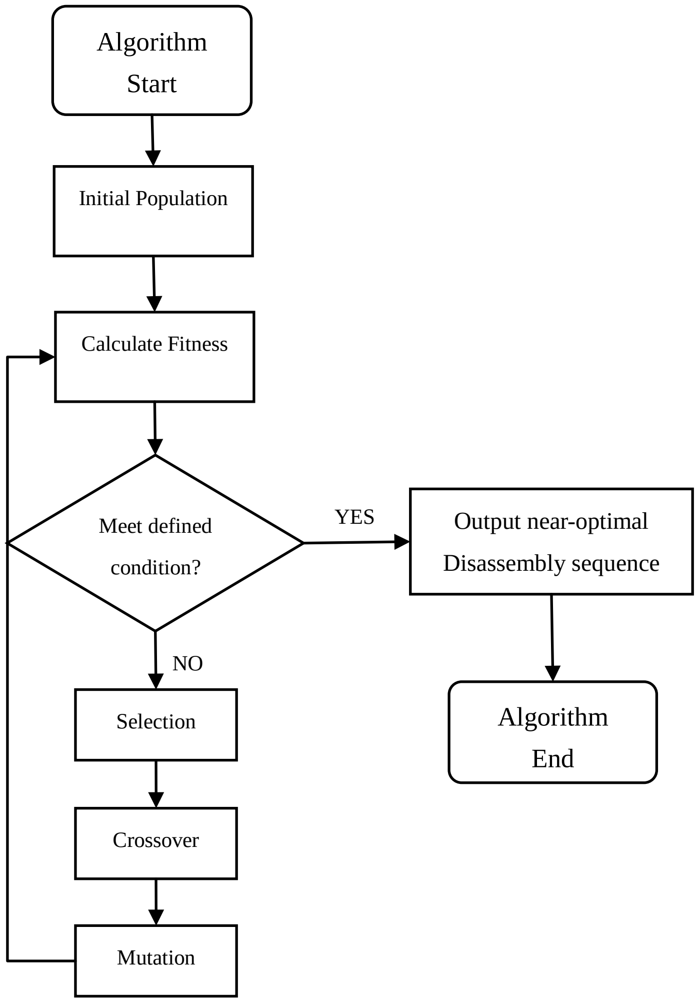


Figure 4-1 Flowchart of the genetic algorithm for DSP

In the genetic algorithm for DSP, disassembly sequences will be evaluated by a predefined fitness function and the algorithm will stop when a predefined criterion is achieved. Those disassembly sequences will go through three genetic operations: selection, crossover and mutation. The purpose of those genetic operations is to create chromosomes, namely new disassembly sequences. Details of this genetic algorithm are discussed in the following subsections.

4.2 Parameters in the genetic algorithm for DSP

(1) Coding chromosomes

Coding for chromosome is the first and most important step in the genetic algorithm. In every GA problem, parameters and solutions must be coded into chromosomes before they can be processed. As mentioned in Section 4.1, a chromosome is represented by a combination of numbers, alphabets, etc (Elif, 2005). Different coding methods require different crossover and mutation operations accordingly (Sharma, 2015). Therefore, the coding method has a great influence on the convergent efficiency of the genetic algorithm (Halim, 2015). In conclusion, a reasonable and effective coding method is the foundation of GA for DSP problems.

Since each component or subassembly has already been numbered during the product's representation, those part numbers could be used as coding numbers directly. For example, a chromosome "$9,5,6,3,2,1,4,7,8$" represents the disassembly sequence is set as $9-5-6-3-2-1-4-7-8$. It is quite clear and straightforward that the generated solution is the disassembly sequence.

(2) Selection Operation

Selection operation is the basic operation for a genetic algorithm. It reflects the process of "survival of the fittest" in the natural world, which means the chromosome with a higher fitness score is more likely to be selected and reproduced to the next generation. In order to simulate this process in GA for DSP, all disassembly sequences are selected randomly while the sequences with higher fitness scores have higher probability to be selected than those sequences with lower fitness scores.

In this thesis, the "roulette wheel" selection method is used to determine the selection probability for each disassembly sequence. This probability is proportional to sequence's fitness score. It is assumed that the population quantity is M, the fitness score of sequence *i* is F_i and the selection probability of sequence *i* is P_i:

$$P_i = F_i \Big/ \sum_{i=1}^{M} F_i \quad (i=1, 2,\ldots, M) \qquad (4\text{-}1)$$

(3) Crossover operation

Crossover operation simulates the genetic recombination process of sexual propagation. It aims to generate new chromosomes while maintains the integrity of chromosome at the same time. In this thesis, two children disassembly sequences are constructed by changing parts of their father sequences. It is called partially matched crossover (PMC). Details are shown as follows:

1: Randomly selected a crossover section in two parents sequences.

Parent 1: 5 2 1 4 / 6 9 7 / 3 8 (crossover section 6 9 7)

Parent 2: 2 6 3 4 / 1 8 7 / 5 9 (crossover section 1 8 7)

2: Attach parent 2's crossover section ahead of parent 1; attach parent 1's crossover section ahead of parent 2.

Child 1: 1 8 7 / 5 2 1 4 6 7 9 3 8

Child 2: 6 7 9 / 2 6 3 4 1 8 7 5 9

3: Delete repeated genes in sequence Child 1 and Child 2

Child 1: 1 8 7 5 2 4 6 9 3

Child 2: 6 7 9 2 3 4 1 8 5

Using the crossover operation, two children sequences are generated and they will be evaluated by the predefined fitness function as their parents sequences did. In addition, compared to other crossover methods, PMC has its unique advantage when two patents sequences are the same. It can still function effectively and generate different children sequences hence to reduce the premature convergent phenomenon (Schneider, 2013). Probability of crossover operation in this genetic algorithm is set as P_c.

(4) Mutation operation

After going through the crossover operation, the disassembly sequences are subjected to a mutation operation. The sequences mutate with a given probability of P_m . During this process, two genes are selected randomly and one gene will be inserted behind the other one. This is called an insert mutation process. Even though the mutation operation could generate new chromosomes, it is more likely to sabotage existing high-quality chromosomes (Schneider, 2013). Therefore, the value of mutation probability P_m must be a small value in this research.

4.3 Fitness evaluation

Fitness function is the key factor to genetic algorithms because it will be used to evaluate each disassembly sequence. In this DSP research, the fitness function is dependent on the increment in total disassembly time. Chapter three introduced two types of disassembly time, namely disassembly time for component and disassembly direction change time. In this chapter, a third factor is considered to better evaluate each disassembly sequence. It is the disassembly tool change time.

(1) Disassembly time for components

The time required to disassemble a component is different from one to another depending on the part's weight and shape complexity, etc (Smith, 2012). Some parts take longer time than others for processing. The longer disassembly time may result from the difficulty of a disassembly operation. Table 4-1 shows the disassembly time for different types of components regarding their geometrical complexity (Li, 2011).

Criterion	No.	Complex level	Additional time required
Geometric complexity	1	Part size: (1) Regular size (2) Small size (3) Large size/heavy (4) Super large/heavy (need additional assistant)	 0s 23s 25s 86s
	2	Handing difficulty (1) Tool or fixture required (2) Difficulty	 20s 18s
	3	Feature may cause jam and tangle (1) YES (2) NO	 12s 0s
Process complexity	4	Fastening type (1) Multi-pieces (nuts and bolts) (2) Screws and nails (3) Rivets, staples and adhesive	 24s 24s 50s
	5	Enough space for disassembly operation (1) YES (2) NO	 0s 20s
	6	Feature may cause jam and tangle (1) YES (2) NO	 22s 0s

Table 4-1 Disassembly time criteria

This is a list of general disassembly time criteria for different components, while different types of products may require more or less disassembly time based on these criteria. However, there is always a rule to follow no matter based on design requirements or the human experience (Tian, 2013).

The disassembly time T_c can be summed as follows (n is the number of components to be removed)

$$T_c = \sum_{i=1}^{n} T_i \tag{4-2}$$

(2) Time for disassembly direction change

As discussed in Chapter three, frequently changing disassembly directions will lead to an extra labor cost and time cost. It is preferable to disassemble as many components as possible in one direction operation. Therefore, the second criterion in the fitness evaluation is the penalty for disassembly direction changes. In this thesis, it is defined that each direction change with 90° will cost 2 seconds and 4 seconds for a direction change with 180° . If no direction change is required, the time penalty is set as 0.

$$T_d = \begin{cases} 0\text{s}, & \text{If no direction change is required} \quad \text{e.g. +X to +X} \\ 2\text{s}, & \text{If } 90^\circ \text{ change is required} \quad \text{e.g. +X to +Y} \\ 4\text{s}, & \text{If } 180^\circ \text{ change is required} \quad \text{e.g. +X to –X} \end{cases}$$

(3) Time for disassembly tool change

The last criterion in the fitness function is the time penalty for disassembly tool change. During the disassembly operation, different tools are used to remove corresponding constraints. It is also preferable to disassemble as many components as possible with the same tool at one operation. In this thesis, it is defined that the time penalty for each tool change is 4 seconds.

$$T_t = \begin{cases} 4s, & \text{If tool change is required} \\ 0s, & \text{If no tool change is required} \end{cases}$$

In conclusion, the total disassembly time is the combination of three aspects: time for components, time for direction change and time for tool change. Hence, the total disassembly time can be defined as follows:

$$T_{\text{total}} = T_c + T_d + T_t = \sum_{i=1}^{n} T_i + \sum_{j=1}^{m} T_d + \sum_{k=1}^{p} T_t \qquad \text{(4-3)}$$

In formula 4-3, *n* means the total number of components to be removed, *m* denotes total numbers of disassembly direction changes, and *p* is the total number of disassembly tool changes.

In this proposed GA method, the objective is to achieve a near-optimal disassembly sequence with near-least total disassembly time. In other words, the objective is to minimize the value of T_{total}. However, for all GAs, the objective is to select the chromosome with the highest fitness value. Hence, the fitness function has to be converted to a maximization function, which is shown in following Formula (4-4).

$$F_{fitness} = C - T_{total} \qquad \text{(4-4)}$$

In Formula 4-4, $F_{fitness}$ is the fitness value of a chromosome and C is the conversion constant which will cover the “low is better - minimization” function to a “high is better - maximization” model. Therefore, constant C should be chosen to satisfy the following condition.

$$C > T_{\text{total}} \qquad \text{(4-5)}$$

4.4 Implementation of the genetic algorithm

A general flowchart of GA for DSP is shown in Figure 4-1. However, the disassembly sequence planning problem has its distinctive feature compared to other problems. Each disassembly sequence is regarded as one chromosome and new chromosomes are created by crossover and mutation operations. All disassembly sequences should meet the geometrical constraints in the first place; otherwise those sequences will be infeasible. In this chapter, a disassembly feasibility check is performed after the crossover and mutation process, and infeasible sequences will be identified and skipped. Product constraint matrix is constructed to check the disassembly feasibility of generated sequences. Basic knowledge of constraint matrix has been introduced in the Chapter 3.

Therefore, the flowchart of the genetic algorithm for DSP in this chapter is updated as shown in Figure 4-2. The disassembly feasibility check will be performed after each crossover and mutation process. Only those feasible disassembly sequences could pass the examination.

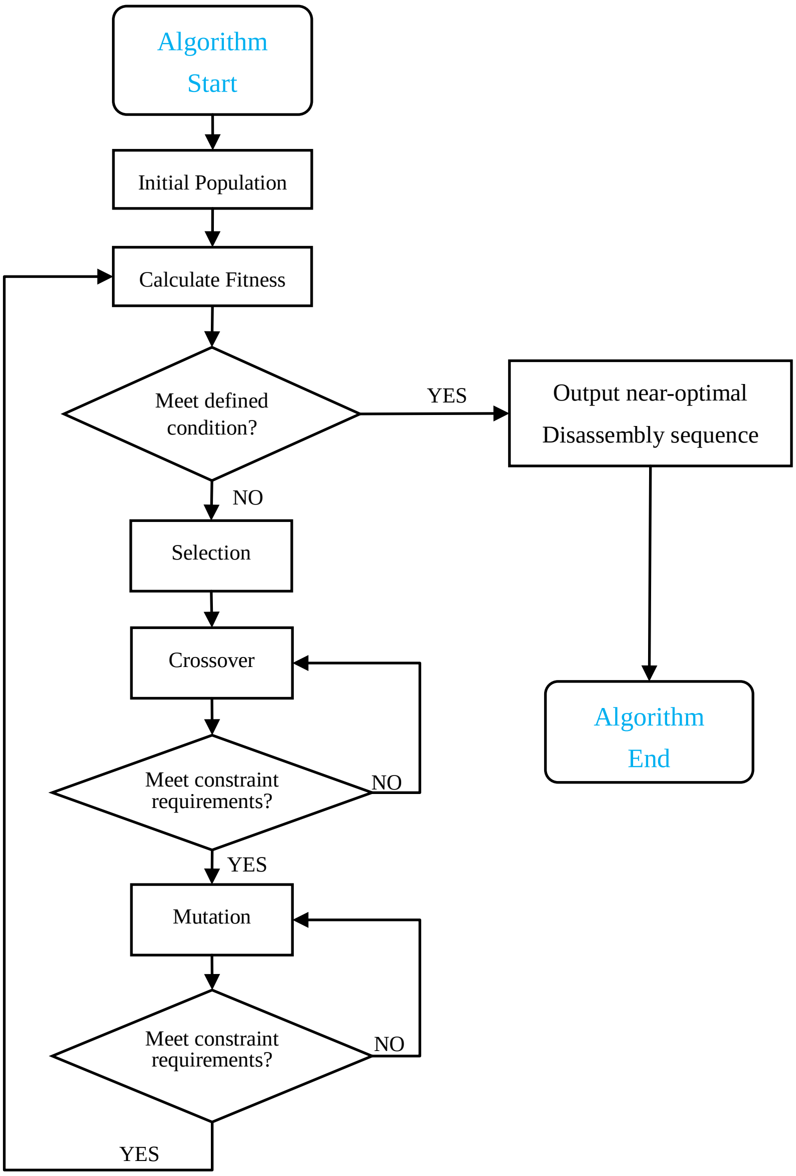


Figure 4-2 Updated flow chart of GA for DSP

When a new disassembly sequence is generated by the crossover operation, it will go through a disassembly feasibility check using the constraint matrix. If the sequence is proved to be a feasible sequence, it will go to the next operation. Otherwise, the sequence will be skipped. When a new disassembly sequence is generated by the mutation operation, it will also go through the same disassembly feasibility check and unfeasible ones will be skipped. In this way, all sequences in the algorithms are ensured as feasible sequences.

As mentioned before, the genetic algorithm will search disassembly sequences based on product's BOM. The algorithm will keep detecting the target component whenever a component is disassembled. For selective disassembly sequence planning, if a large subassembly with the target component is disassembled, the searching process moves to the next level (small subassembly level). Only when the target component itself is disassembled, the searching process stops.

Taking the gear box in Figure 3-2 as an example, if component 12 is selected as the target component, according to the above description, only two matrices should be searched. They are shown in Figure 3-5 and Figure 3-6 (c), respectively. When searching the first matrix, they are totally $4!=24$ searches using the traversal algorithm. However, there are only 17 feasible sequences are searched using the genetic algorithm. When searching the second matrix, they are totally $5!=120$ searches using the traversal algorithm. In comparison, they are only 56 feasible sequences are searched based on the genetic algorithm. The comparison is made as shown in Table 4-2.

	Sequences searched In the first layer	**Sequences searched In the second layer**
Traversal algorithm	4!=24	5!=120
Genetic algorithm	17	56
Reduced	29.1%	53.3%

Table 4-2 Comparison between traversal algorithm and GA

In conclusion, there are fewer sequences searched by using the genetic algorithm and hence the DSP efficiency is improved. However, there are still some infeasible sequences generated in the genetic algorithm and they should be identified and skipped. In this thesis, disassembly feasibility check is implemented to ensure those infeasible sequences are searched and deleted.

4.5 Case Study for disassembly feasibility check

In this section, a camshaft assembly is used as an example to show how the disassembly feasibility check works during the crossover and mutation. A comparison will be made between the genetic algorithm with and without disassembly feasibility check. Component 6 is selected as the target component. An explosion graph of the camshaft assembly is shown in Figure 4-3. The camshaft model is from a research publication and all the constraints are identified (Han, 2007).

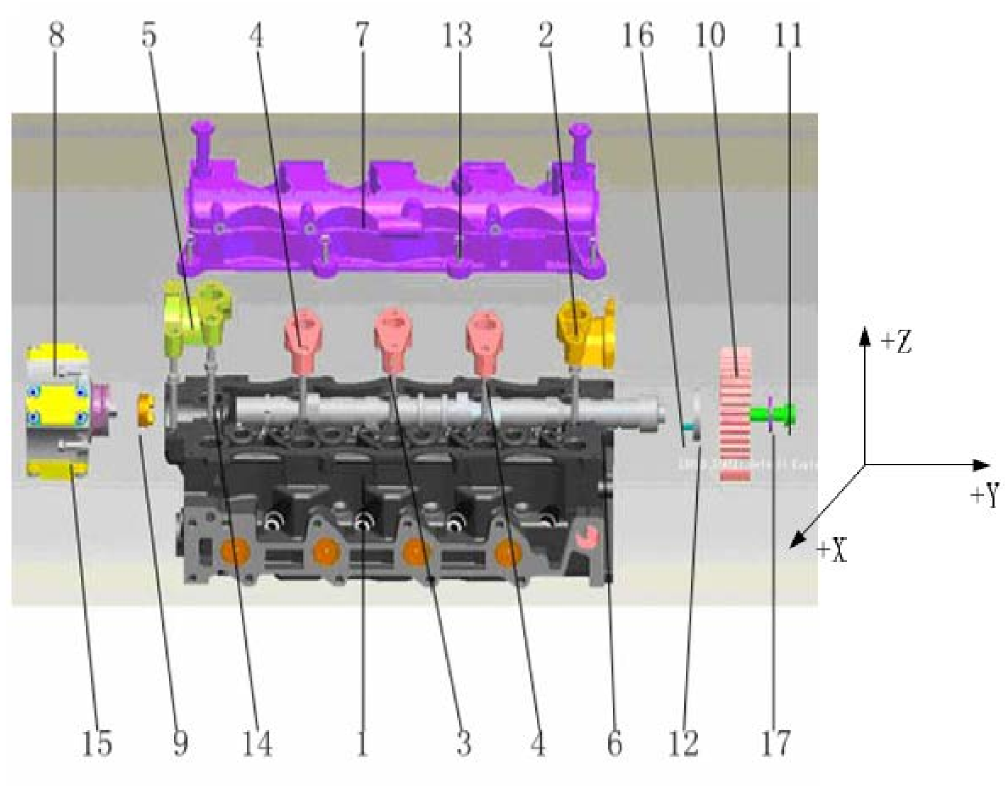


Figure 4-3 Explosion graph of the camshaft assembly

According to the component listed in Table 4-5, this camshaft assembly consists of 12 types of components. In order to simplify the camshaft assembly, components quantity of all types is set as one because the same type of the components is processed in one operation. Their detailed information is shown in Table 4-3.

Part No.	Part Name	Quantity	Tool	Direction
1	Body	1	N/A	+Z
2	Camshaft bearing cap (front)	1	T1	+Z
3	Camshaft bearing cap (middle)	1	T1	+Z
4	Camshaft bearing cap (side)	2	T1	+Z
5	Camshaft bearing cap (back)	1	T1	+Z
6	Camshaft	1	T2	+Z
7	Body cover	1	T3	+Z
8	High-pressure pump	1	T3	-Y
9	Connection piece	1	T3	-Y
10	Gear	1	T4	+Y
11	Gear fastening bolt	1	T5	+Y
12	Camshaft oil-seal	1	T6	+Y
13	Body cover bolt	8		
14	Camshaft bearing cap fastening bolt	12		
15	High-pressure pump fastening bolt	3		
16	Gear locating pin	1		
17	Gear shim	1		

Table 4-3 Component list and information

(1) Initial population

Initial population's quantity and quality have a great influence on the GA's final convergence (Tseng, 2010), the quantity of initial population in this chapter is set as 40 and they are shown in Table 4-4. In order to make a good comparison between the GA with and without disassembly feasibility check, the initial population is set based on the recommendation of the research publication (Han, 2007). As discussed before, the disassembly sequence planning has its unique requirements on the initial population. High quality of an initial population can generate a near optimal sequence in a shorter time (Haichao, 2015). As component 1 is the base of camshaft, it supports other components in stable. Therefore, component 1 should be removed after the rest components. The sequences will not consider component 1.

11, 8, 9, 7, 12,10, 5, 4, 3, 2, 6	8, 9, 11, 7, 10, 5, 4, 12, 3, 2, 6,
8, 9, 7, 12,11, 10, 5, 4, 3, 2, 6	11, 8, 9, 7, 10, 5, 4, 3, 2, 12, 6
8, 9, 11, 7, 10, 5, 4, 3, 2, 12, 6	8, 9, 7, 11 ,10, 5, 4, 3, 2, 12, 6
11 ,8, 9, 7, 10, 5, 4, 3, 12, 2, 6	8, 9, 11, 7, 10, 5, 4, 3, 12, 2, 6
8, 9, 7, 11, 10, 5, 4, 3, 12, 2, 6	11, 8, 9, 7, 10, 5, 4, 12, 3, 2, 6
8, 9, 11, 7, 10, 5, 4, 12, 3, 2, 6	8, 9, 7, 11, 10, 5, 4, 12, 3, 2, 6
11, 8, 9, 7, 10, 5, 12, 4, 3, 2, 6	8, 9, 11, 7, 10, 5, 12, 4, 3, 2, 6
8, 9, 7, 11, 10, 5, 12, 4, 3, 2, 6	11, 8, 9, 7, 10, 12, 5, 4, 3, 2, 6
8, 9, 11, 7, 10, 12, 5, 4, 3, 2, 6	8, 9, 7, 11, 10, 12, 5, 4, 3, 2, 6
11, 7, 8, 9, 12,10, 5, 4, 3, 2, 6	7, 11, 8, 9, 10, 5, 4, 12, 3, 2, 6,
7, 8, 9, 12,11, 10, 5, 4, 3, 2, 6	11, 7, 8, 9, 10, 5, 4, 3, 2, 12, 6
7, 11, 8, 9, 10, 5, 4, 3, 2, 12, 6	11, 7, 8, 9, 10, 5, 4, 3, 12, 2, 6
11, 7, 8, 9, 10, 5, 4, 12, 3, 2, 6	7, 11, 8, 9, 10, 5, 4, 12, 3, 2, 6
7, 8, 9, 11, 10, 5, 4, 12, 3, 2, 6	11, 7, 8, 9, 10, 5, 12, 4, 3, 2, 6
7, 8, 9, 11, 10, 5, 12, 4, 3, 2, 6	11, 7, 8, 9, 10, 12, 5, 4, 3, 2, 6
7, 11, 8, 9, 10, 12, 5, 4, 3, 2, 6	11, 7, 10, 8, 9, 5, 4, 3, 2, 12, 6
11, 7, 10, 8, 9, 5, 4, 3, 2, 12, 6	7, 11, 10, 8, 9, 5, 4, 3, 2, 12, 6
11, 7, 10, 8, 9, 5, 4, 3, 12, 2, 6	7, 11, 10, 8, 9, 5, 4, 3, 12, 2, 6
11, 7, 10, 8, 9, 5, 4, 12, 3, 2, 6	7, 11, 10, 8, 9, 5, 4, 12, 3, 2, 6
11, 7, 10, 8, 9, 5, 12, 4, 3, 2, 6	7, 11, 10, 8, 9, 5, 12, 4, 3, 2, 6

Table 4-4 Initial population of GA

(2) Genetic operation and fitness evaluation

In the genetic algorithm, each chromosome is evaluated by a predefined fitness function and goes through three operations: selection, crossover and mutation. In this case study, GA parameters are defined as follows (Xing, 2012):

Initial population = 40

Crossover probability $P_c = 0.8$

Mutation probability $P_m = 0.1$

According to disassembly time criteria in Table 4-1, the disassembly time for each component is identified in Table 4-5.

Part No.	1	2	3	4	5	6	7	8	9	10	11	12
Required Time	8	4	4	4	6	2	9	5	2	3	2	2

Table 4-5 Disassembly time (in second) for components in the camshaft assembly

Fitness formula is shown as follows:

$$F_{fitness} = C - T_{total}$$

$$T_{\text{total}} = T_c + T_d + T_t = \sum_{i=1}^{n} T_i + \sum_{j=1}^{m} T_d + \sum_{k=1}^{p} T_t$$

Take sequence (S=7, 8, 9, 11, 10, 2, 3, 4, 5, 12, 6) as an example, its fitness score is calculated as follows:

$$F_{fitness}(\text{S}) = \text{C} - T_{\text{total}} = C - (\sum_{i=1}^{n} T_i + \sum_{j=1}^{m} T_d + \sum_{k=1}^{p} T_t) = 200 - (56+20+12) = 112$$

As mentioned before, C is the conversion constant that coverts the “lower is better - minimization” function to a “higher is better - maximization” model. In this case study, its value is set as 200. In order to reach a stable fitness value, the algorithm will not stop until the generated sequence's fitness value keeps stable for more than 10 generations.

The constraint matrix of this product is constructed as shown in Figure 4-4. It is used to perform the disassembly feasibility check. Finally, the genetic algorithm is performed in the MATLAB software and the result of the GA with disassembly feasibility check is shown in Figure 4-5.

	1	2	3	4	5	6	7	8	9	10	11	12
1	000000	000010	000010	000010	000010	000010	000010	000010	000010	000010	000010	000010
2	000001	000000	000100	000100	000100	111101	000010	000100	000100	001000	001000	001000
3	000001	001000	000000	000100	000100	111101	000010	000100	000100	001000	001000	001000
4	000001	001000	001000	000000	000100	111101	000010	000100	000100	001000	001000	001000
5	000001	001000	001000	001000	000000	111100	000010	111101	000100	001000	001000	001000
6	000001	111110	111110	111110	111100	000000	000010	110100	110100	110011	110011	110011
7	000001	000001	000001	000001	000001	000001	000000	111101	111101	111101	111101	111101
8	000001	001000	001000	001000	111110	111000	111110	000000	111011	001000	001000	001000
9	000001	001000	001000	001000	001000	111000	111110	110111	000000	001000	001000	001000
10	000001	000100	000100	000100	000100	110011	111110	000100	000100	000000	111111	110011
11	000001	000100	000100	000100	000100	110011	111110	000100	000100	111111	000000	110111
12	000001	000100	000100	000100	000100	110011	111110	000100	000100	110011	111011	000000

Figure 4-4 Constraint matrix of the camshaft

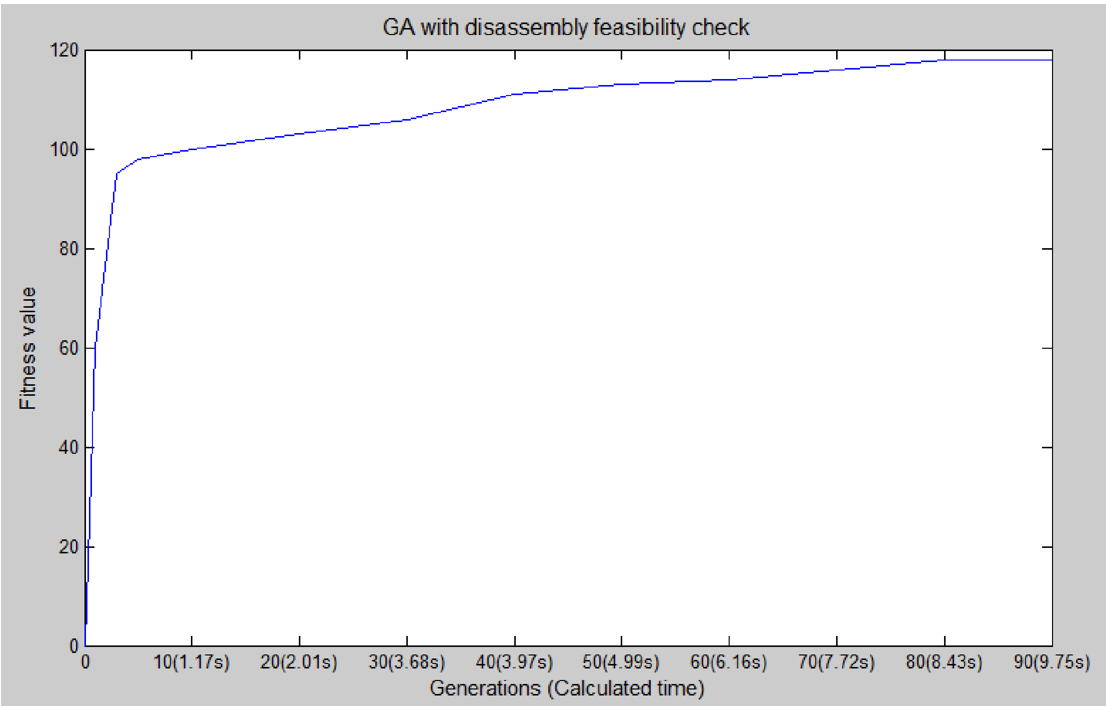


Figure 4-5 GA with the disassembly feasibility check

According to the graph in Figure 4-5, the fitness value can get to a relative high value in a short time and then the fitness value reaches a stable level after 80 generations. Based on the stopping criterion defined before, two chromosomes at generation 90 are selected as the near-optimal disassembly sequences, and their fitness scores are both 118. The near-optimal disassembly sequences are shown in Table 4-6.

11-10-12-7-8-9-2-3-4-5-6	11-10-12-8-9-7-2-3-4-5-6

Table 4-6 Near-optimal disassembly sequences of camshaft assembly

In comparison, the result of the GA without disassembly feasibility check is shown in Figure 4-6. According to the graph in Figure 4-6, the fitness value takes more time to get a relative high value and the fitness value reaches a stable level after 130 generations. The algorithm stops after 140 generations and the output fitness value is 85.

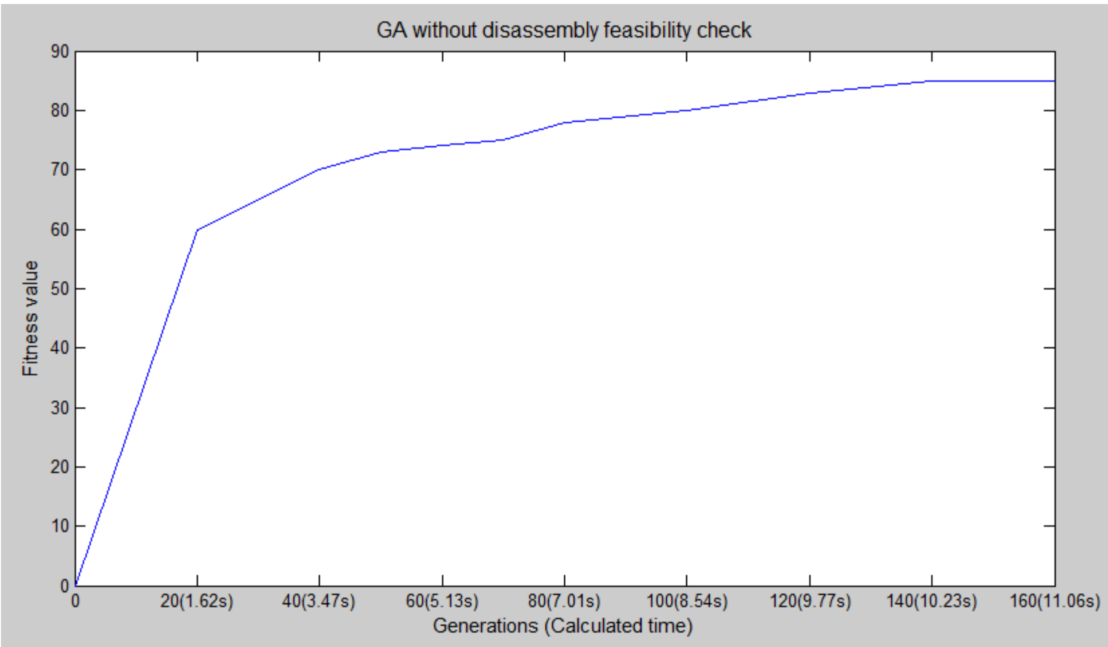


Figure 4-6 GA without disassembly feasibility check

In conclusion, both methods could work well and find a solution finally, but it is obvious that the GA with disassembly feasibility check could generate a better disassembly sequence with a shorter generations. The comparison is shown in Table 4-7.

DSP Method	Find a solution?	Generations	Fitness value	CPU time
GA without disassembly feasibility check	YES	160	85	11.056s
GA with disassembly Feasibility check	YES	90	118	7.993s

Table 4-7 Comparison for GAs with and without feasibility check

4.6 Summary

In this chapter, a genetic algorithm is used to generate a near-optimal disassembly sequence for a target component. Total disassembly time is used as the evaluation criterion. It consists of three aspects, namely disassembly time for components, time for disassembly directions change and time for disassembly tools change, respectively. Compared to the traversal algorithm, there are fewer disassembly sequences searched, which helps to improve the planning efficiency. Furthermore, a disassembly feasibility check is performed after the crossover and mutation operation. Constraint matrix can identify and skip infeasible sequences. Therefore, all sequences in the genetic algorithm are ensured as feasible ones. A case study of the camshaft assembly is used to compare the genetic algorithm with and without disassembly feasibility check. The comparison in Table 4-7 shows that the disassembly feasibility check can help to generate a better disassembly sequence with shorter generations. Furthermore, the algorithm efficiency is improved because the CPU time in MATLAB is reduced.

Chapter 5

Destructive Method for DSP

5.1 Introduction

In this chapter, a destructive disassembly method is proposed to generate the sequence for the selective disassembly. Non-destructive disassembly operations aim to remove constraints between components while maintaining all components undamaged. However, in some cases, it is not necessary to keep all components undamaged. For example, in the selective disassembly operation for an End-of-Life (EOL) product, as long as the target component is undamaged, the rest components could be destroyed in order to find a short disassembly path. In addition, when two components are connected by rivets or welding connections as shown in Figure 5-1, a destructive operation has to be performed because those constraints could not be removed by non-destructive disassembly operations.

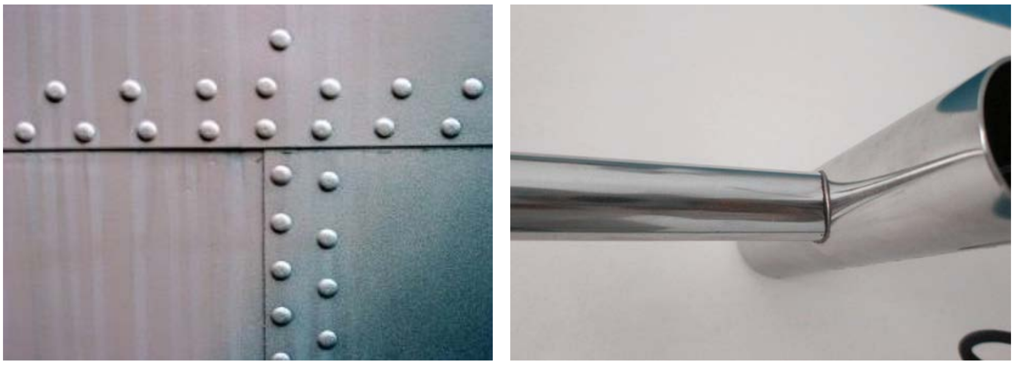


Figure 5-1 Riveting and welding products

During the disassembly sequence searching process, each constraint's type is identified for corresponding disassembly operation. For example, if a connection is identified as welding, then a destructive disassembly operation will be applied; if a connection uses the screw, a screwdriver will be used in a non-destructive operation. In this way, all constraints can be removed by specific disassembly operations to ensure the generated disassembly sequences are practical to guide the disassembly process.

Although destructive disassembly operations may take more disassembly time and effort than non-destructive operations, they can approach the target component with fewer components to be removed (Tseng, 2011). The total disassembly time or cost may be reduced. The disassembly cost is calculated based on both non-destructive disassembly and destructive disassembly operations. The sequence with less cost will be selected for the disassembly operation.

5.2 Literature review and research objective

Destructive disassembly methods have been studied by researchers. Pak *et al.* generated a shorter disassembly sequence by destroying bolts and screws using elastic waves. Elastic waves are modeled in a one-dimensional bar, which transfers the impact of the energy to a protruded bolt head mounted in an elastic medium (Pak, 2002). This method could destroy the bolts and screws in a short time but it is particularly used in damaged fasteners at the end of product life cycle. Umeda *et al.* optimized Pak's method using a split line to destruct the product in a desired shape and then extracted the target component (Umeda, 2015). Kyonchun *et al.* proposed an algorithm for the destructive disassembly to guide the design-for-disassembly (Kyonchun, 1998). Reap *et al.* also explored the recovery value of products' materials by a semi-destructive disassembly method and proposed a design-for-disassembly idea (2002). Pan *et al.* proposed a partial destructive method to remove certain constraints to find a physical short disassembly path for electronic products (Pan, 2012). Zhou introduced a method to identify “UTD” (unable to disassembly) components in a disassembly sequence and applied destructive operations to remove those components. He also used several criteria to evaluate the disassembly cost with destructive ways to guide the design for disassembly (DFD) (Zhou, 2014). Song *et al.* used the disassembly graph to represent the product and constraints. An object inverse-directed method was proposed to optimize the disassembly design and model reconstruction to achieve the better DSP (Song, 2014). In conclusion, most studies on the destructive disassembly focused on how to destroy components and optimize the design process. Few of them considered how to use destructive methods to optimize the disassembly sequence planning for existing products. Therefore, in this chapter, a destructive disassembly method is proposed to find an economical disassembly path. Time cost comparison is made between the destructive disassembly operation and non-destructive disassembly operation. Case studies of a mechanical arm and an MP4 player are used to verify the proposed method.

5.3 UTD problems in the non-destructive disassembly

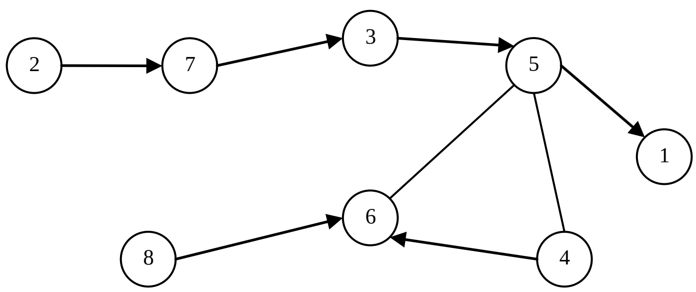


Figure 5-2 Graph representation example

Figure 5-2 shows a constraint directed graph for a product with 8 components. The graph representation method has been introduced in Chapter two. In Figure 5-2, arrowed segments indicate that two components have the pre-defined disassembly precedence from one component to another, while segments without arrows means that two components have the equal disassembly precedence. For example, $C8 \rightarrow C6$ means that component 6 cannot be disassembled until component 8 is removed. $C5 - C4$ means that either component 5 or component 4 can be removed first. In Figure 5-2, the disassembly precedence between components is shown without constraints' details. For example, if the constraint between components 4 and 6 is a riveting connection, these two components have to be bypassed in a non-destructive disassembly sequence as they can only be removed by destructive disassembly operations.

If component 1 is selected as the target component, the ideal shortest disassembly sequence is $8 \rightarrow 6 \rightarrow 5 \rightarrow 1$, there are four components in total to be removed. However, as mentioned before, if the connection between components 6 and 4 is riveting, the generated non-destructive disassembly sequence will not be able to get the target component unless the riveting constraint between components 6 and 4 is released. If the component 6 is bypassed using a non-destructive sequence, a longer disassembly path has to be selected: $2 \rightarrow 7 \rightarrow 3 \rightarrow 5 \rightarrow 1$. This is called the "detour" phenomenon (Shana, 2011).

In this case, component 6 is defined as an "UTD" (unable to disassemble) component. In conclusion, without considering the connection that can only be removed by destructive operations, many "optimal or near-optimal" disassembly sequences may not be verified to achieve the target component.

Non-destructive disassembly sequences have to bypass those "UTD" components, while the destructive disassembly can destroy them. In this chapter, a destructive disassembly method is proposed to improve limitations of the non-destructive disassembly method. General steps of this method are as follows:

(1) Represent the product with the graph representation.

(2) Represent the product with multi-level constraint matrix and fastener-component matrix.

(3) Generation of feasible disassembly sequences considering both operations.

(4) Evaluation of the solutions for non-destructive and destructive methods using established criteria.

(5) Selection of the disassembly sequence with the minimum cost.

Details of each step are discussed in following sections. Case studies of a mechanical arm and an MP4 player are used to verify the proposed method.

5.4 Product presentation and component constraints

5.4.1 Constraint information for the destructive disassembly

Constraints are identified to represent component connections of a product. For example in Chapter 3, multi-level constraint matrices are constructed to represent a product. However, without considering destructive disassembly methods, the disassembly sequences may have the "detour" problem for bypassing constraints that can only be removed by destructive ways.

In this chapter, constraints' types are extended including constraints that can only be removed by destructive ways. This information is integrated in the products' representation.

Constraints are classed into following four categories based on connection relations: Fastener constraint (F), Mating constraint (M), Buckling constraint (B) and Destructive constraint (D). In this way, constraints can be identified and then released by corresponding disassembly operations.

Details are shown in Table 5-1.

C_{ij}^{k}	Constraint(F)	Constraint(D)	Constraint(M)	Constraint(B)
Constraint Type	Fastener connect	Destructive	Mating contact	Buckling connect

Table 5-1 Constraints type between component *i* and *j*

Where C_{ij}^{k} *refers to the constraint between components i and j; k is the type of constraints*

(1) Fastener constraint (F): it is commonly used in products such as thread connections using bolts and screws. Normally, the screwdriver and wrench are common tools used to remove this type of constraints. Fastener connection force is strong between components but it can be disassembled by tools in a non-destructive way.

(2) Mating constraint (M): two components are connected by their geometric shape. This type of constraints can be easily released by hands or simple tools. For example, a plug is fixed on a socket by its geometric shape, which can be removed easily by hands.

(3) Buckling constraint (B): it is commonly used in some non-heavy duty loading products such as small electric appliance and plastic products. The constraint force of bucking is much smaller than fastener constraints.

(4) Destructive constraint (D): this type of constraints can only be removed in a destructive way. For example, two components are jointed together by gluing, welding, riveting, *etc*. In this chapter, all components with destructive constraints are defined as “UTD” components.

In conclusion, for those constraints that can only be removed by destructive operations, they will be identified as type D constraints during the product representation. In the disassembly sequence searching process, they will be either destroyed by destructive operations or bypassed by non-destructive operations. Although destructive disassembly operations may take more time and effort, it can reach the target component with fewer components to be removed. The total disassembly cost may be reduced.

5.4.2 Multi-level constraint matrices and fastener matrix

In the following sections, multi-level constraint matrices are constructed based on product's BOM. They can represent constraints relations between components. Those components do not include fasteners. Therefore, a fastener matrix is built to show constraints from fasteners. Fasteners should be removed before disassembling components. Details will be introduced in Section 5.6.1 for the case study of a mechanical arm.

5.5 Disassembly sequence planning

5.5.1 Disassembly cost criteria

The disassembly cost can be counted by the disassembly time. The total disassembly time is used as a criterion to evaluate the disassembly sequence. Disassembly time varies in different disassembly operations. Table 5-2 lists the disassembly time based on tools used for two disassembly patterns (Ilgin, 2011).

<table>
<tr><th colspan="2">Disassembly type</th><th>Constraint(F)</th><th>Constraint(B)</th><th>Constraint(M)</th><th>Constraint(D)</th></tr>
<tr><td rowspan="2">Non-destructive</td><td>Tool</td><td>Screwdriver
Wrench</td><td>Pliers</td><td>Hands</td><td>Hammer
Saws</td></tr>
<tr><td>Time</td><td>8-18s</td><td>12-25s</td><td>3-10s</td><td>10s-100s</td></tr>
<tr><td>Destructive</td><td>Tool</td><td>Electrical
drill
Pliers</td><td>Hammer
Saws</td><td>N/A</td><td>Electric
saw/drill</td></tr>
</table>

	Time	15-30s	5-15s	N/A	10-60s

Table 5-2 Disassembly time using tools for different constraints

5.5.2 Flowchart of proposed disassembly sequence planning

When a target component is determined, subassemblies containing the target component in each level of BOM can be identified. Only these subassemblies will be disassembled while other subassemblies are not to be considered. A disassembly sequence searching process starts from the target component to the end product. Two types of matrices are used to generate the desired disassembly sequences, namely multi-level constraint matrices and fastener-component matrix. Multi-level constraint matrices are used to examine the constraints between components and to skip infeasible sequences. Fastener - component matrix is used to find fasteners that should be removed in advance. For those fasteners or constraints that can only be removed by destructive disassembly operations, they will be identified and removed by specific tools.

A flow chart of this disassembly sequence planning is shown in Figure 5-3.

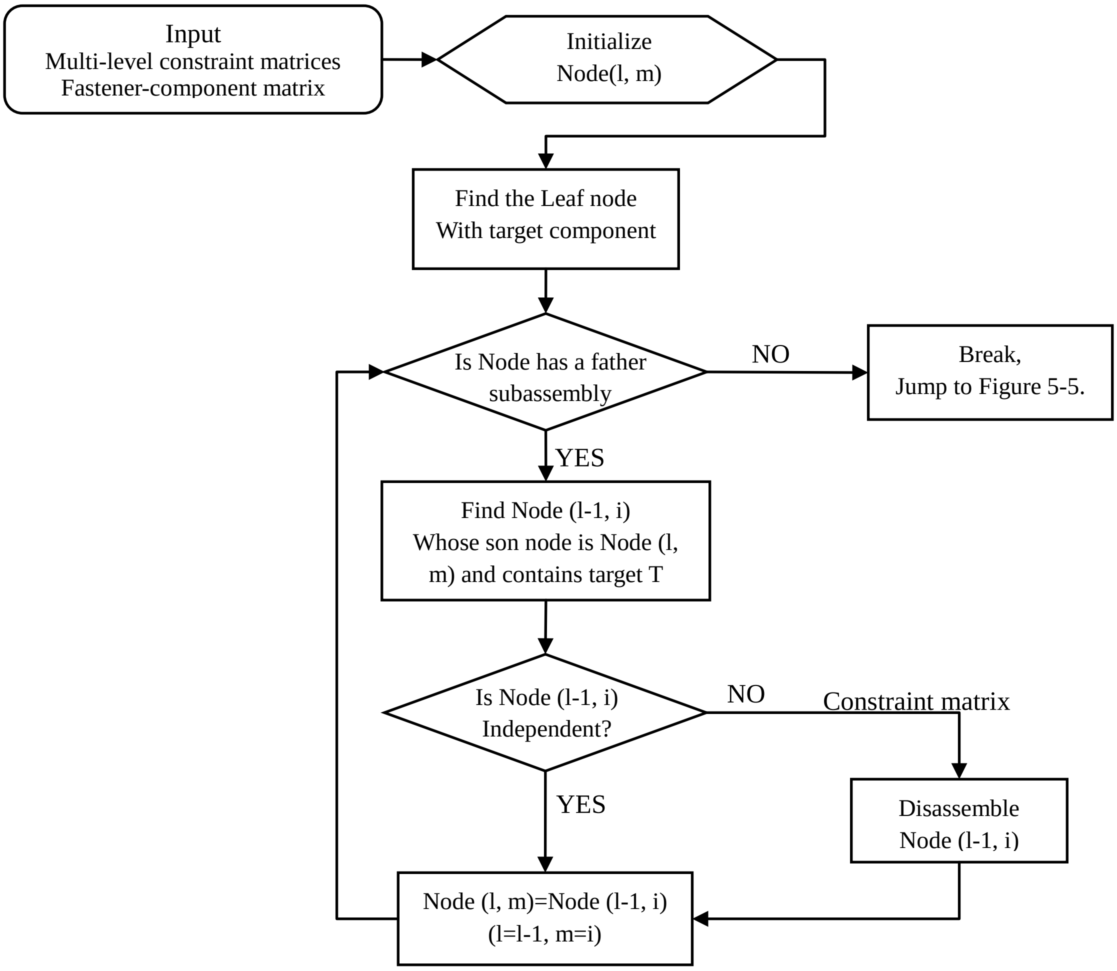


Figure 5-3 Search subassemblies with the target component

After inputting the multi-level constraint matrices, the product and subassemblies are recorded as a tree structure. The hierarchical relations of tree structure derive from the BOM of the product (Dong, 2006). Each node represents a subassembly. An example of the tree structure is shown in Figure 5-4.

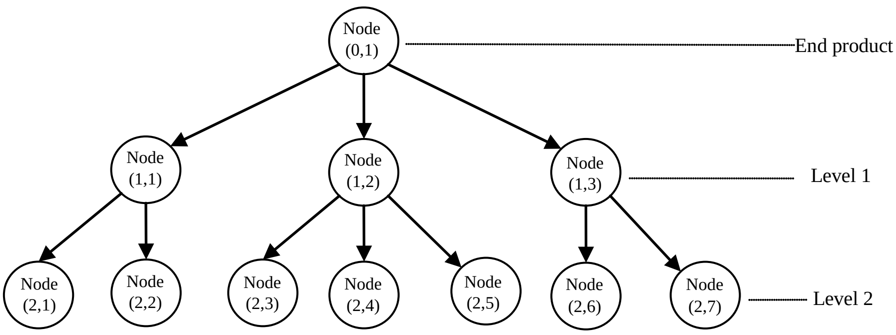


Figure 5-4 An example of the tree structure

The end product is a root node and recorded as Node (0, 1). Node (n, j) denotes the j^{th} node in the n^{th} level. Therefore, in Figure 5-4, the first subassembly in the first layer is recorded as Node (1, 1) and its son nodes are Node (2, 1) and Node (2, 2). The constraint matrix for Node (n, j) is M(n, j). The lowest subassembly is defined as "Leaf Node" that means no more subassemblies can be divided from the lowest level subassembly.

The searching algorithm starts from the target component to the end product. Since the target component T is determined, the "Leaf nodes" with the target component T can be identified. Only for those subassemblies whose father subassembly containing the target component will be searched. The searching algorithm will not stop until the end product is reached. The subassemblies are separated using the multi-level constraint matrices.

After the disassembly sequence from the end product to the lowest subassembly with the target component is generated, a following search algorithm is performed to get the target component from the lowest subassembly. The corresponding constraint matrix can be used to determine the number of components in this subassembly. For example, if the multi-level constraint matrix is a 7×7 matrix, the number of components is 7 and the permutation is 7!.

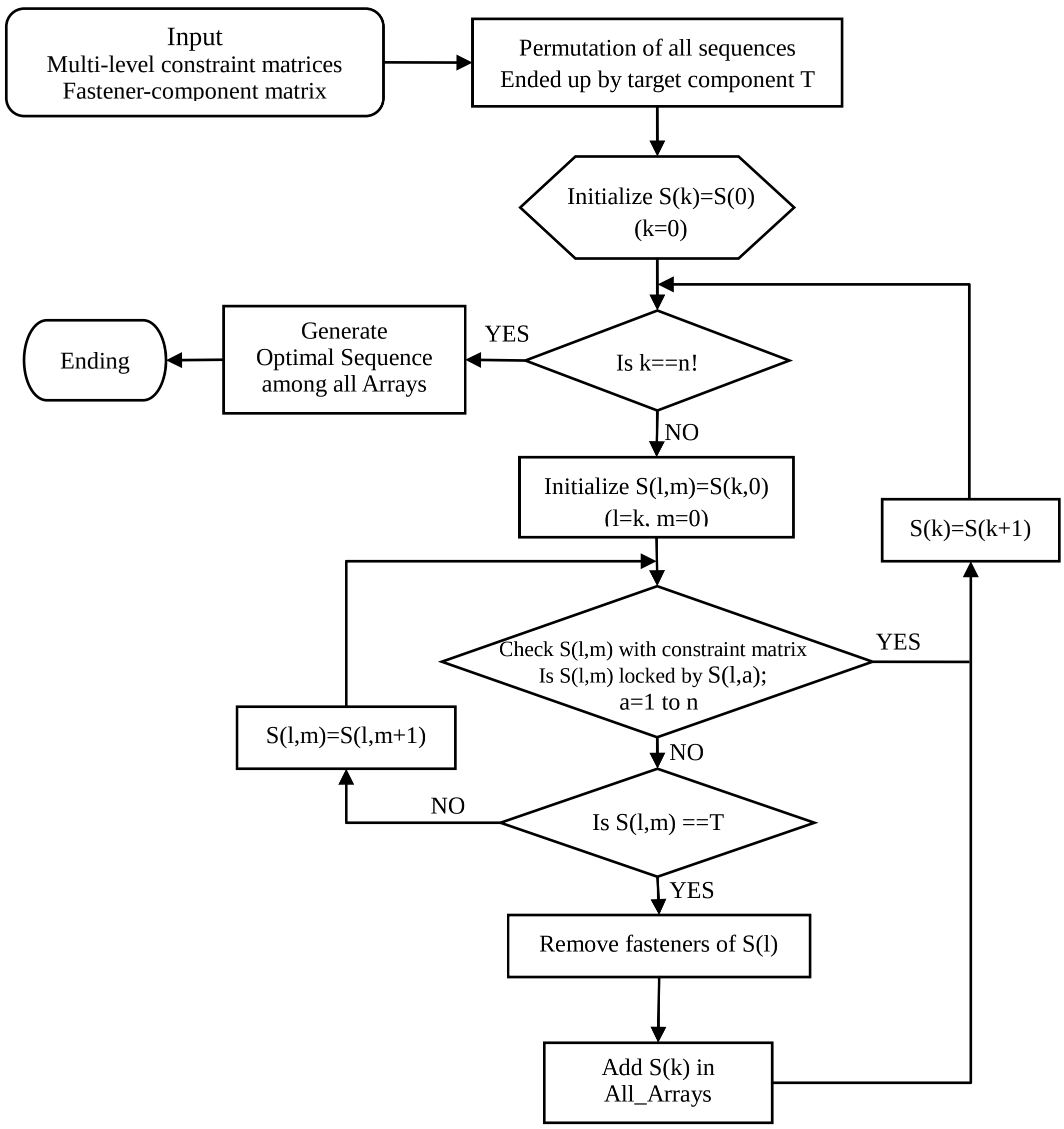


Figure 5-5 disassembly sequence search for the target component

According to Figure 5-5, it is assumed that the lowest subassembly with the target component has n components in total. Therefore, a full permutation will be made in MATLAB software to form n! sequences. S(i) denotes the ith sequence, and S(i,j) means the jth component in the ith sequence. Some sequences are not feasible. Therefore, all sequences are examined by the multi-level constraint matrix to determine the disassembly feasibility between components. For example, the searching algorithm checks the disassembly sequence S(i). Then the first component S(i,1) in sequence S(i) will be examined using the corresponding constraint matrix. If the component S(i,1) can be removed, a following check with the fastener matrix is applied to find the fasteners that should be removed in advance. Then the second component S(i,2) will be checked. If the component S(i,1) fails the feasibility check, it means the component cannot be removed, hence the sequence S(i) is infeasible and will be skipped. All sequences are ended up by the target component T and all feasible disassembly sequences are recorded in the "All_Arrays". Disassemble cost comparison is made among all feasible sequences and the optimal one will be selected. Details of this process will be introduced in the case study of a mechanical arm in Section 5.6.1.

5.6 Case studies for the destructive DSP

5.6.1 Mechanical arm

Figure 5-6 shows a simplified mechanical arm. It is commonly used in manufacturing industries for replacing human workers in repetitive activities or dangerous working environments. The mechanical arm consists of 12 components (components 1 to 12) and 11 fasteners (components 13 to 23) shown in Table 5-3. In this case study, component 8 is selected as the target component for recycling.

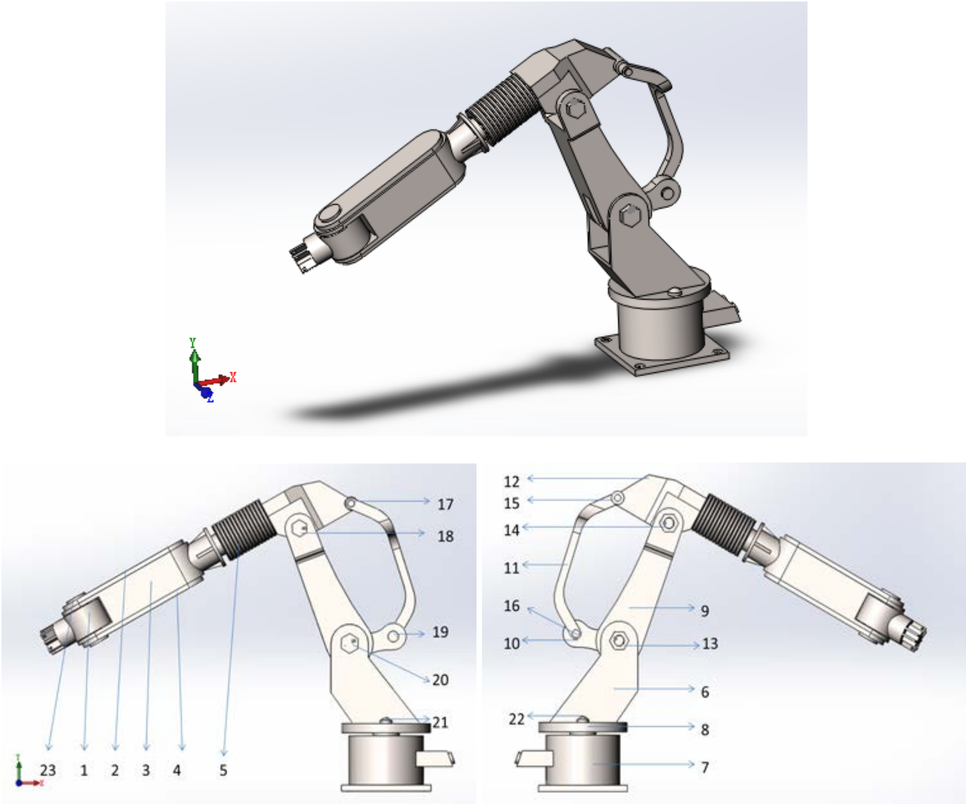


Figure 5-6 3D model of the mechanical arm

No.	Component Name	Quantity	Time non-destructive	Time destructive	Constraint type
1(C1)	Rotation joint	1	12s	23s	Pin
2(C2)	Connection plate	1	17s	20s	Pin
3(C3)	Front Arm	1	13s	17s	Pin+mating
4(C4)	Connection plate	1	10s	15s	Pin
5(C5)	Joint spring	1	18	26s	Mating
6(C6)	Base connector	1	12s	28s	Bolt
7(C7)	Base	1	15s	20s	Riveting (D)
8(C8)	Base plate	1	8s	20s	Riveting (D)
9(C9)	Arm support	1	11s	15s	Bolt+mating
10(C10)	Adapting piece	1	15s	22s	Pin+mating
11(C11)	Arm crank	1	20s	26s	Pin
12(C12)	Mechanical wrist	1	19s	31s	Bolt_pin

13(F1)	Nut	1	10s	25s	Bolt-nut
14(F2)	Nut	1	10s	25s	Bolt-nut
15(F3)	Fixture	1	8s	20s	Pin-fixture
16(F4)	Fixture	1	8s	20s	Pin+fixture
17(F5)	Pin	1	11s	18s	Pin+fixture
18(F6)	Bolt	1	12s	22s	Bolt-nut
19(F7)	Pin	1	9s	18s	Pin+fixture
20(F8)	Bolt	1	13s	22s	Bolt-nut
21(F9)	Rivet	1	N/A	25s	Rivet (D)
22(F10)	Rivet	1	N/A	25s	Rivet (D)
23(F11)	Rotate pin	1	19s	30s	Pin

Table 5-3 Components' information of the Mechanical arm

Based on the design information, the mechanical arm's bill of material (BOM) is formed as shown in Figure 5-7. According to the BOM, the mechanical arm can be separated into two first-level subassemblies. They are shown in Figure 5-8. The target component is in subassembly 2.

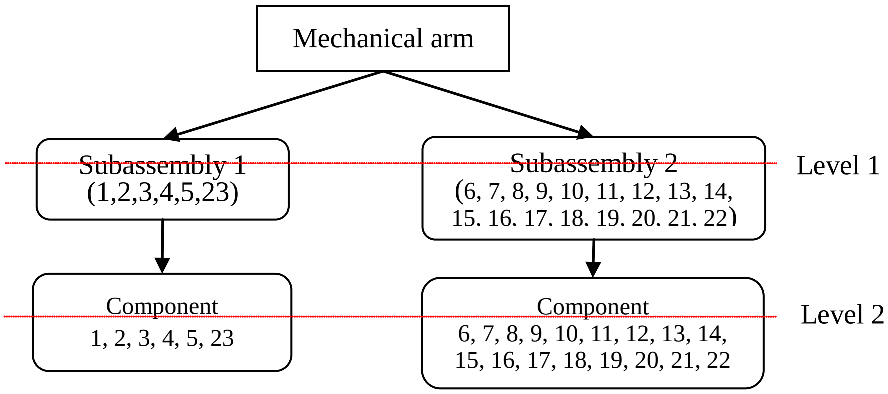
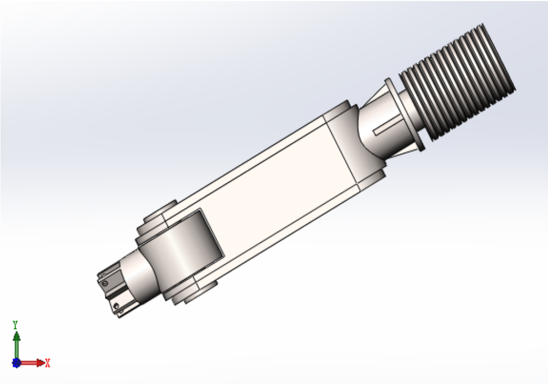
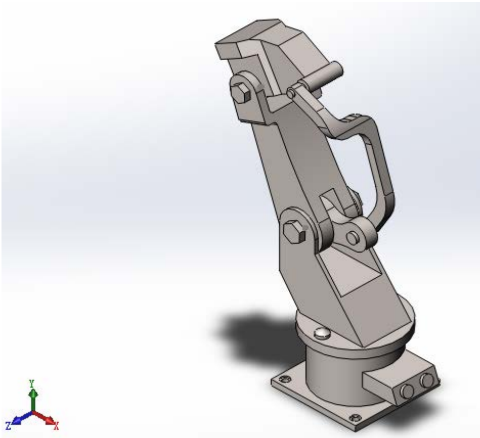


Figure 5-7 BOM of the mechanical arm



(a)



(b)

Figure 5-8 (a) Subassembly 1, (b) Subassembly 2

The graph representation of this mechanical arm is shown in Figure 5-9. Each component is represented by a circle with a component number inside. If two components are contacted and constrained, they are connected by a solid line. The solid line with an arrow denotes the constraint from fasteners which can be removed by non-destructive disassembly operations. The dash line with an arrow indicates the constraint from fasteners which can only be removed by a destructive way. All fasteners must be removed before disassembling components.

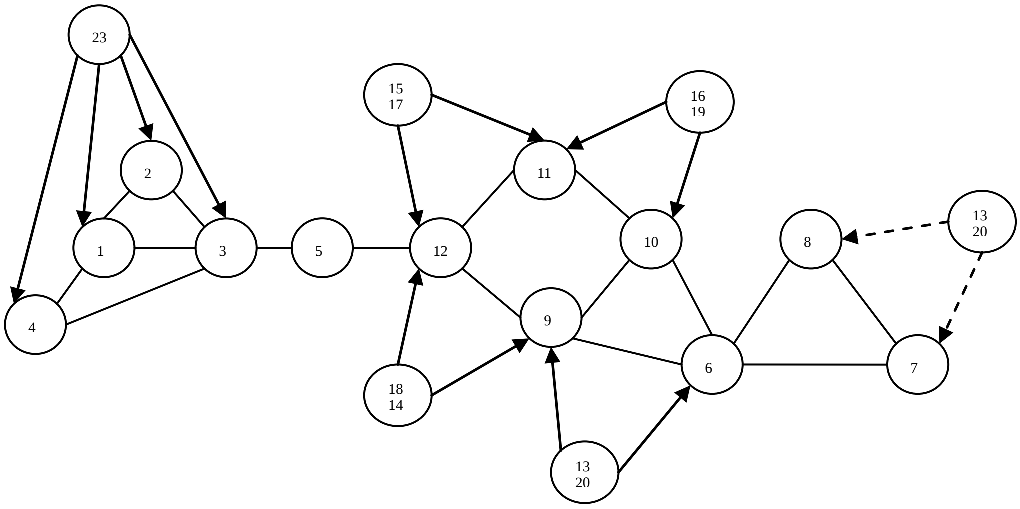


Figure 5-9 Graph representation of the mechanical arm

Multi-layer constraint matrices can be built based on product's BOM and graph representation. Basic knowledge of the multi-level constraint matrices has been introduced in Chapter 3. They represent the constraint relations between pairs of components in six disassembly directions. Fasteners are not included in the constraint matrices. Therefore, only component 1 to component 12 are showed in the multi-level constraint matrices. The constraint matrices of the product and two first-level subassemblies are built as shown in Figure 5-10.

$$
\begin{array}{c|cc}
 & FS1 & FS2 \\
\hline
FS1 & 000000 & 101111 \\
FS2 & 101111 & 000000
\end{array}
$$

(a)

$$
\begin{array}{c|ccccc}
 & 1 & 2 & 3 & 4 & 5 \\
\hline
1 & 000000 & 001000 & 101100 & 000100 & 100000 \\
2 & 000100 & 000000 & 000100 & 000100 & 000101 \\
3 & 011100 & 001000 & 000000 & 000100 & 101111 \\
4 & 001000 & 001000 & 001000 & 000000 & 001010 \\
5 & 010000 & 001010 & 011111 & 000101 & 000000
\end{array}
$$

(b)

	6	7	8	9	10	11	12
6	000000	000100	001011	001011	101000	001000	001000
7	001000	000000	001000	001000	001000	001000	001000
8	000111	000100	000000	100111	100011	101000	111011
9	000111	000100	011011	000000	100001	001000	000100
10	010100	000100	010011	010010	000000	111011	010000
11	000100	000100	010100	000100	110111	000000	010011
12	000100	000100	110111	001000	100000	100011	000000

(c)

Figure 5-10 (a) Constraint matrix of the product

(b) Constraint matrix of subassembly 1

(c) Constraint matrix of subassembly 2

Figure 5-11 shows constraints between fasteners and components. This is a 12×11 matrix. The row and column elements represent parts and fasteners, respectively. “1” means that the component is constrained by a fastener that can be removed by non-destructive disassembly operations while “0” represents there is no constraint between the component and fastener. “2” shows the constraints can only be released by destructive operations. For example, the element in the 6nd row and the 8th column is “1”, which means components 6 is connected by fastener 8. The fastener 8 should be removed before disassembling component 6. The element in the 7nd row and the 9th column is “2”, which means the constraint can only be removed by destructive operations. In addition, some components are connected by more than one fastener. The components can be disassembled only when the fasteners are all removed.

	$F1$	$F2$	$F3$	$F4$	$F5$	$F6$	$F7$	$F8$	$F9$	$F10$	$F11$
$C1$	0	0	0	0	0	0	0	0	0	0	1
$C2$	0	0	0	0	0	0	0	0	0	0	1
$C3$	0	0	0	0	0	0	0	0	0	0	1
$C4$	0	0	0	0	0	0	0	0	0	0	0
$C5$	0	0	0	0	0	0	0	0	0	0	0
$C6$	1	0	0	0	0	0	0	1	2	2	0
$C7$	0	0	0	0	0	0	0	0	2	2	0
$C8$	0	0	0	0	0	0	0	0	2	2	0
$C9$	1	1	0	0	0	1	0	1	0	0	0
$C10$	1	0	0	0	0	0	0	1	0	0	0
$C11$	0	0	1	1	1	0	1	0	0	0	0
$C12$	0	1	1	0	1	1	0	0	0	0	0

Figure 5-11 Fastener-component matrix of the mechanical arm

Some fasteners can be disassembled by both non-destructive and destructive operations such as bolts and nuts *etc*. The difference is that the disassembly tools and time are different. But certain fasteners are required to be removed by destructive operations. For example, in Table 5-3, fasteners F9 and F10 are rivets and the corresponding non-destructive time is N/A. Therefore, they can only be disassembled by destructive methods and the required time is 25s for each rivet.

As the mechanical arm is disassembled into first-level subassemblies first, the search process also starts from the first-level constraints matrix. Constraint matrices are constructed as shown in Figure 5-10. Because subassembly 2 contains the target component, subassembly 2 should be disassembled in the first step. After subassembly 2 is removed, a following disassembly operation should be performed to get the target component 8 in the low-level matrix.

Traversal algorithm is used to find all feasible disassembly sequences. Disassembly time is employed to evaluate the sequences and generate the optimal one. For example, if subassembly 2 is disassembled as a whole, a following search should be performed to get the target component.

First, part numbers (6, 7, 8, 9, 10, 11, 12) are made a full permutation. All sequences are ended up by target component 8. For example, the sequence (6, 7, 11, 8, 9, 10, 12) will be recorded as (6, 7, 11, 8). Many permutated sequences are infeasible and they will be skipped by the disassembly feasibility check using the constraint matrices in Figure 5-10. According to Formula 3.1 ($a_{id} = a_{i1d} + a_{i2d} + ... + a_{ind}$), if $a_{id} = a_{i1d} + a_{i2d} + ... + a_{ind} = 0$, $d \in (\pm X, \pm Y, \pm Z)$, component *i* can be removed. Otherwise, component *i* is fixed and cannot be disassembled. The disassembly feasibility check starts from the first component in the permutated sequence. Take the sequence (6, 7, 11, 8) as an example, component 6 is examined by the constraint matrix in Figure 5-10 (c). It is found that $a_{6\text{-x}} = a_{61\text{-x}} + a_{62\text{-x}} + a_{63\text{-x}} + a_{64\text{-x}} + a_{65\text{-x}} + a_{66\text{-x}} = 0$. Therefore, component 6 can be removed along –X direction. At the same time, a following check with fasteners' constraints is performed using Figure 5-11. All fasteners should be removed before components. For component 6, fasteners (21, 22, 13, 20) should be removed before the component. Destructive disassembly time and non-destructive disassembly time are shown in Table 5-3. After component 6 and corresponding fasteners are removed, component 7 will be examined in the next step. The disassembly sequence is feasible only when all components can be removed. The sequence will be skipped when any component fails in the feasibility check.

When a disassembly sequence of the first-layer subassembly is generated, the same method will be used to search the lower-level matrices with the target component while other matrices will be skipped. After searching operations of each level's matrices are finished, sequences of disassembling each level's subassemblies will be combined together. By evaluating all feasible disassembly sequences, an optimal sequence can be selected. The DSP process can be concluded as follows:

Step 1: determine a target component, such as C8;

Step 1: search for the lowest subassembly (FS2) which contains the target component;

Step 2: generate all feasible disassembly sequence using the multi-level constraint matrix (Figure 5-10) and the fastener-component matrix (Figure 5-11);

Step 3: compare feasible sequences using predefined criteria (Table 5-3);

Step 4: output the optimal disassembly sequence.

An optimal disassembly sequence can be generated as $FS2 \to 21 \to 22 \to 7 \to 8$ using the destructive disassembly method. The main code is shown in Appendix 2. According to Table 5-3 and Figure 5-9, component 7 is fixed by rivets and could not be disassembled by a non-destructive operation. Therefore, a destructive disassembly operation is used to remove the riveting connection. Components 21 and 22 (rivets) are removed by a destructive way and the rest components are disassembled by non-destructive operations. Based on the disassembly cost listed in Table 5-3, the disassembly cost of this sequence is calculated as $18+20+20+15+8=81\text{s}$.

5.6.2 MP4 player

Figure 5-12 shows an MP4 player. Its battery needs to be recycled at the end-of-life (EOL) stage. A destructive disassembly method is used to find a short path to reduce the disassembly cost.

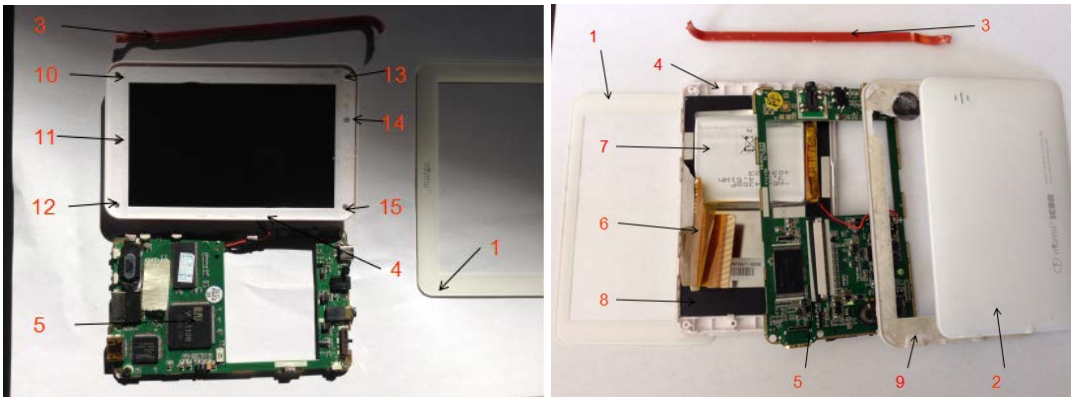


Figure 5-12 An MP4 Player of the EOL stage

This MP4 player has 9 components and 7 fasteners. Their information is shown in Table 5-5. Component 7 (battery) is selected as the target component to be recycled.

No.	**Component name**	**Quantity**	**Time Non-destructive**	**Time destructive**	**Constraint type**
1(C1)	Screen Cover	1	N/A	13s	Adhesive
2(C2)	Back cover	1	N/A	19s	Adhesive (D)
3(C3)	Frame	1	17s	12s	Buckling
4(C4)	Screen frame	1	N/A	25s	Fastener+adhesive +buckling (D)
5(C5)	Circuit board	1	21s	40s	Mating+buckling
6(C6)	Screen cable	1	N/A	33s	Welding (D)
7(C7)	Battery	1	20s	10s	Mating+buckling

8(C8)	Screen	1	N/A	17s	Adhesive+welding (D)
9(C9)	Back frame	1	N/A	26s	Adhesive+buckling (D)
10(F1)	Screw	1	11s	18s	Fastener
11(F2)	Screw	1	11s	18s	Fastener
12(F3)	Screw	1	11s	18s	Fastener
13(F4)	Screw	1	11s	18s	Fastener
14(F5)	Screw	1	11s	18s	Fastener
15(F6)	Screw	1	11s	18s	Fastener
16(F7)	Tape	N/A	N/A	11s	Gluing(D)

Table 5-4 Components' information of the MP4

The graph representation of this MP4 player is shown in Figure 5-13.

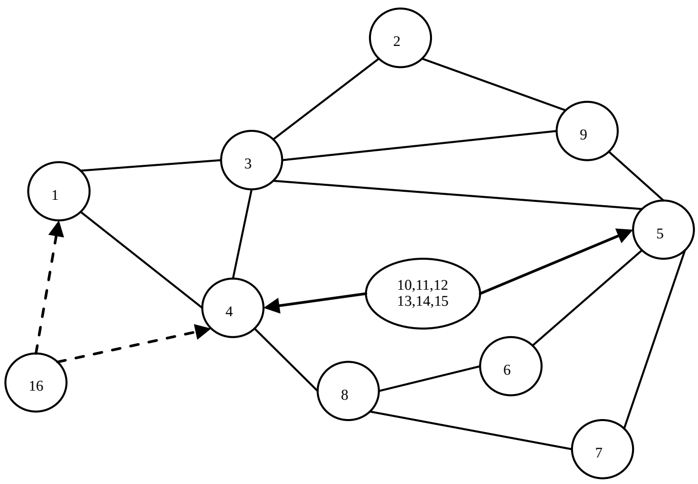


Figure 5-13 Graph representation of the MP4 player

The constraint matrix of this MP4 player is shown in Figure 5-14.

	$C1$	$C2$	$C3$	$C4$	$C5$	$C6$	$C7$	$C8$	$C9$
$C1$	000000	000001	011111	111101	000001	000001	000001	000001	000001
$C2$	000010	000000	111110	000010	000010	000010	000010	000010	000010
$C3$	100000	111101	000000	101100	101100	000000	101100	101100	101100
$C4$	111110	000001	010000	000000	000001	000001	000001	000001	000001
$C5$	000010	000001	011100	000010	000000	000010	111100	000010	000001
$C6$	000010	000001	000000	000010	000001	000000	000000	111100	000001
$C7$	000010	000001	011100	000010	111100	000000	000000	111100	000001
$C8$	000010	000001	011100	000010	000001	111100	111100	000000	000001
$C9$	000010	000001	011100	000010	000010	000010	000010	000010	000000

Figure 5-14 Constraint matrix of the MP4 player

The fastener-component matrix is constructed as shown in Figure 5-15.

	$F1$	$F2$	$F3$	$F4$	$F5$	$F6$	$F7$
$C1$	0	0	0	0	0	0	2
$C2$	0	0	0	0	0	0	2
$C3$	0	0	0	0	0	0	0
$C4$	1	1	1	1	1	1	2
$C5$	1	1	1	1	1	1	0
$C6$	0	0	0	0	0	0	0
$C7$	0	0	0	0	0	0	0
$C8$	0	0	0	0	0	0	0
$C9$	0	0	0	0	0	0	2

Figure 5-15 Fastener-component matrix of MP4 player

According to Table 5-4, components 1 and 4, components 2 and 9 are connected by adhesive glues. Those connections should be disassembled using destructive operations.

Because the screen cover (component 1) is adhered to the screen frame (component 4) and they can only be removed by destroying the adhesive. The screws are accessible only after the screen cover is removed. Therefore, a non-destructive disassembly cannot be applied to this product. The MP4 player can only be disassembled by a destructive way.

Using the same disassembly sequence searching algorithm, the objective destructive disassembly sequence is generated as: $C1 \to (\mathrm{F}1, \mathrm{F}2, \mathrm{F}3, \mathrm{F}4, \mathrm{F}5, \mathrm{F}6) \to C3 \to C4 \to C5 \to C7$. Components 1 and 3 are disassembled with destructive operations and the rest components are removed by a non-destructive way. The total disassembly time for the optimal disassembly sequence is $13 + 6 \times 11 + 12 + 25 + 21 + 20 = 157\mathrm{s}$ based on the data in Table 5-4.

In conclusion, this MP4 player has to be disassembled by a destructive way because of many adhesive connections. Non-destructive disassembly methods cannot get the target component. Therefore, the design-for-disassembly (DFD) conception may be considered to improve the design for disassembly of this product. This is a topic of the future work

5.7 Summary

In this chapter, a destructive method is proposed to generate the disassembly sequence with the less cost for selective disassembly. In non-destructive disassembly operations, the UTD components have to be bypassed and this may lead to a longer disassembly sequence to the target component. However, in destructive disassembly operations, the UTD components are identified and removed by destructive ways. It can shorten the disassembly path and reduce the total disassembly cost. Multi-level constraint matrices are constructed to identify constraints between components. Fastener-component matrix is built to show the constraints between components and fasteners. Those type D constraints are identified in the matrix. By removing type D constraints using destructive ways, the detour problem can be solved and the generated sequences are practical to guide the disassembly operation.

The disassembly cost is calculated for both destructive disassembly method and non-destructive method. The method with the less disassembly cost is selected. Case studies of a mechanical arm and an MP4 player are used to verify this method. The results show that by using the destructive method, a target component can be removed with less disassembly cost in selective disassembly operations.

Chapter 6

Conclusion and Future work

6.1 Research contribution

When products come to their End-of Life stage, reusable components and materials can be recycled for resources economically and environmental friendly. An efficient disassembly sequence can help to remove the target component with less disassembly time and labor cost. In this thesis, an optimized method is proposed to generate an optimal or near-optimal disassembly sequence for selective disassembly operations.

First, based on product's BOM, multi-level constraint matrices are constructed to show the target component's hierarchical position. Compared to one-layer constraint matrix, the proposed multi-level constraint matrices can reduce the searching size and improve the efficiency of sequence planning.

Second, the disassembly feasibility check is implemented in the genetic algorithm. All infeasible disassembly sequences generated by crossover and mutation operations are skipped. A comparison is made between the GA with and without the disassembly feasibility check. The result shows that the GA with the disassembly feasibility check can generate a better solution with a shorter time.

In addition, the destructive disassembly is considered to remove the constraints such as riveting and adhesive. A fastener-component matrix is constructed to identify constraints' types. Two case studies are used to verify this destructive disassembly method that can reach the target component with less cost.

6.2 Future work

Even though the proposed methods have improved the efficiency of disassembly sequence planning, there are still some aspects to be further studied as follows.

The evaluation criteria should take more factors into consideration. Especially for destructive disassembly operations, electric tools will bring dust and noisy pollutions etc. Those invisible factors are detrimental to the environment and human health. Therefore, they should be considered when evaluating the disassembly sequences.

In addition, the disassembly sequences planning research is not an isolated research topic. It should be considered during the design stage as design for disassembly.

References

An H., Chen S., Huang H. (2015). “Simultaneous optimization of stacking sequences and sizing with two-level approximations and a genetic algorithm”. Composite Structures, v 123, p.180.

Basdere B., Seliger G. (2003). “Disassembly factories for electrical and electronic products to recover resources in product and material cycles”. Environmental Science and Technology, v 37, p.5354-5362.

Behdad S., Thurston D. (2012). “Disassembly and reassembly sequence planning tradeoffs under uncertainty for product maintenance”. Journal of Mechanical Design Transactions of ASME, v 134, n 4, p.45.

Benedetto L., Klemes J. (2010). “The environmental bill of material and technology routing: an integrated LCA approach”. Clean Technologies and Environment Policy, April 2010, v 12, Issue 2, p.191-196.

Berg L., Behadad S., Judy V. (2015). “Disassembly Sequence Evaluation: A User Study Leveraging Immersive Computing Technologies”. Journal of Computing and Information Science in Engineering, v 15, p.127-129.

Bhattacharya A., Dey P. (2015). “Green manufacturing supply chain design and operations decision support”. International Journal of Production Research, p.1-5.

Dong B., Zhang R. (2013). “Disassembly Sequence Planning Based on Synchronous Net”. Computational Intelligence and Design, v 2, p.297-300.

Bras B., Sundin E. (2004). “Making functional sales environmentally and economically beneficial through product remanufacturing,” Journal of Cleaner Production, v 13, Issue 9, p. 913–925.

Cao, P. B., Xiao, R. B (2007). “Assembly planning using a novel immune approach”, The

International Journal of Advanced Manufacturing Technology, p.770-782.

Cappelli F., Delogu M., Pierini M., Schiavone F. (2007). “Design for disassembly: a methodology for identifying the optimal disassembly sequence”. Journal of Engineering Design, v 18, n 6, p.563-575.

Chen X., Lou P. (2015). “Product disassembly sequence planning based on partial destruction mode”. Journal of Nanjing University of Aeronautics & Astronaut, v 46, n 6, p.945-950.

Chen Y., Chen W. (2014). “Product disassembly sequence optimization based on profit-probability under uncertain environment”. Computer Integrated Manufacturing Systems, CIMS, v 20, n 4, p.793-798.

Dong T., Zhang L., Tong R., Dong J. (2006). “A hierarchical approach to disassembly sequence planning for mechanical product”. The International Journal of Advanced Manufacturing Technology, v 30(5), p.507-520.

Elsayed A., Kongar E., Gupta S., Sobh T. (2012). “A robotic-driven disassembly sequence generator for end-of-life electronic product”. Journal of Intelligent and Robotic Systems: Theory and Applications, v 68, n 1, p.43-52.

Elif K., Suendra M. (2005). “Disassembly sequencing using genetic algorithm”. The International Journal of Advanced Manufacturing Technology, v 30, p.497-506.

Federico C., Massimo D., Marco P. (2008). “Design for disassembly: a methodology for identifying the optimal disassembly sequence”. Journal of Engineering Design, v 18, issue 6, p.75-79.

Fongondo O., Williams D. (2010). “How are WEEE doing? A global review of the management of electrical and electronic wastes”. Waste Management, v 31, Issue 4, p. 714–730.

Gao J., Dong X., Chen H., Duan G. (2003). “Disassembly AND/OR graph model for disassembly for recycling”. IEEE International Symposium on Electronics and the Environment, p. 54-59.

Giudice, F. (2010). “Disassembly depth distribution for ease of service: a rule-based approach”. Journal of Engineering Design, v 21(4), p.1159-1166.

Govindan, K., Kannan D., Shankar M. (2014). “Evaluation of green manufacturing practices using a hybrid MCDM model combining DANP with PROMETHEE”. International journal of Production Reaserch, p.19-23.

Godipa T., Wahab R. (2011). “Design Framework for End-on-life Vehicles Recovery: Optimization of Disassembly Sequence Using Genetic Algorithms”. American Journal of Environmental Sciences, v 6, n 4, p.350-356.

Han H.J., Yu, J.M., Lee D.H. (2013). “Mathematical model and solution algorithms for selective disassembly sequencing with multiple target components and sequence-dependent setups”. International Journal of Production Research, v51, n 16, p.4979-5010.

Han J. (2007). “Research on disassembly sequence planning based on genetic algorithms”. Master thesis, Hangzhou University of Science and Technology.

Hassanzadeh R., Eslahchi C., Sung W. (2012). “Constructing phylogenetic supernetworks based on simulated annealing”. Molecular Phylogenetics and Evolution, v 63(3), p.56-59.

Hoang N., Langseth M., Porcaro R., Hanssen A. (2011). “The effect of the riveting process and aging on the mechanical behavior of an aluminum self-piercing riveted connection”. European Journal of Mechanics, v 30, n 5, p.619-630.

Homem de Mello, L.S (1990). “AND/OR graph representation of assembly plan” *IEEE Transactions on Robotics and Automation* 6 (2), p.188-199.

Jiang J., Wang Y. (2013). "Disassembly sequence planning based on extended interference matrix and genetic algorithm". Computer Engineering and Design, v 34, n 3, p.1064-1068.

Kai X., Liang G., Li W. (2014). "Disassembly sequence planning using a Simplified Teaching-Learning-Based Optimization Algorithm". Advanced Engineering Informatics, v 28, n 4, p.518-527.

Kai X., Gao C., Wang L., Li W. (2013). "A Simplified Teaching-Learning-Based Optimization Algorithm for Disassembly Sequence Planning". E-Business Engineering, 2013 IEEE 10th International Conference, p.393-398.

Kalayci C., Gupta S. (2013). "Ant colony optimization for sequence-depend disassembly line balancing problem". Jounal of Manufacturing Technology Management, v 24, n 3, p. 413-427.

Kashkoush M., Elmaraghy H. (2014). "Product design retrieval by matching bills of materials". Jounal of Mechanical Design, Transaction of the ASME, v 136, n 1, p.36-42.

Kara S., Pornprasitpol P., Kaebernick H. (2006). "Selective disassembly sequencing: a methodology for the disassembly of end-of-life products". CIRP Annals-Manufacturing Technology, v (55), p. 37-40.

Keung K.W., Lp W.H., Lee T.C. (2001). "The solution of a multi-objective tool selection model using the GA approach." The International Journal of Advanced Manufacturing Technology, v 18, p. 771-777.

Koc A., Sabuncuoglu I., Erel E. (2009). "Two exact formulations for disassembly line balancing problems" with task precedence diagram construction using an AND/OR graph". IIE Transactions, v 41, n 10, p.866-881.

Kuo T., Zhang H., Huang S. (2000). “Disassembly analysis for electromechanical products: A graph-based heuristic approach”. International Journal of Production Research, v 38, issue 5, p154-155.

Lambert A.J.D. (2003). “Disassembly sequencing : a survey”. International Journal of Production Research, v41, p. 3721-3759.

Lambert A.J.D., Gupta S.M. (2005) Disassembly modeling for assembly, maintenance, reuse, and recycling, CRC Press, Boca Raton, Florida.

Lee K., Gadh R. (1996). “Computer aided design for disassembly: a destructive approach”. Proceedings of the 1996 IEEE International Symposium on Electronics and the Environment, p.173-178.

Li J., Li P. K. (2002). “A novel representation scheme for disassembly sequence planning”. *The Internatianl Journal of Advanced Manufacturing Technology*, v 20(8), p.621-630.

Li J., Li P. K. (2006). “Generation of possible multiple components disassembly sequence for maintenance using a disassembly constraint graph.” International journal economics, p. 51-65.

Li W., Peng Q., Xing M. (2011). Bioreactor improvement based on design for assembly in virtual environments. ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, p.733-742.

Li S., Zhu, B. (2008). “Research on Virtual Disassembly Simulation Based on Constraint Matrix”. Power Electronics and Intelligent Transportation System, Workshop, p. 587-591.

Li L., Xie Z., Yang X. (2011). “Research on Disassembly Sequence Planning Algorithm Based on Interference”. Ship Electronic Engineering, v 31, n 7, p.148-152.

Liu Y., Li S., Wang, J. (2014). “Interactive disassembly approach for assembly sequence planning based on product hierarchical classification”. Computer Integrated Manufacturing systems, CIMS, v 20, n 4, p. 787-792.

Liu A., Zhou Z., Li M. (2012). “Disassembly sequence planning based on Genetic Algorithm”. Equipment and Manufacturing Technology, v5, p.1672-545.

Lotfi S., Deja J., Rem P., Mroz R., Roekel V. (2014). “Mechanical recycling of EOL concrete into high-grade aggregates”. Resources, Conservation and Recycling, v 87, p. 117-125.

Lu, C., Wong, Y., Fuh, J. (2006). “An enhanced assembly planning approach using a multi-objective genetic algorithm”. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, v 220(2), p.255-272.

Maroua K., Moez T., Nizar A. (2014). “Disassembly sequence planning based on a genetic algorithm”. Journal of Mechanical Engineering Science, v 229, p.2281-2290.

Mascle C, Balasoiu B (2003). “Algorithmic selection of a disassembly sequence of a component by a wave propagation method”. Robotics and Computer-Integrated Manufacturing, v19, p168-175.

McGovern S.M., Gupta S. (2006). “Ant colony optimization for disassembly sequencing with multiple objectives”. The international Journal of Advanced Manufacturing Technology, v 30(5), p. 481-496.

Mitrouchev P, Wang C (2015). “Selective disassembly sequence generation based on lowest level disassembly graph method”. The International Journal of Advanced Manufacturing Technology, v80, p.141-159.

Min L., Lai J., Shen W. (2013). “A method for transformation of Engineering bill of materials to maintenance bill of materials”. Robotics and Computer-Integrated Manufacturing, v 30, n 2, p.142-149.

Min S., Zhu X. (2010). “Mechanical Product Disassembly and/or Graph Construction”. 2010 International Conference on Measuring Technology and Mechatronics Automation, v 2, p. 627-631.

Pachauri A., Mishara G., (2015). “A path and branch based approach to fitness computation for program test data generation using genetic algorithm”. 2015 1st International Conference on Futuristic Trends in Computational Analysis and Knowledge Management, p 49-55.

Pan X., Song X. (2012). “Electromechanical product disassembly sequence planning based on partial destruction mode”. Computer Integrated Manufacturing Systems, v 18, n 5, p174-179.

Pak, K.G., Sodhi R. (2002). “Destructive disassembly of bolts and screws by impact fracture”. Journal of Manufacturing Systems, v 21, n 4, p. 316-324.

Pornsing C., Watanasungsuit A. (2014). “Discrete particle swarm optimization for disassembly sequence planning”. Management of Innovation and Technology (ICMIT), IEEE International Conference, p. 480-485.

Pomares, J., Puente, S., Torres, F. (2004) “Virtual disassembly of products based on geometric models”. *Computers in Industry*, v 55(1), p.1-14.

Reap J., Bras B. (2002). “Design for disassembly and the value of robotic semi-destructive disassembly”. ASME 2002 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, v 3, p. 275-281.

Richter S., Nagel T., Fraulob S. (2010). “Drive engineering. Toothed belt drives. Small parts to allow new designs. Miniature gears in the engineering of devices”. Mechatronik, v 118, n 4, p.18-20.

Rickli J., Camelio J. (2014). “Partial disassembly sequencing considering acquired end-of-life product age distribution”. International Journal of Production Researh, v 52, n 24, p. 7496-7512.

Riggs R., Jin X. (2015). “Two-stage sequence generation for partial disassembly of products with sequence dependent task times”. 22nd CIRP Conference on Life Cycle Engineering, v 29, p. 698-703.

Rodriguez F., Keribar A., Armbruster J. (2007). “A versatile tool for combined dynamic analysis of chain and gear drive systems”. VDI Berichte, n 1997, p. 213-233, ISSN: 00835560; ISBN-10: 3180919973, ISBN-13: 9783180919973.

Rusinko, C.A. (2007). “Green Manufacturing: An Evaluation of Environmentally Sustainable Manufacturing Practices and Their Impact on Competitive Outcomes”. Engineering Management, IEEE Transactions v 54, p. 445-454.

Shana S.S., Chen W. (2011) “Rule-based recursive selective disassembly sequence planning for green design”. Advanced Engineering Informatics, v 25, p. 77–87.

Shana S.S., Chen W. (2012). “Disassembly sequence structure graph: An optimal approach for multiple-target selective disassembly sequence planning”. *Advanced Engineering informatics*, v26(2), p. 306-316.

Sharma P., Saroj (2015). “Discovery of classification rules using distributed genetic algorithm”. Procedia Computer Science, v 46, p. 276-284.

Satou H., Hashimoto A. (2009). “Study on structural behavior of friction type high strength bolted joints having gap in the contact plane”. Journal of Structural and Construction Engineering, v

74, n 644, p. 1869-1878.

Saranya M., Mohan A.K. (2015). “Algorithm for enhanced image security using DNA and genetic algorithm”. 2015 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES), p.1-5.

Schneider M., Grahl J., Francas D., Vigo D. (2013). “A problem-adjusted genetic algorithm for flexibility design”. International Journal of Production Economics, v 141, p. 55-56.

Shih, H.M (2014). “Migrating product structure bill of materials Excel files to STEP PDM implementation”. International Journal of Information Management, v 34, n 4, p.489-516.

Song X., Zhou W. (2013). “Disassembly sequence planning for electro-mechanical product under a partial destructive mode”. Assembly Automation, v34, n 1, p. 106-114.

Song X., Zhou W., Pan X., Feng K. (2014). “Disassembly sequence planning for electro-mechanical products under a partial destructive mode”. Assembly Automation, v 12, p 77-81.

Sun, X., Chu X., Su Y., Tang C. (2010). “A new directed graph approach for automated setup planning in CAPP. International Journal of Production Research, v 48, p. 22-25.

Tian G., Zhou M., Chu J. (2013). “A chance constrained programming approach to determine the optimal disassembly sequence”. IEEE Transactions on Automation Science and Engineering, v 10, n 4, p. 1004-1013.

Tseng Y.J., Kao H., Huang F. (2010). “Intergrated assembly and disassembly sequence planning using a GA approach”. International Journal of Production Reasearch, v 48(20), p. 5991-6013.

Umeda Y., Miyaji N., Shiraishi Y., Fukushige S. (2015). “Proposal of a design method for semi-destructive disassembly with split lines”. Manufacturing Technology, v 16, p 246-251.

Van W., Jong G., Nijland, H. “Accelerating car scrappage: A review of research into the environmental impacts”. Transport Reviews, **v** 31(5), p. 549-569.

Venkatesh M., Zhou J. (2009). Mechanical and Electronics Engineering proceedings of the International Conference on ICMEE 2009. International Conference on Mechanical and Electronics Engineering.

Viganò R., Gómez G. (2013). “Automatic assembly sequence exploration without precedence definition”. International Journal on Interactive Design and Manufacturing, v 7(2), p. 79-89.

Walther, G., Steinborn, J., Spengler, T. (2010). “Implementation of the WEEE-directive—economic effects and improvement potentials for reuse and recycling in Germany”. The International Journal of Advanced Manufacturing Technology, 2010, v 47(5-8), p. 461-474.

Wang W., Shi X. (2014). “Disassembly sequence planning based on poisoning ant colony algorithm”. Key Engineering Material, v 572, n 1, p. 340-343.

Xie, Y., Huang, M., Zhong Y., Kuang B. (2007). “Disassembly sequence planning based on the simulated annealing and genetic algorithm”. Mechanical Engineer, v1, p 44-48.

Xue J., Qian S., Zhang Y. (2010). “Disassembly sequence planning based on ant colony optimization algorithm”. Bio-Inspired Computing: Theories and Application. IEEE fifth International Conference, p.1125-1129.

Yeh W., Wei S. (2013). “Simplified swarm optimization in efficient tool assignment of disassembly sequencing problem”. 2013 IEEE Congress on Evolution Computer, p. 2712-2719.

Zhang G, Deng K (2006). “Research on assembly modeling and sequence planning based on features”. *International Technology and Innovation Conference*, p.408-411.

Zhang L., Liu Z., Yang M. (2011). “Disassembly sequence planning based on interpretative structural model”. Journal of Computer-Aided Design and Computer Graphics, v 23, p 667-675.

Zhang W., Su Q., Liu P. (2013). “Research on Virtual Maintenance Disassembly Sequence Intelligent Planning”. Journal of System Simulation, v 25, n 8, p.1912-18.

Zhang X., Yu G. (2015). “Parrallel disassembly sequence planning for complex product based on genetic algorithm”. Journal of Computer-Aided Design and Computer Graphics, v 27, p 1327-1333.

Zhou L, Xiang D., Duan G. (2008). “Disassembly stability planning method in and or graph model”. Microcomputer Information, n 6, p.1-3.

Zhou W. (2014). “Design for disassembly based on destructive disassembly sequence planning”. Master thesis, Zhejiang University, 2014.

Zhou X., Huang K., Xi L. (2015). “Preventive maintence modeling for multi-component systems with considering stochastic failures and disassembly sequence”. Reliablility Engineering and System Safety, v142, p 231-237.

Zhu B., Sarigecili M. (2013). “Disassembly information model incorporating dynamic capabilities for disassembly sequence generation”. Robotics and Computer Integrated Manufacturing, v 29, p 396-400.

Appendix 1

```
D=ones(24,4)

A=[1 2 3 4]
B=perms(A)
for m=1:1:4*3*2
        C=[0 0 0 0 0 0 1 1 1 1 0 1 1 1 1 1 0 1 0 0 0 0 0 1;1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 1 0 
1;1 1 1 1 1 0 1 0 1 1 0 0 0 0 0 0 0 0 1 1 1 1 0 1;0 0 0 0 1 0 1 1 1 1 1 0 1 1 1 1 1 0 0 0 0 0 0 0]

    for n=1:1:4
        i=B(m,n)
          for j=1:1:6
                 sum(i,j,m)=C(i,j)+C(i,6+j)+C(i,12+j)+C(i,18+j)
                     if sum(i,j,m)==0
C(:,6*(i-1)+1)=0;C(:,6*(i-1)+2)=0;C(:,6*(i-1)+3)=0;C(:,6*(i-1)+4)=0;C(:,6*(i-1)+5)=0;C(:,6*(i-1)+6)=
0
                         D(m,n)=sum(i,j,m)
                         direction(m,n)=j
                         break
                     end
           end
    end
end


for k=1:1:24
    if find(D(k,:)~=0)
            B(k,:)=0
        direction(k,:)=0
    end
end

B(find(B(:,3)==0),:)=[]
direction(find(direction(:,3)==0),:)=[]
direction(:,4)=[]

[mmax,nmax]=size(B)
[imax,jmax]=size(direction)

T(1)=10;T(2)=20;T(3)=30;T(4)=40;
```

```
k=3
for m=1:mmax
    Total(m)=0
    for n=1:find(B(m,:)==k)
        Total(m)=Total(m)+T(B(m,n))
    end
end
Total=Total'

c=find(Total==min(Total))
best_order=B(c,:)

angle_matrix=[0 180 90 90 90 90;180 0 90 90 90 90;90 90 0 180 90 90;90 90 180 0 90 90;90 90 90 90
0 180;90 90 90 90 180 0]
for m=1:mmax
    Total_angle(m)=0
    for j=1:2
      Total_angle(m)=Total_angle(m)+angle_matrix(direction(m,j),direction(m,j+1))
    end
end

Total_angle=Total_angle'
d=find(Total_angle==min(Total_angle))
best_order2=B(d,:)
best_direction=direction(d,:)
```

Appendix 2

```
D=ones(7*6*5*4*3*2*1,7)
A=[6 7 8 9 10 11 12]
B=perms(A)
for m=1:1:7*5*6*4*3*2
      C=[0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 0 1 1 0 0 1 0 1 1 1 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0;
         0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0;
         0 0 0 1 1 1 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 1 1 1 0 0 0 1 1 1 0 1 0 0 0 1 1 1 0 1 1;
         0 0 0 1 1 1 0 0 0 1 0 0 0 1 1 0 1 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0;
         0 1 0 1 0 0 0 0 0 1 0 0 0 1 0 0 1 1 0 1 0 0 1 0 0 0 0 0 0 0 1 1 1 0 1 1 0 1 0 0 0 0;
         0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 1 0 0 1 1 0 1 1 1 0 0 0 0 0 0 0 1 0 0 1 1;
         0 0 0 1 0 0 0 0 0 1 0 0 1 1 0 1 1 1 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 1 0 0 0 0 0]

     for n=1:1:7
         i=B(m,n)
           for j=1:1:6
               sum(i,j,m)=C(i,j)+C(i,6+j)+C(i,12+j)+C(i,18+j)+C(i,24+j)+C(i,30+j)+C(i,36+j)
                 if sum(i.j,m)==0


C(:,6*(i-1)+1)=0;C(:,6*(i-1)+2)=0;C(:,6*(i-1)+3)=0;C(:,6*(i-1)+4)=0;C(:,6*(i-1)+5)=0;C(:,6*(i-1)+6)=
0
                     D(m,n)=sum(i,j,m)
                     break
                 end
           end
     end
end

for k=1:1:7*6*5*4*3*2*1
    if find(D(k,:)~=0)
        b(k,:)=0
    end
end
B(find(B(:,3)==0),:)=[]

FC=[0 0 0 0 0 0 0 0 0 0 1;
    0 0 0 0 0 0 0 0 0 0 1;
    0 0 0 0 0 0 0 0 0 0 1;
    0 0 0 0 0 0 0 0 0 0 0;
    0 0 0 0 0 0 0 0 0 0 0;
    1 0 0 0 0 0 0 1 2 2 0;
```

```
    0 0 0 0 0 0 0 0 2 2 0;
    0 0 0 0 0 0 0 0 2 2 0;
    1 1 0 0 0 1 0 1 0 0 0;
    1 0 0 0 0 0 0 1 0 0 0;
    0 0 1 1 1 0 1 0 0 0 0;
    0 1 1 0 1 1 0 0 0 0 0]

for i=1:1:12
    for j=1:1:11
        p=FC(i,j)
        if FC(i,j)~=0
            BB=[B(1:(i-1)) j+12 B(i:end)]
        end
    end
end
```