

Disassembly matrix for disassembly processes of products

YUAN MAO HUANG^{†*} and CHUN-TING HUANG[†]

The study proposes a feasible and effective method that improves existing methods for computer-aided disassembly planning. The elements of three interference matrices of components in an assembly in the directions of the x , y and z -axes are considered as the right digit, middle digit and left digit, respectively, for the elements of the proposed disassembly matrix in a binary system. By using a Boolean operation or an arithmetic operator with the depth-first-search method, the algorithm is developed and a software program is generated to create all of the possible disassembly sequences and directions for components in a tree diagram. The proposed disassembly matrix provides more possible disassembly processes than the existing disassembly methods to determine preferred disassembly processes. The software program is especially useful for determining preferred disassembly processes for complex devices. This disassembly matrix not only can help disassemble disposable devices, but also can evaluate potential products for ease of disassembly during design.

1. Introduction

A proper design for disassembly is important to facilitate environmental protection and make the maintenance and repair of products more cost-effective. Environmentally protective design or the green design of products links the end of the product's life cycle to its beginning to minimize the harmful impact of the disposed products on the environment by recycling and reuse of the disposed material.

The disassembly processes of some products are the reversal of their assembly processes. Therefore, the reversal of the assembly process can be used as a reference to revise and improve the disassembly sequence for a device. Ko and Lee (1987) used the relationships and mating conditions between components in an assembly to generate a hierarchical tree and investigated the interference between components to generate an assembly sequence. Tonshoff *et al.* (1992) applied a knowledge-based system with the aid of the computer to design products for the automation of assembly. Gottipolu and Ghosh (1997) used an assembly sequence graph to obtain all of the possible assembly sequences. Choi and Zha (1998) utilized an artificial intelligence language to search for preferred assembly sequences under known conditions by eliminating some unreasonable sequences.

Homem *et al.* (1991) described the disassembly problem in terms of inputs and outputs. They illustrated that certain disassembly sequences cannot be derived from simply reversing the assembly sequence. Boothroyd and Alting (1992) focused on design for assembly and disassembly and proposed some methods for analyzing the

Revision received June 2001.

[†] Department of Mechanical Engineering, National Taiwan University, Taipei, Taiwan 10674, Republic of China.

* To whom correspondence should be addressed. e-mail: ymhuang@ccms.ntu.edu.tw

assembly of a product. Arai and Iwata (1993) generated a three-dimensional geometric model of a product and used kinematics simulation to analyse possible directions of motion and required displacements of every component to determine possible disassembly sequences. However, users' judgment must be applied to the system to determine possible disassembly sequences. Gadh (1997) investigated the disassembly of products through de-manufacturing analysis and classified the applications of recycling, refurbishing and reusing.

Expression of the relationships among the components of an assembly is important for generation of possible assembly or disassembly sequences. An inappropriate expression can create undesirable effects. The tree diagram often used for functional designs can also be used to show the structure of an assembly with some sub-assemblies (Pahl and Beitz 1988). Each sub-assembly can then be disassembled into separate groupings and each grouping can further be disassembled into individual components. However, the tree diagram does not provide enough information to generate disassembly sequences. Dini and Santochi (1992) proposed a liaison matrix for replacing the connection graph used in a liaison diagram. This method is similar to the liaison graph method, but available information is not sufficient for disassembling products.

Dini and Santochi (1992) proposed a mathematical model consisting of an interference matrix, a contact matrix and a connection matrix. The interference matrix and the contact matrix were used to obtain the disassembly information from the CAD system. Laperriere and ElMaraphy (1994) proposed a directed graph expression that can quickly generate the relationships among components including the interference relation, the contact relation and the connection relation. They disassembled the original assembly into two parts and generated a simple diagram of the disassembled sequences for the assembly. Nevertheless, their directed graph expression method takes time and manpower because it requires human judgment to evaluate all practical conditions. In addition, this method may create the problem requiring a fixture for disassembly because two components might be disassembled simultaneously.

Gottipolu and Ghosh (1995) proposed a relational model expression that used the relationship between the geometry and the location of components in the assembly to describe all possible theoretical and actual component relationships. However, the relational model expression requires extensively a lot of recorded information and is not suitable for the disassembly of most products. Lambert (1997) utilized a liaison diagram and a disassembly graph to determine the most cost-effective disassembly sequence. However, many possibilities needed to be evaluated and efficiency was low. In addition, due to lack of information about and the unclear relationships between the components, this method is not suitable for disassembling products. Genc *et al.* (1998) discussed a joining technique relying on integral attachment design with snap-fits, actual attachment features, selection of the interface geometry, and an assembly sequence depending on the specific type and number and placement of the attachment features used.

Zussman and Zhou (1999) used a simplified telephone to illustrate the disassembly process planning methodology. Zussman *et al.* (1994) showed all possible disassembly sequences generated by the AND/OR operators in a tree diagram and determined the preferred disassembly sequence with consideration of all values of components. Harjula and Rapoza (1996) evaluated different levels of disassembly to reduce time and costs for production, while maintaining environmental safety.

Gungor and Gupta (1997) defined joint types and directions for components in a disassembly and used weighting factors for methods of disassembly and directions to determine the preferred disassembly sequence and to calculate the total disassembly time. Johnson and Wang (1998) proposed an evaluation method that determined disassembly sequences based on the economic consideration of disposing, recycling, re-manufacturing, and reusing materials after disassembly. In summary, some of these methods can be used to provide insight on disassembly and to help generate a relational model expression.

Most of the researchers focused on the evaluation of disassembly in the context of economical performance, but they neglected to focus on the determination of the optimum disassembly sequences. A good disassembly sequence is very important for a product. Components with interference may not be in contact with each other. Therefore, this study will propose a feasible and effective method for disassembly by introducing a disassembly matrix with the depth-first-search method and will generate a software program. This disassembly matrix will not only provide all the disassembly sequences and directions in the tree diagram, but also will reduce necessary memory compared with that of three interference matrices used by Dini and Santochi (1992). The disassembly matrix will provide all disassembly sequences with one component disassembled at one time to improve the Laperriere and ElMaraphy's directed graph expression and existing methods.

2. Approach

2.1. Interference matrix

The relationship between two components in an assembly is required in order to determine possible disassembly sequences. If the component e_i of an assembly does not interfere with another component e_j in the direction of the k -axis, the component e_i can be disassembled freely from the component e_j in the direction of the k -axis. Consequently, the component e_j can be disassembled from the component e_i in the direction of the $-k$ -axis. If the component e_i that moves in the direction of the k -axis has interference with another component e_j , then the component e_j that moves in the direction of the $-k$ -axis will in turn have interference with the component e_i . The interference matrices in the direction of the k -axis can be generated as (Dini and Santochi 1992):

$$H_k = \begin{matrix} & e_1 & e_2 & \cdots & e_n \\ \begin{matrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{matrix} & \begin{bmatrix} h_{11k} & h_{12k} & \cdots & h_{1nk} \\ h_{21k} & h_{22k} & & \\ \vdots & \vdots & \ddots & \\ h_{n1k} & h_{n2k} & & h_{nmk} \end{bmatrix} \end{matrix} \quad (1)$$

The elements in row e_i of equation (1) stand for the interference relationship of the component e_i with all of the components in the direction of the k -axis, and the elements in column e_j stand for the interference relationship of the component e_j with all of the components in the direction of the $-k$ -axis. If interference exists between components e_i and e_j , the element h_{ijk} in the matrix is 1. Vice versa, the element h_{ijk} in the matrix is 0. The element h_{mmk} is always equal to 0 because no component can have interference with itself.

2.2. Disassembly matrix

The interference matrices in the directions of the x -, y - and z -axes as shown in equation (1) can be combined into one matrix, the disassembly matrix, by using the binary system as

$$A = \begin{array}{c} e_1 \\ e_2 \\ \vdots \\ e_n \end{array} \begin{array}{c} e_1 \\ e_2 \\ \vdots \\ e_n \end{array} \begin{array}{c} \cdots \\ \cdots \\ \cdots \\ \cdots \end{array} \begin{array}{c} e_n \\ \text{BIN} \end{array} \begin{array}{c} \text{result} \\ h_{az}h_{by}h_{cx} \\ h_{dz}h_{ey}h_{fx} \\ h_{gz}h_{hy}h_{lx} \\ h_{mz}h_{oy}h_{px} \\ h_{qz}h_{ry}h_{sx} \\ h_{tz}h_{uy}h_{vx} \end{array} \quad (2)$$

The elements h_{ijz} , h_{ijy} and h_{ijx} in matrices H_z , H_y and H_x , respectively, in equation (1) are the left digit, the middle digit and the right digit, respectively, of the element $h_{ijz}h_{ijy}h_{ijx}$ in the disassembly matrix. If the decimal system is used, the element in the disassembly matrix is $h_{ijz}x2^2 + h_{ijy}x2^1 + h_{ijx}x2^0$ and the disassembly matrix is

$$A = \begin{array}{c} e_1 \\ e_2 \\ \vdots \\ e_n \end{array} \begin{array}{c} e_1 \\ e_2 \\ \vdots \\ e_n \end{array} \begin{array}{c} \cdots \\ \cdots \\ \cdots \\ \cdots \end{array} \begin{array}{c} e_n \\ \text{DEC} \end{array} \begin{array}{c} h_{11z} \cdot 2^2 + h_{11y} \cdot 2 + h_{11x} \\ h_{12z} \cdot 2^2 + h_{12y} \cdot 2 + h_{12x} \\ \vdots \\ h_{n1z} \cdot 2^2 + h_{n1y} \cdot 2 + h_{n1x} \\ h_{n2z} \cdot 2^2 + h_{n2y} \cdot 2 + h_{n2x} \\ \vdots \\ h_{nz} \cdot 2^2 + h_{ny} \cdot 2 + h_{nx} \end{array} \quad (3)$$

The element in the disassembly matrix indicates the interference of one component with the other components of a product. If the right digits of all elements, h_{ijx} , in row e_i are equal to zero, then the component e_i does not have interference in the direction of the x -axis with the component e_j , so the component e_i can be disassembled freely in the direction of the x axis from the component e_j . If the right digit of any element in row e_i is equal to 1, then the component e_i cannot be disassembled in the direction of the x -axis. If the right digits of all elements, h_{ijx} , in column e_j are equal to zero, then the component e_j can be disassembled freely in the direction of the $-x$ -axis. If the right digit of any element in column e_j is not equal to zero, then the component e_j cannot be disassembled in the direction of the $-x$ -axis. Similarly, the middle digits and left digits of all elements in the disassembly matrix are used to investigate the interference of components in the directions of the y and $-y$ -axes, and the z and $-z$ -axes, respectively.

The Boolean operator 'OR' can be used to combine the right, middle and left digits, respectively, of the elements in the disassembly matrix at any row or column to determine whether a component can be freely disassembled in a specific direction. If the right digits of all elements, h_{ijx} , in row e_i , $j = 1, 2, \dots, n$, are equal to 0, then the result of h_{ijx} in this row is equal to 0, and the component e_i can be disassembled freely in the direction of the x -axis. If the right digit of any element, h_{ijx} , in row e_i , $j = 1, 2, \dots, n$, is not equal to 0, then the result of h_{ijx} in this row is equal to 1, and the component e_i cannot be disassembled in the direction of the x -axis. Similarly, the

results of h_{ijy} and h_{ijz} , respectively, $j = 1, 2, \dots, n$, for any row can be determined, and the interference of all components in the directions of the y and z axes can be determined.

If the right digits of all elements, h_{ijx} , in column e_j , $i = 1, 2, \dots, n$, are equal to 0, then the result of h_{ijx} in this column is equal to 0, and the component e_j can be disassembled freely in the direction of the $-x$ -axis. If the right digit of any element, h_{ijx} , in column e_j , $i = 1, 2, \dots, n$, is not equal to 0, then the result of h_{ijx} in this column is equal to 1, and the component e_j cannot be disassembled in the direction of the $-x$ -axis. Similarly, the results of h_{ijy} and h_{ijz} , respectively, $i = 1, 2, \dots, n$, for any column e_j can be determined, and the interference of all components in the directions of the $-y$ and $-z$ -axes can be determined.

The arithmetic operator ‘+’ can also be used to investigate the sum of the left, middle and right digits of elements, respectively, at any row or at any column. If any one of the right digits in row e_i is equal to 1, then the result produced from using the arithmetic operator ‘+’ for the right digits in row e_i is not equal to 0. Therefore, the component e_i cannot be disassembled freely in the direction of the x -axis, which is the result obtained by using the Boolean operator; and so on for the middle and left digits at any row or at any column.

2.3. Generation of disassembly sequence

A state of an assembly with n components is expressed as a node as shown in figure 1. If the component e_i in the assembly can be disassembled in the direction of the $+k$ -axis, then the assembly without the component e_i and the disassembled component e_i in the direction of the $+k$ -axis are added below the previous node as shown in figure 1. The symbol $+k$ within the parenthesis stands for the direction of the axis that the component e_j can be disassembled.

Disassembling the component e_i and eliminating the corresponding row and column for the component e_i from the disassembly matrix, equation (2) becomes

$$A = \begin{matrix} & \begin{matrix} e_1 & e_2 & \dots & e_n \end{matrix} \\ \begin{matrix} e_1 \\ e_2 \\ \vdots \\ e_n \end{matrix} & \left[\begin{array}{cccc} h_{11z}h_{11y}h_{11x} & h_{12z}h_{12y}h_{12x} & \dots & h_{1nz}h_{1ny}h_{1nx} \\ h_{21z}h_{21y}h_{21x} & h_{22z}h_{22y}h_{22x} & & \\ \dots & \vdots & & \dots \\ h_{n1z}h_{n1y}h_{n1x} & h_{n2z}h_{n2y}h_{n2x} & & h_{nnz}h_{nny}h_{nnx} \end{array} \right] \end{matrix} \quad (4)$$

The dashed lines stand for the row and the column eliminated. The Boolean operator ‘OR’ for the simplified disassembly matrix is again used to continue disassembly processes. If there is more than one component at this node and no component can be disassembled further, this node will be eliminated automatically. If only one component exists at this node, it is impossible to disassemble this node any further.

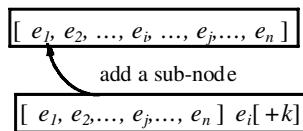


Figure 1. Input sub-node.

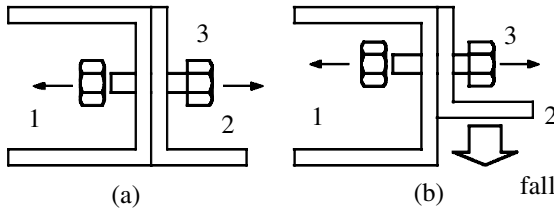


Figure 2. Stability of assembly.

The stability of the other components after one component has been disassembled affects the efficiency of disassembly. Generally speaking, stability is achieved if the other components can maintain their original states after one component is disassembled. For example, the component 2 as shown in figure 2(a) stays stable after the component 3, a bolt, is removed away. Contrarily, if the component 2 as shown in figure 2(b) is joined with the component 1 by the component 3, a bolt, then the component 2 must be disassembled right after the component 3 is disassembled. Otherwise, the component 2 will fall down because it is unstable.

Therefore, existing constraints should be taken into account when generating disassembly sequences. Two symbols ‘>’ and ‘-’ are used for constraints of disassembly. The symbol $e_j > e_k$ means that the component e_j must be disassembled before the component e_k is disassembled. It is necessary to investigate whether the component e_j still exists after the component e_k is disassembled. If the component e_j still exists, then this disassembly sub-node is excluded as shown in figure 3. The symbol within the parentheses stands for the directions that the component e_j can be disassembled, i. e. the directions of $+k$ and $-k$ -axes.

The symbol $e_j - e_k$ means that only the component e_k , and none of the other components, must be disassembled right after the component e_j is disassembled. However, the component e_j does not necessarily need to be disassembled before the component e_k is disassembled. If the component e_k has been disassembled, then this constraint is void. It is necessary to examine the existence of the component e_k after the component e_j is disassembled. If the component e_k exists, it must be disassembled right after the component e_j is disassembled, and the sub-node of disassembling the component e_k must be followed as shown in figure 4. The constraint ‘-’ has the highest priority. The existence of the constraint ‘-’ will affect the number of sub-nodes.

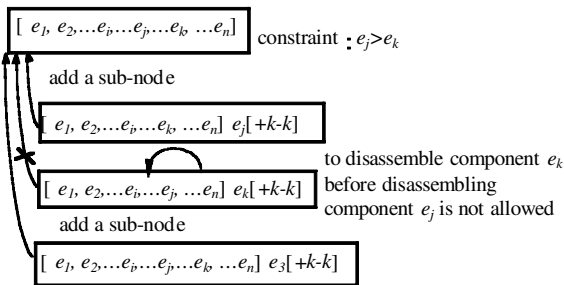


Figure 3. Generation of sub-nodes with constraint.

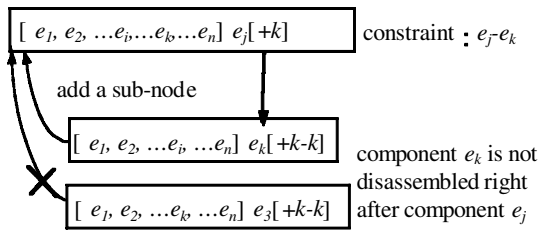


Figure 4. Generation of a sub-node with constraint.

If the component e_j must always be disassembled before the component e_k is disassembled, and the component e_k must uniquely be disassembled right after the component e_j is disassembled, then the symbols $e_j > e_k$ and $e_j - e_k$ are used simultaneously. All of the disassembly sequences can be determined according to their constraints by using the depth-first-search method.

2.4. Disassembly evaluation method

After all of the disassembly sequences are generated, the preferred disassembly sequence must be determined to most efficiently disassemble a product. Several factors affect the efficiency of disassembly. For example, changing the disassembly directions of components during disassembly affects the efficiency of disassembly. If the disassembly direction of a component is the same as that of the component disassembled previously, then the disassembly time can be reduced. The recommended grades for evaluating changes of disassembly directions and tools are shown in tables 1 and 2. The lower the grade in terms of time, second, the better the design for disassembly.

3. Results

A software program is generated based on the algorithm developed in this study by using the Borland C++ Builder 3.0 language. The elements of the disassembly matrix are used as input to the software program. Two assemblies are used to ensure

Direction change (°)	Grade (s)	Example
0	0	No change of direction
90	5	from +x-axis to -y-axis
180	10	from -z-axis to +z-axis

Table 1. Grades for change of disassembly direction.

Tool	Grade (s)	Example
No change	0	Same screwdriver
Change	10	change from screwdriver to pliers

Table 2. Grades for changing tools.

the acceptance of the disassembly matrix, and then the disassembly process of a compressor is determined.

3.1. Disassembly of assembly 1

The interference matrices in the directions of the x , y and z -axes by inspection for the assembly 1 as shown in figure 5, respectively, are

$$H_x = \begin{matrix} & a & b & c & d \\ a & \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \\ b & \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \\ c & \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \\ d & \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix} \end{matrix} \quad (5)$$

$$H_y = \begin{matrix} & a & b & c & d \\ a & \begin{bmatrix} 0 & 1 & 0 & 1 \end{bmatrix} \\ b & \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \\ c & \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \\ d & \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix} \end{matrix} \quad (6)$$

$$H_z = \begin{matrix} & a & b & c & d \\ a & \begin{bmatrix} 0 & 1 & 1 & 1 \end{bmatrix} \\ b & \begin{bmatrix} 1 & 0 & 1 & 1 \end{bmatrix} \\ c & \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \\ d & \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix} \quad (7)$$

The disassembly matrices in the binary system and the decimal system, respectively, are

$$A = \begin{matrix} & a & b & c & d \\ a & \begin{bmatrix} 000 & 110 & 100 & 111 \end{bmatrix} \\ b & \begin{bmatrix} 100 & 000 & 100 & 111 \end{bmatrix} \\ c & \begin{bmatrix} 000 & 000 & 000 & 111 \end{bmatrix} \\ d & \begin{bmatrix} 011 & 011 & 011 & 000 \end{bmatrix} \end{matrix}_{\text{BIN}} \quad (8)$$

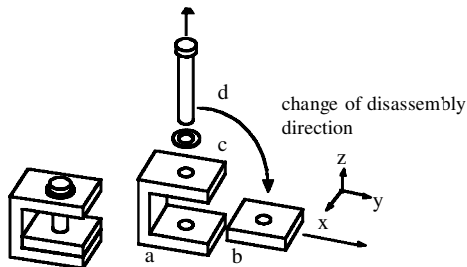


Figure 5. Change of disassembly direction.

and

$$A = \begin{matrix} & a & b & c & d \\ a & \begin{bmatrix} 0 & 6 & 4 & 7 \end{bmatrix} \\ b & \begin{bmatrix} 4 & 0 & 4 & 7 \end{bmatrix} \\ c & \begin{bmatrix} 0 & 0 & 0 & 7 \end{bmatrix} \\ d & \begin{bmatrix} 3 & 3 & 3 & 0 \end{bmatrix} \end{matrix}_{DEC} \quad (9)$$

The combined results of the right, middle and left digits, respectively, for any row and any column of the disassembly matrix in equation (8) by using the Boolean operator are:

$$A = \begin{matrix} & a & b & c & d & result \\ a & \begin{bmatrix} 000 & 110 & 100 & 111 \end{bmatrix} & 111 \\ b & \begin{bmatrix} 100 & 000 & 100 & 111 \end{bmatrix} & 111 \\ c & \begin{bmatrix} 000 & 000 & 000 & 111 \end{bmatrix} & 111 \\ d & \begin{bmatrix} 011 & 011 & 011 & 000 \end{bmatrix} \end{matrix}_{BIN} \quad (10)$$

result 111 111 111 111

Only the component d can be disassembled freely in the direction of the z -axis. Disassembling the component d and eliminating the component d from the corresponding row and column of the disassembly matrix results in

$$A = \begin{matrix} & a & b & c & d \\ a & \begin{bmatrix} 000 & 110 & 100 & \vdots 11 \end{bmatrix} \\ b & \begin{bmatrix} 100 & 000 & 100 & \vdots 11 \end{bmatrix} \\ c & \begin{bmatrix} 000 & 000 & 000 & \vdots 11 \end{bmatrix} \\ d & \begin{bmatrix} \vdots 011 & \vdots 011 & \vdots 011 & \vdots 000 \end{bmatrix} \end{matrix}_{BIN} \Rightarrow \begin{matrix} & a & b & c \\ a & \begin{bmatrix} 000 & 110 & 100 \end{bmatrix} & 110 \\ b & \begin{bmatrix} 100 & 000 & 100 \end{bmatrix} & 100 \\ c & \begin{bmatrix} 000 & 000 & 000 \end{bmatrix} & 000 \end{matrix} \quad (11)$$

100 110 100

The remaining components can be disassembled and the directions of disassembly are $a : \{+x - x - y\}$; $b : \{+x - x + y\}$ and $c : \{+x - x + y - y + z\}$. Any component a , b or c can be selected first to continue disassembly processes. If the component a is disassembled after the component d has been disassembled, then eliminating the corresponding row and column of the component a from the disassembly matrix results in:

$$a \begin{matrix} & a & b & c \\ \begin{bmatrix} \vdots 000 & \vdots 110 & \vdots 100 \end{bmatrix} \\ b & \begin{bmatrix} 100 & 000 & 100 \end{bmatrix} \\ c & \begin{bmatrix} 000 & 000 & 000 \end{bmatrix} \end{matrix}_{BIN} \Rightarrow \begin{matrix} & b & c & result \\ b & \begin{bmatrix} 000 & 100 \end{bmatrix} & 100 \\ c & \begin{bmatrix} 000 & 000 \end{bmatrix} \end{matrix}_{BIN} \quad (12)$$

result 000 100

Using the Boolean operator ‘OR’, the results for the components are also shown in equation (12). The remaining components can be disassembled. The possible disassembly components and the disassembly directions of the components are $b : \{+x - x + y - y - z\}$ and $c : \{+x - x + y - y + z\}$. Similarly, all of the sub-nodes can be created.

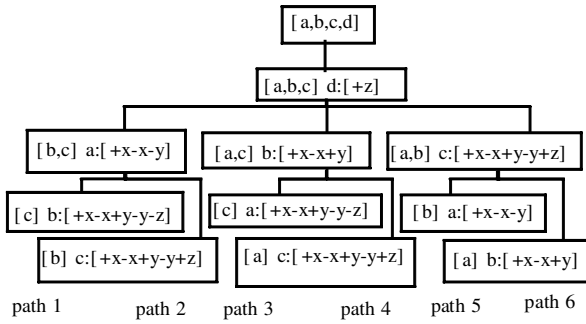


Figure 6. Tree diagram for disassembly sequences of assembly 1.

Inputting the elements of the disassembly matrix in equation (8) into the software program results in the disassembly alternative sequences in the tree diagram as shown in figure 6. The possible disassembly path 1 is from $\{a, b, c, d\}$ to $\{a, b, c\}d : \{+z\}$ to $\{b, c\}a : \{+x - x - y\}$ to $\{c\}b : \{+x - x + y - y - z\}$. There are five possible directions of disassembling the component b , three possible directions of disassembling the component a , and one possible direction of disassembling the component d . Therefore, there are $5 \times 3 \times 1 = 15$ possible disassembly processes for the path 1. Similarly, the possible disassembly processes for the paths 2–6 are $5 \times 3 \times 1 = 15$, $5 \times 3 \times 1 = 15$, $5 \times 3 \times 1 = 15$, $3 \times 5 \times 1 = 15$ and $3 \times 5 \times 1 = 15$, respectively. The total possible disassembly processes is 90 that are obtained from $15 + 15 + 15 + 15 + 15 + 15$.

Considering the direction changes of components during disassembly with the grade in terms of time, the preferred disassembly sequences with disassembly components and directions are $d : \{+z\}a : \{+x\}b : \{+x\}$, $d : \{+z\}a : \{-x\}b : \{-x\}$, $d : \{+z\}a : \{-y\}b : \{-y\}$, $d : \{+z\}a : \{+x\}c : \{+x\}$, $d : \{+z\}a : \{-x\}c : \{-x\}$, $d : \{+z\}a : \{-y\}c : \{-y\}$, $d : \{+z\}b : \{+x\}a : \{+x\}$, $d : \{+z\}b : \{-x\}a : \{-x\}$, $d : \{+z\}b : \{+y\}a : \{+y\}$, $d : \{+z\}b : \{+x\}c : \{+x\}$, $d : \{+z\}b : \{-x\}c : \{-x\}$, $d : \{+z\}b : \{+y\}c : \{+y\}$, $d : \{+z\}c : \{+z\}a : \{+x\}$, $d : \{+z\}c : \{+z\}a : \{-x\}$, $d : \{+z\}c : \{+z\}a : \{-y\}$, $d : \{+z\}c : \{+x\}a : \{+x\}$, $d : \{+z\}c : \{-x\}a : \{-x\}$, $d : \{+z\}c : \{-y\}a : \{-y\}$, $d : \{+z\}c : \{+z\}b : \{+x\}$, $d : \{+z\}c : \{+z\}b : \{-x\}$, $d : \{+z\}c : \{+z\}b : \{+y\}$, $d : \{+z\}c : \{+x\}b : \{+x\}$, $d : \{+z\}c : \{-x\}b : \{-x\}$ and $d : \{+z\}c : \{+y\}b : \{+y\}$.

3.2. Disassembly of assembly 2

The disassembly matrix of the assembly 2 with four components as shown in figure 7 is

$$A = \begin{matrix} & a & b & c & d \\ \begin{matrix} a \\ b \\ c \\ d \end{matrix} & \begin{bmatrix} 000 & 011 & 011 & 011 \\ 111 & 000 & 000 & 000 \\ 111 & 100 & 000 & 000 \\ 111 & 100 & 110 & 000 \end{bmatrix} & \cdot & \end{matrix} \quad (13)$$

Inputting the elements of the disassembly matrix into the software program results in the disassembly sequences in the tree diagram as shown in figure 8. Based on the

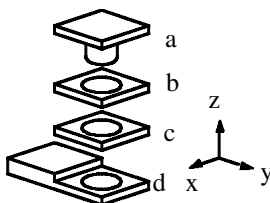


Figure 7. Assembly 2 with four components

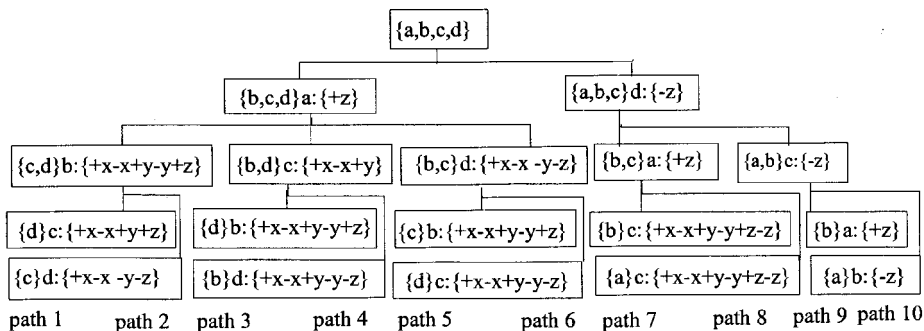


Figure 8. Disassembly tree diagram of assembly 2.

result of the disassembly sequences, it is known that the first step is to disassemble component *a* in the direction of the $+z$ -axis or to disassemble component *d* in the direction of the $-z$ -axis. The numbers of possible disassembly paths are: path 1, $4 \times 5 \times 1 = 20$; path 2, $4 \times 5 \times 1 = 20$; path 3, $5 \times 3 \times 1 = 15$; path 4, $5 \times 3 \times 1 = 15$; path 5, $5 \times 4 \times 1 = 20$; path 6, $5 \times 4 \times 1 = 20$; path 7, $5 \times 1 \times 1 = 5$; path 8, $5 \times 1 \times 1 = 5$; path 9, $3 \times 1 \times 1 = 3$; and path 10, $3 \times 1 \times 1 = 3$. The number of possible total disassembly processes is 126 that are obtained from $20 + 20 + 15 + 15 + 20 + 20 + 5 + 5 + 3 + 3$. The preferred disassembly sequences with corresponding disassembly components and directions are $a : \{+z\} b : \{+z\} c : \{+z\}$ and $d : \{-z\} c : \{-z\} b : \{-z\}$, because neither of them needs to change disassembly direction.

3.3. Disassembly of assembly 3

The assembly 3 is the rotary compressor with sliding vanes as shown in figure 9, which challenges to the currently existing compressors. There are two diagonally symmetric inlet ports and outlet ports, respectively. When the rotor rotates, the vanes slide to change the volume segments and air is drawn in. Air is discharged from the outlet ports after it is compressed. The components of the assembly are shown in figure 10. Components *a*, *b*, *d* and *e* are the front, rear, upper and lower cover plates, respectively. Components *f* and *g* are bearings. There are four bolts on each cover plates. Components *h*, *i*, *j* and *k* stand for four bolts on the plates, *a*, *b*, *d* and *e*, respectively, to reduce the size of disassembly matrices, the computer memory and running time. Components *c*, *l* and *m* are a stator, a rotor and vanes, respectively. There are 13 components totally. If components are constrained in the direction of the *y*-axis to reduce the unnecessary cost of computer running time, the

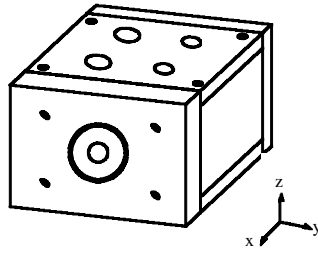


Figure 9. Rotary compressor with sliding vanes.

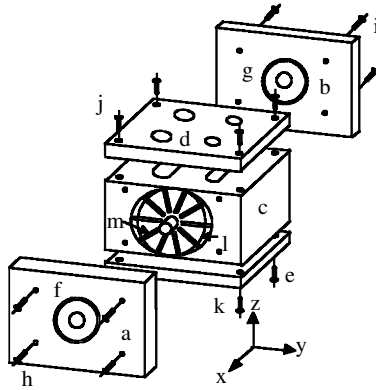


Figure 10. Components of the rotary vane compressor.

simplified interference matrices in the directions of the x -, y - and z -axes, respectively, are

$$H_x = \begin{matrix} & a & b & c & d & e & f & g & h & i & j & k & l & m \\ \begin{matrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ i \\ j \\ k \\ l \\ m \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} & \end{matrix} \tag{14}$$

$$H_y = \begin{matrix} & a & b & c & d & e & f & g & h & i & j & k & l & m \\ \begin{matrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ i \\ j \\ k \\ l \\ m \end{matrix} & \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \end{matrix} \quad (15)$$

First disassembly			Second disassembly		
Sequence	Grade for change		Sequence	Grade for change	
	Direction (s)	Tool (s)		Direction (s)	Tool (s)
$f\{+x\}$			$f\{+x\}$		
$h\{+x\}$	0	10	$h\{+x\}$	0	10
$a\{+x\}$	0	10	$a\{+x\}$	0	10
$m\{+x\}$	0	0	$m\{+x\}$	0	0
$l\{+x\}$	0	0	$l\{+x\}$	0	0
$g\{-x\}$	10	0	$g\{-x\}$	10	0
$i\{-x\}$	0	10	$i\{-x\}$	0	10
$b\{-x\}$	0	10	$b\{-x\}$	0	10
$j\{+z\}$	5	10	$k\{-z\}$	5	10
$d\{+z\}$	0	10	$e\{-z\}$	0	10
$k\{-z\}$	10	10	$j\{+z\}$	10	10
$e\{-z\}$	0	10	$d\{+z\}$	0	10
Sub-total	25	80		25	80
Total	105			105	

Table 3. Comparison of two disassembly processes.

$$H_z = \begin{matrix} & a & b & c & d & e & f & g & h & i & j & k & l & m \\ \begin{matrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ i \\ j \\ k \\ l \\ m \end{matrix} & \left[\begin{array}{cccccccccccc} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array} \right] \end{matrix} . \quad (16)$$

The disassembly matrices in the binary system and the decimal system, respectively, are

$$A = \begin{matrix} & a & b & c & d & e & f & g & h & i & j & k & l & m \\ \begin{matrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ i \\ j \\ k \\ l \\ m \end{matrix} & \left[\begin{array}{cccccccccccc} 000 & 010 & 010 & 010 & 010 & 110 & 010 & 111 & 010 & 010 & 010 & 110 & 010 \\ 011 & 000 & 011 & 011 & 011 & 011 & 110 & 011 & 110 & 011 & 011 & 111 & 011 \\ 011 & 010 & 000 & 110 & 010 & 011 & 010 & 111 & 110 & 111 & 011 & 010 & 010 \\ 011 & 010 & 010 & 000 & 010 & 010 & 010 & 010 & 010 & 111 & 010 & 010 & 010 \\ 011 & 010 & 110 & 110 & 000 & 010 & 010 & 110 & 110 & 110 & 011 & 110 & 110 \\ 110 & 010 & 010 & 010 & 010 & 000 & 010 & 010 & 010 & 010 & 010 & 010 & 010 \\ 010 & 110 & 010 & 010 & 010 & 011 & 000 & 010 & 010 & 010 & 010 & 010 & 011 & 011 \\ 110 & 010 & 110 & 110 & 010 & 010 & 010 & 000 & 010 & 010 & 010 & 010 & 010 & 010 \\ 011 & 111 & 111 & 110 & 010 & 010 & 010 & 011 & 000 & 010 & 010 & 010 & 010 & 010 \\ 011 & 010 & 011 & 011 & 010 & 010 & 010 & 010 & 010 & 010 & 000 & 010 & 010 & 010 \\ 011 & 010 & 111 & 110 & 111 & 010 & 010 & 010 & 010 & 010 & 110 & 000 & 010 & 010 \\ 111 & 110 & 110 & 110 & 010 & 111 & 110 & 010 & 010 & 010 & 010 & 010 & 000 & 110 \\ 011 & 010 & 110 & 110 & 010 & 111 & 010 & 010 & 010 & 010 & 010 & 010 & 110 & 000 \end{array} \right] \end{matrix} \quad (17)$$

and

$$A = \begin{matrix} & a & b & c & d & e & f & g & h & i & j & k & l & m \\ \begin{matrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ i \\ j \\ k \\ l \\ m \end{matrix} & \begin{bmatrix} 0 & 2 & 2 & 2 & 2 & 6 & 2 & 7 & 2 & 2 & 2 & 6 & 2 \\ 3 & 0 & 3 & 3 & 3 & 3 & 6 & 3 & 6 & 3 & 3 & 7 & 3 \\ 3 & 2 & 0 & 6 & 2 & 3 & 2 & 7 & 6 & 7 & 3 & 2 & 2 \\ 3 & 2 & 2 & 0 & 2 & 2 & 2 & 2 & 2 & 7 & 2 & 2 & 2 \\ 3 & 2 & 6 & 6 & 0 & 2 & 2 & 6 & 6 & 6 & 3 & 6 & 6 \\ 6 & 2 & 2 & 2 & 2 & 0 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 6 & 2 & 2 & 2 & 3 & 0 & 2 & 2 & 2 & 2 & 3 & 3 \\ 6 & 2 & 6 & 6 & 2 & 2 & 2 & 0 & 2 & 2 & 2 & 2 & 2 \\ 3 & 7 & 7 & 6 & 2 & 2 & 2 & 3 & 0 & 2 & 2 & 2 & 2 \\ 3 & 2 & 3 & 3 & 2 & 2 & 2 & 2 & 2 & 0 & 2 & 2 & 2 \\ 3 & 2 & 7 & 6 & 7 & 2 & 2 & 2 & 2 & 6 & 0 & 2 & 2 \\ 7 & 6 & 6 & 6 & 2 & 7 & 6 & 2 & 2 & 2 & 2 & 0 & 6 \\ 3 & 2 & 6 & 6 & 2 & 3 & 2 & 2 & 2 & 2 & 2 & 6 & 0 \end{bmatrix} \end{matrix} \quad (18)$$

The constraint conditions are $h > a$, $h - a$, $j > d$, $j - d$, $k > e$, $k - e$, $i > b$, $i - b$, $m > l$, $l > g$, $a > c$, $b > c$, $d > c$, $e > c$, $b > d$ and $b > e$. If the constraints $f > a$ and $g > b$ are added, the total number of the disassembly paths obtained from the disassembly matrix is 492. If the constraints $f > a$ and $g > b$ are further added, the disassembly paths can be reduced into two; $f\{+x\}$, $h\{+x\}$, $a\{+x\}$, $m\{+x\}$, $l\{+x\}$, $g\{-x\}$, $i\{-x\}$, $b\{-x\}$, $j\{+z\}$, $d\{+z\}$, $k\{-z\}$, $e\{-z\}$ and $f\{+x\}$, $h\{+x\}$, $a\{+x\}$, $m\{+x\}$, $l\{+x\}$, $g\{-x\}$, $i\{-x\}$, $b\{-x\}$, $k\{-z\}$, $e\{-z\}$, $j\{+z\}$, $d\{+z\}$. These two processes are reasonable because they are examined and satisfy with the constraints. Based on the disassembly processes, a new compressor is designed, and the number of components is reduced to 9 as shown in figure 11. The disassembly matrix in the decimal system is

$$A = \begin{matrix} & a & b & c & d & e & f & g & h & i \\ \begin{matrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \\ i \end{matrix} & \begin{bmatrix} 0 & 2 & 2 & 6 & 2 & 7 & 2 & 6 & 2 \\ 3 & 0 & 3 & 3 & 6 & 3 & 6 & 7 & 3 \\ 3 & 2 & 0 & 2 & 2 & 7 & 6 & 6 & 6 \\ 6 & 2 & 2 & 0 & 2 & 2 & 2 & 2 & 2 \\ 2 & 6 & 2 & 3 & 0 & 2 & 2 & 3 & 3 \\ 6 & 2 & 6 & 2 & 2 & 0 & 2 & 2 & 2 \\ 3 & 7 & 7 & 2 & 2 & 3 & 0 & 2 & 2 \\ 7 & 6 & 6 & 7 & 6 & 2 & 2 & 0 & 6 \\ 3 & 2 & 6 & 3 & 2 & 2 & 2 & 6 & 0 \end{bmatrix} \end{matrix} \quad (19)$$

The constraints are $f > a$, $f - a$, $g > b$, $g - b$, $d > a$, $e > b$, $a > c$, $b > c$, $h > e$ and $i > h$. The disassembly sequence is $d\{+x\}$, $f\{+x\}$, $a\{+x\}$, $i\{+x\}$, $h\{+x\}$, $e\{-x\}$, $g\{-x\}$, $b\{-x\}$. The grade for this disassembly is shown in table 4. The number of components for the new design is four less than the previous design. The time required for disassembly is 55 s shorter.

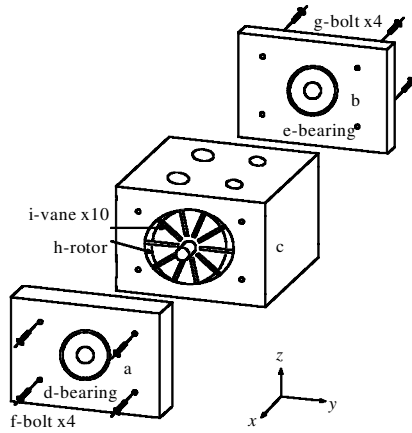


Figure 11. New compressor.

Disassembly sequence	Grade for change	
	Direction (s)	Tool (s)
$d\{+x\}$		
$f\{+x\}$	0	10
$a\{+x\}$	0	10
$i\{+x\}$	0	0
$h\{+x\}$	0	0
$e\{-x\}$	10	0
$g\{-x\}$	0	10
$b\{-x\}$	0	10
Sub-total	10	40
Total	50	

Table 4. Grade for a new compressor.

4. Discussion

If two components are in contact with each other, then interference exists between them, as the component b and c in figure 7. However, interference may exist between two components that are not in contact with each other, as the component b and d in figure 7. Therefore, interference matrices in the directions of the x , y and z -axes are used to obtain the disassembly matrix for determining the disassembly sequences of assemblies. The disassembly matrix not only helps to disassemble disposed devices, but also evaluates potential products for ease of disassembly for disposal during design. The disassembly matrix in the decimal system or the binary system shows the relationships among the components with the complete disassembly information. The computer memory of the disassembly matrix should be less than that of the Dini and Santochi (1992) method using three interference matrices. This study makes it possible to attain all the disassembly sequences and

directions that are not provided by Dini and Santochi and Laperriere and ElMaraphy (1994). Laperriere and ElMaraphy proposed a directed graph expression method, and the directed graph expression of the assembly 2 as shown in figure 7 is shown in figure 12. Based on Laperriere and ElMaraphy's method, six possible paths without considering the disassembly directions of the components are shown in figure 13 for comparison. With consideration of disassembly direction of components for the assembly 2 as shown in figure 7, 126 possible disassembly sequences are obtained from the proposed disassembly matrix. With consideration of grades for changes of disassembly directions of components, the preferred disassembly sequences can be determined to reduce the time required. Using the result of this study will reduce human error to obtain the preferred disassembly processes.

The disassembly matrix can be used to disassemble components one by one. The problems of fixture for disassembly that arises from simultaneously disassembling two components using Laperriere and ElMaraphy's method as shown in figure 2 do not come up in this study.

There are some limitations to applying the results of this study. First, determination of the orientation of the product in the beginning of disassembly is important to maintain the stability of components during disassembly and to reduce the total disassembly time. Second, although the software program generated in this study can provide the reasonable disassembly sequences, users need to input the data of the disassembly matrix in the binary system or the decimal system manually. Further study should associate with the solid modeling of the assembly in the CAD system and the program should be enhanced with the capability to determine interference of components and to improve the efficiency of disassembly. Third, before adopting the solid modeling in the CAD system, users have to provide disassembly constraints and experienced users should provide a complete database for further disassembly

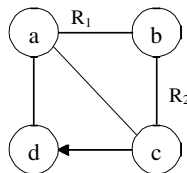


Figure 12. Directed graph expression.

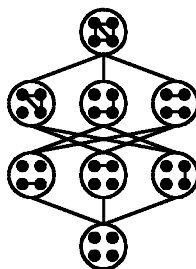


Figure 13. Six possible paths.

study to improve the capability of this software. Fourth, the disassembly matrix is limited for products that can be disassembled in three orthogonal directions.

Different tools may be needed to disassemble different components. Changing tools during disassembling affects the efficiency of disassembly. In this study, the preferred disassembly sequences are not evaluated based on the time of changing and using the tools because to obtain them is tedious and high labour cost. However, an important objective of generating all the possible disassembly sequences is to find the preferred process or the process with the minimum process time. With that goal in mind, attention should be focused on measuring and recording the different types of tools used and the total time to change disassembly directions. Although the CAD system automatically generates all possible disassembly sequences, this system is inefficient because the times for innumerable processes must be inputted and experienced users are required personally to evaluate the system and make adjustments. Therefore, the recommended grades for evaluation of the time required to change tools during disassembly are shown in table 2 for the preliminary study. The longer the time required, the worse the disassembly sequence and the design for disassembly.

The minimum number of disassembly tools should be used to reduce the time for changing tools for product designs. For example, computers are designed in such a way that only a screwdriver is needed for assembly and disassembly to reduce the time needed to change tools. Of course, the best design would need no tools for assembly and disassembly.

5. Conclusion

Although some limitations exist and the software needs to be enhanced in this study, the proposed disassembly matrix needs a smaller computer memory than that of the Dini and Santochi method, and it can also determine all the disassembly sequences and directions of components to improve the existing methods. Therefore, the software program developed in this study can provide a feasible and effective systematic methodology to disassemble disposed products or evaluate the potential products during design. Application of the results from this study will reduce human errors during determining disassembly processes. Importantly, by applying the methodology described in this study, disassembly sequences for complex devices will dramatically improve.

Acknowledgements

The authors sincerely thank the National Science Council of the Republic of China for Grant NSC89-2212-E-002-024 for publishing the study and to Grant D. Huang for comments and revisions made on the manuscript.

Appendix A: Nomenclature

A	disassembly matrix,
a, b, c, d	component of assembly,
H	interference matrix,
h	element of interference matrix,
R	relationship between two components, and
k, x, y, z	direction of axis.

Subscripts

- k, x, y, z direction of axis,
 i row of matrix,
 j column of matrix.

References

- ARAI, E. and IWATA, K., 1993, CAD system with product assembly/disassembly planning. *Robotics and Computer-Integrated Manufacturing*, **10**, 41–48.
- BOOTHROYD, G. and ALTING, L., 1992, Design for assembly and disassembly. *Annals of the CIRP*, **42**, 625–636.
- CHOI, C. K. and ZHA, X. F., 1998, On the automatic generation of product assembly sequences. *International Journal of Production Research*, **36**, 617–633.
- DINI, G. and SANTOCHI, M., 1992, Automated sequencing and subassembly detection in assembly planning. *Annals of the CIRP*, **41**, 1–4.
- GADH, R., 1997, Design for disassembly to support de-manufacturing Progress Report, University of Wisconsin.
- GENC, S., MESSLER, R. W. JR. and GABRIELE, G. A., 1998, Integral attachment using snap-fit feature: a key to assembly automation. *Assembly Automation*, **18**, 68–74.
- GOTTIPOLU, R. B. and CHOSH, K., 1997, Representation and selection of assembly sequences in computer-aided assembly process planning. *International Journal of Production Research*, **35**, 3447–3465.
- GUNGOR, A. and GUPTA, S. M., 1997, Evaluation methodology for disassembly processes. *Computers and Industrial Engineering*, **33**, 329–332.
- HARJULA, T. and RAPOZA, B., 1996, Design for disassembly and the environment. *CIRP Annals — Manufacturing Technology*, **45**, 109–114.
- HOMEM, L. S., MELLO, D. E. and LEE, S., 1991, *Computer-Aided Mechanical Assembly Planning* (Dordrecht Kluwer), pp. 264–266, 349–350.
- JOHNSON, M. R. and WANG, M. H., 1998, Economical evaluation of disassembly operations for recycling, re-manufacturing and reuse. *International Journal of Production Research*, **36**, 3227–3252.
- KO, H. and LEE, K., 1987, Automatic assembling procedure generation from mating conditions. *Computer-Aided Design*, **19**, 5–10.
- LAMBERT, A. J. D., 1997, Optimal disassembly of complex products. *International Journal of Production Research*, **35**, 2509–2523.
- LAPERRIERE, L. and ELMARAGHY, H. A., 1994, Assembly sequences planning for simultaneous engineering applications. *International Journal of Advanced Manufacturing Technology*, **9**, 231–244.
- PAHL, G. and BEITZ, W., 1996, *Engineering Design: a System Approach* (Berlin: Springer).
- TONSHOFF, H. K., MENZEL, E. and PARK, H. S., 1992, A knowledge-based system for automated assembly planning. *Annals of the CIRP*, **41**, 19–24.
- ZUSSMAN, E., KRIWET, A. and SELIGER, G., 1994, Disassembly-oriented assessment methodology. *Annals of the CIRP*, **43**, 9–14.
- ZUSSMAN, E. and ZHOU, M. C., 1999, A methodology for modeling and adaptive planning of disassembly processes. *IEEE Transactions on Robotics and Automation*, **15**, 190–194.