Development a Collide-Based automated assembly sequence algorithm in Industry 4.0

Chi-Hsin Wang
Department of Industrial Engineering and Enterprise Information
Tunghai University, Taichung, Taiwan
Tel: (+886) 4- 2359-0121 ext. 148, Email: stevenwang525@gmail.com

Chen-Yang Cheng †
Department of Industrial Engineering & Management
National Taipei University of Technology (Taipei Tech), Taipei, Taiwan
Tel: (+886) 2- 2771-2171 ext. 2341, Email: cycheng@ntut.edu.tw

Abstract In order achieves industrial 4.0, enterprises move forward as automatic factory and intelligent manufacturing. However, the issue of product's assembly sequence planning by using automation has been concerned. Workload of the product assembly accounted for 20-70 percent of total manufacturing workload. Traditional assembly sequence planning takes a lot of manpower and time. Therefore, the assembly sequence planning plays an important role. Nowadays, there are many systems generate automatically assembly sequence. Even so, there are too many human judgment actions in the process of generating the sequence. This paper developed a Collision-Based Assembly Sequence (CBAS) algorithm that analyzes the relationships between parts of product. This algorithm generates relationship matrix and interference matrix through collide detection and interference detection. At last, automatically generates a set of feasible assembly sequence. By generating feasible assembly sequence automatically, system can reduced the workload of assembly and manufacturing time.

Keywords: assembly sequence planning, automation, collide detection, interference matrix

1. INTRODUCTION

In order achieves industrial 4.0, enterprises move forward as automatic factory and intelligent manufacturing. Therefore, companies began import automation to the product's manufacturing process. Among the entire of product's manufacturing process, cost of product assembly accounted over 40 percent of total cost of product manufacturing process(Shan, Li, Gong, & Lou, 2006). Workload of the product assembly accounted for 20-70 percent of total manufacturing workload. Product's assembly sequence planning (ASP) always plays an important role in the period of product's development, design of production system and production scheduling (Yin, Ding, Li, & Xiong, 2003). Traditional assembly sequence planning takes a lot of manpower and time. Even so, the assembly sequence is not necessarily the best (Zeng, LI, & Zhang, 2014). If the product assembly sequence can generate automatically, not only reduce time and cost of manufacturing, it also satisfy customers' rapidly changing needs immediately.

To generate assembly sequence automatically, this paper have to define two types of matrices. The first matrix is relationship matrix. Relationship matrix represents the contact relationship between component and other components. The second matrix is interference matrix. Most of the literatures which are related to the generating assembly sequence have mentioned the interference matrix which can be used to decide the precedence feasibility of the given assembly sequence (Lv & Lu, 2010). The relationship matrix and interference matrix are indispensable when generating assembly sequence. Most of literatures only determine the meaning of value from the matrix and the way to revise assembly sequences. These literatures have less description about the way to generate relationship matrix and interference matrix. Therefore, this paper develops a Collision-Based Assembly Sequence (CBAS) algorithm that can generate automatically relationship matrix and interference matrix from CAD file. The CAD file is the only input information for CBAS. This algorithm will generate relationship matrix and interference automatically. Through these matrices, the algorithm can produce a feasible assembly sequence.

The rest of this paper organized as follows: Section 2 reviews the literatures of generating assembly sequence by using interference matrix and the way of generating interference matrix. Section 3 shows the CBAS algorithm. Section 4 is CBAS algorithm verification. Section 5 presents the result of this research and suggests further direction of the future study.

2. LITERATURE REVIEW

2.1 Assembly sequence generated automatically

The existing literature can be classified into two directions. The first direction calculates the assembly sequence through algorithm. Chung-Ching and Hwai-En (2010) propose the CBCSA to build an association network and modular product. Enhancing the effect of assembly sequence optimization and simplify the complex assembly interference problem. Lv and Lu (2010) propose DPSO to solve ASP problems. Zeng et al. (2014) advance firefly algorithm to compute assembly sequence and provide fitness function to improve traditional ASP. This literature proposes an algorithm which can compute more reasonable number of assembly tools change. Hu, Xiang, Peng, and Shuanggao (2012) propose an ASP by using weighted directed graph. This literature verifies that the ASP has certain feasibility and effectiveness through the aircraft parts. Lee and Yu (2004) propose the disassembly sequencing in the recycling of retired products. This literature analysis and generate a disassembly sequence through relationship matrix and components classification. This paper will use this literature's method to generate assembly sequence. The second direction through CAD software to crawl information and computing assembly sequence. Dong-Ru (2008) develop a system that extract CAD file's information through CAD software's API and automatically generate disassembly sequence. This literature uses Object Relationship Graph (ORG) to show the relationship between the structures of the product. Pan (2005) develops an automatic ASP that can extract geometric information from STEP file and list all possible assembly sequence. Pintzos, Triantafyllou, Papakostas, Mourtzis, and Chryssolouris (2016) propose the concept of the part based on the geometric characteristic of the assembly layer classification and generate assembly sequence through the CAD software. Most of literatures have mentioned the relationship matrix and interference matrix. Next, we will introduce these matrices' definition and the way to revise assembly sequence.

2.2 Interference matrix

Interference analysis calculates the directions in which a component may be assembled without causing any interference(Ou & Xu, 2013). Interference matrix describes the collision interference relationships between the moving component and the other components on the way from the original assembly position of the moving component to the infinite distance along the axes of Cartesian coordinates (±x, ±y, ±z) (Yu & Wang, 2013). The interference matrix was first proposed by Dini and Santochi (1992) in assembly sequence planning. This paper's interference matrix verified that the interference matrix $\mathrm{A_k}$ is a square matrix of a product formed by *n* components $\mathrm{e_s}$ (with *s*=0 to *n*). The A_k defined as Equation (1):

$$A_k = \begin{array}{c} \\ e_1 \\ e_2 \\ \vdots \\ e_\mathrm{n} \end{array} \begin{array}{c} \begin{array}{cccc} e_1 & e_2 & \dots & e_\mathrm{n} \end{array} \\ \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,\mathrm{n}} \\ a_{2,1} & a_{2,2} & & a_{1,\mathrm{n}} \\ \vdots & \vdots & \ddots & \vdots \\ a_{\mathrm{n},1} & a_{\mathrm{n},2} & \cdots & a_{\mathrm{n},\mathrm{n}} \end{bmatrix} \end{array} \quad (1)$$

Where $a_{s,t} = 1$ if the component e_s interferences with the component e_t during the translation along the direction +k. If A_k is the interference matrix evaluated along the direction +k, the transpose matrix ${A_k}^T$ represents the interferences along the opposite direction –k, otherwise $a_{s,t} = 0$. Because the component does not interferences with itself, so $a_{s,s}$ is always equals to zero.

2.3 Revised assembly sequence

Dini and Santochi (1992) develops an assembly sequence planner that can generate automatically all possible sequences. This literature's method to generate assembly sequence is to check each component which could disassembly from product. When the disassembly direction is chosen, the system checks the interference matrix which the component can disassemble from this direction. After finished this process, it choose next direction and do the process again until complete disassembly of the product. This method checks each component in the interference matrix in the process of generating assembly sequence.

Some literatures use collision detection or analysis to revise the assembly sequence (Pintzos et al., 2016). The collision analysis of a CAD assembly model is a process permitting to identify the paths in which a component can be assembled or disassembled without interferences (Hadj, Trigui, & Aifaoui, 2015). Yi and Jianhua (2013) mention the firefly algorithm and use interference matrix to revise assembly sequence more efficiency.

2.4 The method of matrices generated

In order to generate automatically assembly sequence, the interference matrix also has to produce automatically. Dini and Santochi (1992) generate interference matrix and contact matrix are built in function of the data deriving from the CAD simulation of the component translations. Yu and Wang (2013) develop an AutoAssem system that can extracts the information of parts first and executes interference detection in global coordinate system directions. Kang and Peng (2014) and Hadj et al. (2015) use collision detection generate interference. Some literatures have to input manually (Yi & Jianhua, 2013) . This paper develops CBAS algorithm to generate a feasible assembly sequence through the relationship matrix and interference matrix which generate automatically.

3. COLLISION-DETECTION-BASED ASSEMBLY SEQUENCE ALGORITHM

CBAS algorithm analyzes CAD file to be a relationship matrix, and generates a set of feasible assembly sequence. CBAS's calculate process shown in Figure 1 and Figure 2. This section defines notation that CBAS used and described the step of algorithm.

3.1. Notations Definition

n: the total number of components.
F: the position that fix component in the array.
CM_n: the matrix that represent *n* components name.
RM_{st}: the matrix that represent the relationship between all the components.
IM_{kst}: the matrix that represent the collision relationship between all the components in each direction.

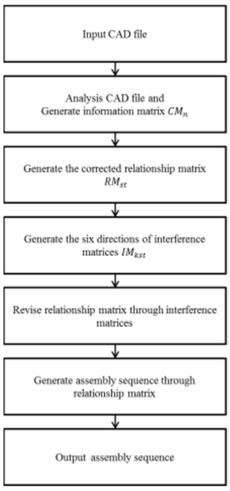

Figure 1. CBAS Diagram

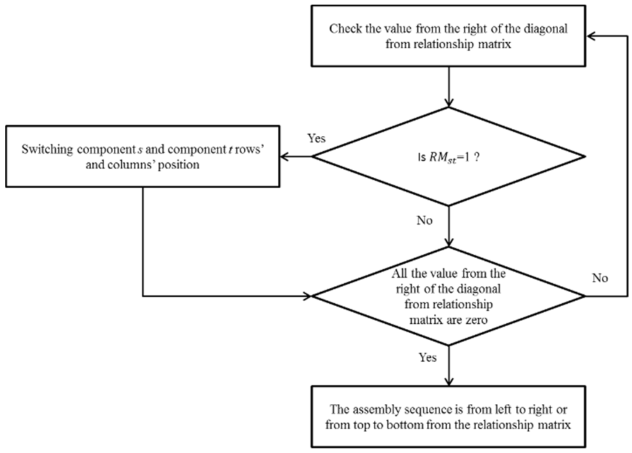

Figure 2. The flow chart of generate assembly sequence

3.1. The step of CBAS

Step 1. Analysis of CAD file and generate initial relationship matrix

To extract components' information from CAD file includes component's name, identification number, quantity, origin coordinate and fixed component or not. The fix component means that the component cannot move while analysis assembly sequence. Collect and organize all information to a components' information table and save serial number into CM_n. After calculating the total number of components, algorithm generate initial relationship matrix RM_{st} and ranks of the components in accordance with the same sequence, filled from top to bottom and from left to right. In the relationship matrix, there are two numbers represent relationship between components:

$$RM_{st} = \begin{cases} 0, \text{component } s \text{ and component } t \text{ do not have relationship;} \\ 1, \text{component } s \text{ and component } t \text{ have contact or interference relationship.} \end{cases}$$

If the assembly sequence is element from top to bottom in the relationship matrix, it represents a feasible assembly sequence. The relationship matrix has to maintain satisfy one condition. This condition is the right of the diagonal from relationship matrix has to not exist "1". Then, CBAS have to set all value from relationship matrix to "0". The function described as below Equation (2)

$$\sum_{s=0}^{n} \sum_{t=0}^{n} RM_{st} = 0 \qquad (2)$$

Take Figure 3 for example, the components from this example correspond to the initial relationship matrix. This paper set the component *A* is fix component.

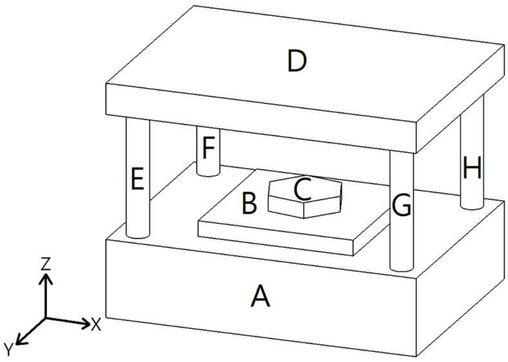

Figure 3. The example of product

Step 2. Generate the uncorrected relationship matrix

Interference detection checks the interference between components and finds out interference situations like surface contact, point contact or overlapping. The interference detection outputs all components what component has interference with another component. And then, the component defined as related with another component. The value of relationship matrix revises to “1”. The function described as below Equation (3)

$$\sum_{i=0}^{\mathrm{In\_n}} RM_{xy} = 1 \qquad \sum_{i=0}^{\mathrm{In\_n}} RM_{xy} = 1 \tag{3}$$

x: the first component’s position that found by interference detection.
y: the second component’s position that found by interference detection.

Take Figure 3 for example, when the component *A* and the component *B* have interference, RM_{21}=1 RM_{12}=1. After CBAS finished all revising, the relationship matrix can represent the contact relationship between all components like Equation (4).

$$RM_{st} = \begin{array}{c} \\ E \\ B \\ A \\ C \\ F \\ D \\ G \\ H \end{array} \begin{array}{c} \begin{array}{cccccccc} E & B & A & C & F & D & G & H \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \end{array} \tag{4}$$

Step 3. Generate six direction of initial interference matrices

Interference matrix represents the interference between all components. There are six direction include *+X*, *-X*, *+Y*, *-Y*, *+Z* and *-Z*. The function described as Equation (5).

$$\sum_{k=0}^{5}\sum_{s=0}^{n}\sum_{t=0}^{n} IM_{kst} = 0 \tag{5}$$

Function (5) output six directions’ interference matrix. Where the $IM_{kst} = 1$, if the component *s* interferences with the component *t* during the translation along the direction *k*.

Step 4. Revise interference matrix through each component’s interference detection

Because interference matrixes represent the component has interference with other components or not. CBAS start to do collision detection. Collision detection checks the component which the component is moving. First, we have to choose one direction and move each component. The fix component does not do collision detection. If other components do not collide any component, the interference matrixes do not do any revise. For example, the component *D* move with *+Z* direction and does not collide any components. The component *D* does not do any revise. When the component couldn’t move, the interference detection will output the component which interference with the moving component and revise the value to “1”. For example, when the component *B* move with *+Z* direction and the component *C* is blocked. So IM_{413}=1. Interference matrixes revise the value to “2” which represents this component will collide other components when it was moving. For example, when the component *C* moves with *+Z* direction and collide the component *D*, the value revise to “2”. And so on, six interference matrixes revised finished and the result shown in Equation (6).

$$
IM_{0st}= \begin{array}{c|cccccccc} +X & E & B & A & C & F & D & G & H \\ \hline E & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ B & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ F & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ D & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ G & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ H & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{array} \quad IM_{1st}= \begin{array}{c|cccccccc} -X & E & B & A & C & F & D & G & H \\ \hline E & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ B & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ F & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ D & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ G & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ H & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{array}
$$

$$
IM_{2st}= \begin{array}{c|cccccccc} +Y & E & B & A & C & F & D & G & H \\ \hline E & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ B & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ F & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ D & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ G & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ H & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{array} \quad IM_{3st}= \begin{array}{c|cccccccc} -Y & E & B & A & C & F & D & G & H \\ \hline E & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ B & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ F & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ D & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ G & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ H & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{array} \qquad (6)
$$

$$
IM_{4st}= \begin{array}{c|cccccccc} +Z & E & B & A & C & F & D & G & H \\ \hline E & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ B & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ F & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ D & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ G & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ H & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array} \quad IM_{5st}= \begin{array}{c|cccccccc} -Z & E & B & A & C & F & D & G & H \\ \hline E & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ B & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ A & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ F & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ D & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ G & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ H & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}
$$

Step 5. Revise relationship matrix through interference matrix

This step revises the relationship matrix through interference matrix. Because we regard fix component as the first assembly part, the fix component's straight value change to "1" and row values change to "0". The function describes as Equation (7) and Equation (8)

$$\sum_{s=0}^{n} RM_{sF} = 1 \qquad (7)$$

$$\sum_{s=0}^{n} RM_{Fs} = 0 \qquad (8)$$

Next, check that there has any component that does not collide other components when it was moving. For example, only the component *D* moves with +*Z* direction does not have any interference with other components when it was moving. We regard the component *D* as the last assembly component. In the relationship matrix, the component *D* of the row value revises to "0". Because the component will not related itself, the value from relationship matrix with the same component revises to "0". The function described as Equation (9).

$$\sum_{s=0}^{n} RM_{ss} = 0 \qquad (9)$$

Search two components that both side value of the diagonal is "1". For example, in the (4), the components that both side value of the diagonal is "1" are component *AB*, *AE*, *BC*, *DE*. Because the component *A* has been already regarded as the first assembly component, the relationship matrix does not do any revise. The remaining components are *BC* and *DE*. Then, check these components which has related with fix component and revise the relationship matrix. Take the components *BC* and *DE* for example, because RM_{12}=1 and RM_{02}=1, the component *B* and the component *D* take priority over the component *C* and the component *E*. RM_{13}=0, RM_{31}=1, RM_{05}= RM_{45}= RM_{65}= RM_{75}=0. The relationship matrix has finished secondary revise and describe as Equation (10)

$$RM_{st} = \begin{array}{c} \\ E \\ B \\ A \\ C \\ F \\ D \\ G \\ H \end{array} \begin{array}{c} \begin{array}{cccccccc} E & B & A & C & F & D & G & H \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \quad (10)$$

Step 6. Revise relationship matrix and generate assembly sequence

This step generates the assembly from the relationship matrix as Equation (10). Because of the right of the diagonal from relationship matrix has to not exist “1”, it has to do the last revise. In order to get a feasible assembly sequence, the value “1” from the right of the diagonal has move to the left. The method is to switch rows’ and columns’ position. The result of revise described as Equation (11). The result of assembly sequence is{ A, B, C, E, F, G, H, D}.

$$RM_{st} = \begin{array}{c} \\ A \\ B \\ C \\ E \\ F \\ G \\ H \\ D \end{array} \begin{array}{c} \begin{array}{cccccccc} A & B & C & E & F & G & H & D \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array} \quad (11)$$

4. SYSTEM VERIFICATION

The step1 to step4 of the CBAS algorithm has been verified on the system that based on SolidWorks 2015, which use build-in VBA program to do collide and interference detection. The main function this system is to crawl the assembly’s information and generate relationship matrix. The step 5 of CBAS algorithm has been verified on the system based on LabVIEW 2013, which generate assembly sequence through analysis relationship matrix. No matter which system generates the relationship matrix, this system will generate a feasible assembly sequence through analysis.

First, input the CAD file and use embedded VBA modules to extract information. Analysis each component and output relationship matrix to TXT files through the built-in collision detection and interference detection. The operation screen shown in Figure 4. Input the TXT file to LabVIEW and generate a feasible assembly sequence. The operation screen shown as Figure 5. The assembly sequence of system output shown in Figure 6.

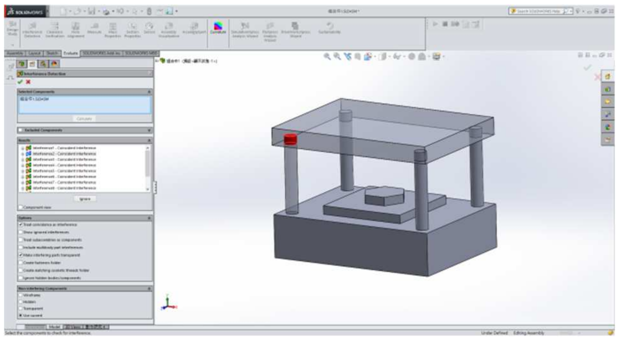

Figure 4. The interference detection of SolidWorks

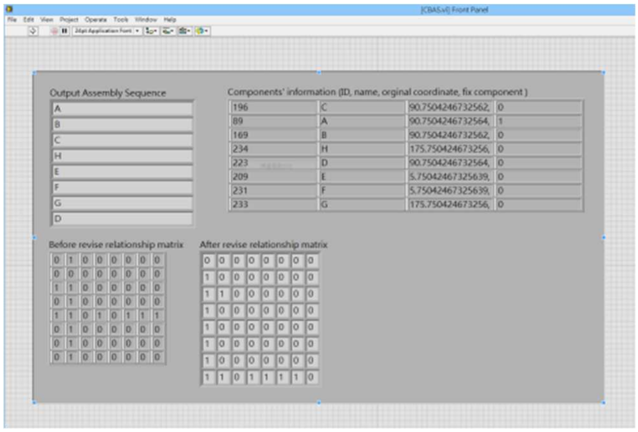

Figure 5. The CBAS result of LabVIEW

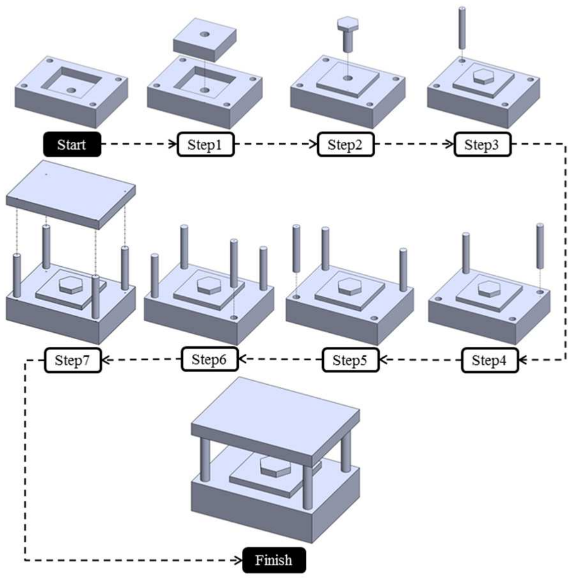

Figure 6. The assembly sequence of CBAS output

5. CONCLUSION

This paper develops CBAS that can generate automatically through two types of matrices. The relationship matrix represents contact relationship between all components. The interference matrix represents the interference relationship which generate through the collision detection and the interference detection. Then, CBAS revises the relationship matrix so that the sequence from left to right or from top to button is one of a feasible assembly sequence. This research develops an algorithm that contributes to calculate feasible assembly sequence immediately on the stage of product development. In the future development, the first is adding the factors that probability affects the assembly sequence (e.g., the contact type, assembly tools, assembly direction or the number of direction change). Let the assembly sequence more close to the optimization. The second let the robotic automatically assembly according to the assembly sequence. In the future, CBAS can formulate a control code table. The control table contains the control code like moving, crawl or change assembly tools. As long as input the robotics' control code table, CBAS can generate the automatically assembly path code automatically. Reach the process automatic from the product's CAD file to the goods.

REFERENCE

Chung-Ching, Yang, & Hwai-En, Tseng. (2010). The modular product design and assembly sequence algorithm study.

Dini, Gino, & Santochi, Marco. (1992). Automated sequencing and subassembly detection in assembly planning. *CIRP Annals-Manufacturing Technology, 41*(1), 1-4.

Dong-Ru, Li. (2008). Application of the API of CAD System to the Establishment of Product Structure Data Model and the Automatic Analysis of Assembly and Disassembly

Hadj, Riadh Ben, Trigui, Moez, & Aifaoui, Nizar. (2015). Toward an integrated CAD assembly sequence planning solution. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, *229*(16), 2987-3001.

Hu, Lu, Xiang, Huang, Peng, Du, & Shuanggao, Li. (2012). Aircraft Assembly Sequence Planning Based on Weighted Directed Graph. *Journal of Nanjing University of Aeronautics & Astronautics*, S1.

Kang, Xiumei, & Peng, Qingjin. (2014). Integration of CAD models with product assembly planning in a Web-based 3D visualized environment. *International Journal on Interactive Design and Manufacturing (IJIDeM), 8*(2), 121-131.

Lee, In-Min, & Yu, Jyh-Cheng. (2004). Disassembly Sequencing in the Recycling of Retired Products. *The 21th National Conference on Mechanical Engineering of CSME*.

Li, Mingyu, Wu, Bo, Hu, Youmin, Jin, Chao, & Shi, Tielin. (2013). A hybrid assembly sequence planning approach based on discrete particle swarm optimization and evolutionary direction operation. *The International Journal of Advanced Manufacturing Technology, 68*(1-4), 617-630.

Lv, HongGuang, & Lu, Cong. (2010). An assembly sequence planning approach with a discrete particle swarm optimization algorithm. *The International Journal of Advanced Manufacturing Technology, 50*(5-8), 761-770.

Ou, Li-Ming, & Xu, Xun. (2013). Relationship matrix based automatic assembly sequence generation from a CAD model. *Computer-Aided Design, 45*(7), 1053-1067.

Pan, Chunxia. (2005). Integrating CAD files and automatic assembly sequence planning.

Pintzos, George, Triantafyllou, Christos, Papakostas, Nikolaos, Mourtzis, Dimitris, & Chryssolouris, George. (2016). Assembly precedence diagram generation through assembly tiers determination. *International Journal of Computer Integrated Manufacturing*, 1-13.

Shan, Hongbo, Li, Shuxia, Gong, Degang, & Lou, Peng. (2006). *Genetic simulated annealing algorithm-based assembly sequence planning.* Paper presented at the Technology and Innovation Conference, 2006. ITIC 2006. International.

Yi, ZBLMZ, & Jianhua, M. (2013). Research on assembly sequence planning based on firefly algorithm. *J. Mech. Eng, 11*, 025.

Yin, ZhouPing, Ding, Han, Li, HanXiong, & Xiong, YouLun. (2003). A connector-based hierarchical approach to assembly sequence planning for mechanical assemblies. *Computer-Aided Design, 35*(1), 37-56.

Yu, Jiapeng, & Wang, Chengen. (2013). A max–min ant colony system for assembly sequence planning. *The International Journal of Advanced Manufacturing Technology, 67*(9-12), 2819-2835.

Zeng, Bing, LI, Ming-fu, & Zhang, Yi. (2014). Assembly sequence planning based on improved firefly algorithm. *Computer Integrated Manufacturing Systems*, *4*, 011.