International Journal of Technology 10(2): 339-350
ISSN 2086-9614

AUTOMATIC PRECEDENCE CONSTRAINT GENERATION FOR ASSEMBLY SEQUENCE PLANNING USING A THREE-DIMENSIONAL SOLID MODEL

Alfadhlani[1*], T.M.A. Ari Samadhi[2], Anas Ma'ruf [2], Isa Setiasyah Toha[2]

[1]*Industrial Engineering Department, Andalas University, Limau Manis, Padang, 25163, Indonesia*
[2]*Industrial Engineering Department, Bandung Institute of Technology, Jl. Ganesha 10, Bandung, 40132, Indonesia*

(Received: March 2017 / Revised: April 2018 / Accepted: February 2019)

ABSTRACT

The assembly sequence planning of a product can be generated through three phases: first, generating precedence constraints; second, searching for assembly sequence alternatives; and third, selecting the best assembly sequence. Assembly sequence generation needs precedence constraints in order to find a feasible assembly. A collision between two components can cause the blocking of one by the other after assembly. This research proposes an automated method for generating precedence constraints. The method employs certain information: the collision-free assembly path; the number of connections between components; and component volume. This information is extracted from the CAD (Computer Aided Design) database. The methods resulting from the research will be used to develop an automated process of assembly sequence generation using a three-dimensional (3D) solid drawing in the form of a stacked drawing in a CAD system.

Keywords: Assembly sequence; CAD; Collision detection; Precedence constraint

1. INTRODUCTION

Assembly is the process of joining parts together to form a completed product, and needs to be evaluated as early as possible in the product design stage so that it will not be difficult to install a component because of tolerance error, inappropriate dimensions or geometry errors. A product designer needs to improve the design if assembly difficulties occur, an additional task which will increase the production cost. It is essential to plan assembly because there are feasible assembly sequence alternatives which can be selected, based on dimensions and geometry. An appropriate assembly sequence will reduce operational difficulties, tool quantity and work hours, so consequently it will also reduce production costs (Lai & Huang, 2004).

Designers can evaluate the assembly process at the early stages of the design using CAD. Previous researchers have proposed assembly sequence generation methods based on CAD systems, such as Delchambre (1992), Ariastuti et al. (1998), Tseng and Li (1999), Toha et al. (2004) and Alfadhlani and Toha (2005; 2008). Some of these studies have proposed automatic methods to generate the assembly sequence.

Possible assembly sequences for complex product assembly planning is determined by consider precedence relation information (Lai & Huang, 2004). Precedence relation is defined as precedence constraint, it contains information about the list of components that must be assembled beforehand (predecessors) and the choice of components to be joined together later

*Corresponding author's email: alfadhlani@eng.unand.ac.id, Tel. +62-0751-72497, Fax. +62-0751-72566
Permalink/DOI: https://doi.org/10.14716/ ijtech.v10i2.3064

(successors). In assembly planning, the precedence relation needs to be determined to ensure that the planned assembly operation can be applied. The geometric model of product assembly can be used to generate precedence constraints.

Delchambre (1992) distinguished two types of precedence constraint: hard constraints and soft constraints. Hard constraints arise because of the geometrical condition of components and their position in the final assembly, while soft constraints comprise stacking and technological constraints. Stacking constraints arise if external fasteners (such as screws) hold together a stack of other components, so it is best to impose a given assembly sequence for this group of components. On the other hand, technological constraints are specified by the operator, and arise because of the use of specific tools. It is recommended that soft constraints are considered in assembly planning. If the generated assembly sequences are feasible without considering the soft constraints, then they can be ignored.

Ariastuti et al. (1998) and Toha et al. (2004) used "face and joint" information as precedence constraints which are determined from the assembly line in the CAD system, while Li et al. (2010) identified such constraints by using a connector knowledge-based approach, employing standard connectors such as threaded fasteners or keys. Morato et al. (2013) generated precedence constraints based on component motion planning and component interaction clusters, which can mutually affect each others' accessibility when assembled. All three methods above used the disassembly approach, which requires complex geometry analysis.

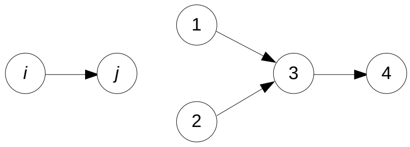
A feasible assembly must be free of collisions between components, which can occur when one component is blocked by another in the assembly. The collision-free assembly path (CFAP) information in this paper was established by using a CFAP algorithm proposed by Alfadhlani et al. (2011). The paper discusses the development of an automated method for generating precedence constraints using the disassembly approach, and considering the CFAP information, the number of component connections, and component volume. All these data were extracted from the CAD database using the component database formation algorithm proposed by Alfadhlani et al. (2011). SolidWorks 2005 was used as the CAD system, and a stacked drawing in a 3D solid model was used as input.

The remainder of the paper is divided into the following sections. Section 2 elaborates the rules of precedence diagramming, while Section 3 explains the process of generating precedence constraints based on a collision-free assembly path. Section 4 explains the priority rules for selecting the component to be released, Section 5 discusses the use of associative law as a rule for improving precedence constraint, and Section 6 describes the algorithm of precedence constraint generation. Section 7 provides an example of the implementation of the proposed methods for generating precedence constraints, and the conclusions are presented in Section 8.

2. METHODS

2.1. Precedence Diagram

A precedence constraint can be represented in the form of a network or arrow diagram. As shown in Figure 1a, a node represents a component to be assembled, with an arrow representing an assembly activity that connects node-*i* and node-*j*.



(a) (b)

Figure 1 Precedence diagram

The arrow length is not proportional to the duration of the activity. Activities (3, 4) cannot be done before activities (1, 3) and (2, 3) have been completed; see Figure 1b.

The precedence diagram is formed by taking into account the following considerations:
1) Each assembly activity is represented by one, and only one, arrow in the diagram.
2) Two or more activities are not allowed to connect the same two nodes.
3) The correct precedence relations in the diagram are ascertained by answering certain questions:
   a) Which activities must be completed before the observed activity can be done?
   b) Which activities must follow the observed activities?
   c) Which activities can be done in parallel with the observed activity?

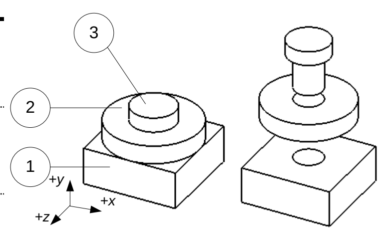
2.2. Precedence Constraint Generation

Precedence constraint is used to guarantee the feasibility of the generated assembly sequence, and is ascertained by considering the CFAP. CFAP information was obtained by using the automatic method proposed by Alfadhlani et al. (2011). The precedence constraint generation in this paper was performed following three steps: (1) determine the components to be released based on the CFAP; (2) conduct the disassembly test; and (3) re-evaluate the CFAP value without involving the components that have been released.

The components to be released are determined based on the CFAP value,which equals 1 in at least one direction of the coordinate system. The sign in the direction of the CFAP must be changed if using the disassembly approach. With reference to Table 1, the evaluation of the CFAP value with a logical AND results in the value being equal to 1, as found in component-1 in direction +*y* and component-3 in direction –*y*. If components 1 and 3 are released, then their direction must be changed to the oppositeway. Component-1 can be released to direction –*y*, and component-3 to direction +*y*.

Table 1 Collision-free assembly path (Alfadhlani et al., 2011)

Component	Contact	+*x*	-*x*	+*y*	-*y*	+*z*	-*z*
	1,2	1	1	1	0	1	1
1	1,3	0	0	1	0	0	0
	AND	0	0	1	0	0	0
	2,1	1	1	0	1	1	1
2	2,3	0	0	1	0	0	0
	AND	0	0	0	0	0	0
	3,1	0	0	0	1	0	0
3	3,2	0	0	0	1	0	0
	AND	0	0	0	1	0	0



If component-3 is selected to be released, the results of the re-evaluation of its CFAP are shown in Table 2. Component-1 can be released to all directions except direction +*y*, whereas component-2 can be released to all directions except direction –*y*. It can be stated that component-3 is the disassembly predecessor of components-1 and 2. Therefore, if component-*i* can be released after component-*j*, then component-*j* is the predecessor of component-*i*.

Table 2 Re-evaluation of the collision-free assembly path

Component	Contact	+*x*	-*x*	+*y*	-*y*	+*z*	-*z*
1	1,2	1	1	1	0	1	1
	AND	1	1	1	0	1	1
2	2,1	1	1	0	1	1	1
	AND	1	1	0	1	1	1

2.3. Component Selection for Disassembly

The disassembly approach is used to generate the precedence constraints; the CFAP must be re-evaluated eachtime a component released. If there is more than one candidate component that can be released, priority rules for selecting them are needed.

The feasibility of assembly sequences is guaranteed by using the CFAP information, while stability is secured using the priority rules. Assembly operations need stability because it is probable that an operator will often make a reorientation of the product during assembly. Alfadhlani and Toha (2005) generated assembly sequences using the criteria of the number of matings between components that have and have not been installed, and component volume. Mating is contact between the components of a product, defined with reference to the direction of the normal vector and the shape of the contacted face.

The priority criterion based on mating is related to the connection between components. This connection is defined as the relationship between components in the final assembly, observed by the presence of mating. The connection may have more than one mating. In this paper, the criterion based on the number of matings is changed to the number of connections, while the volume criterion follows that proposed by Alfadhlani and Toha (2005). The research uses the disassembly approach; however, Alfadhlani and Toha used the assembly approach, so component selection priority is not based on the largest volume, but the smallest.

The following criteria for selecting a component are proposed if there are various disassembly component candidates:

1) Select the component that has the fewest connections with other components that have yet to be selected.
2) Select the candidate component that has the smallest volume and meets criterion 1.
3) Select a component arbitrarily from the components that meet priority criteria 1 and 2 if there is more than one candidate.

The first selection criterion considers the close relationship between the next component to be assembled and those that have already been assembled, as proposed by Alfadhlani and Toha (2005). The closeness of the relationship is related to the connection and stability between the components in the final assembly. Because the disassembly approach is used in this research, then the next component to be released is based on its having the fewest connections with the components yet to be released. Criterion 2 is used if the first is unable to find a unique candidate.

De Fazio and Whitney (1987) formed sub-assembly and assembly sequences using the criteria of the maximum number of actions completed perassembly operation, and the maximum number of liaisons completed perassembly operation. On the other hand, Ariastuti et al. (1998) used the number of exploded assembly line criterion. All these criteria are the same as that of the number of connections, as proposed in this research. The liaison is a representation of the connection between the components, while the exploded assembly line exists because of the connections between the components. Ariastuti et al. (1998) prioritized that the component to be assembled first was that which had the highest number of the exploded assembly lines. This is similar to the idea of making a choice based on the highest number of connections.

As shown in Figure 2, component-1 and component-3 are candidates for disassembly and have the same number of connections. Using the above priority criteria, component-3 is selected as that to be disassembled first because it has a smaller volume than component-1.

The proposed selection criteria are formulated following Set A of rules and are used in the algorithm of precedence constraint generation. The priority rules for selecting candidates are as follows:

A: Priority rules for selecting candidate components.

1) If there is more than one candidate that can be released, then choose a component that has the fewest connections with components that have not yet been selected.
2) If rule 1 results in more than one candidate, then select a candidate component that has the smallest volume.
3) If rules 1 and 2 result in more than one candidate, then select a component arbitrarily that meets rules 1 and 2.

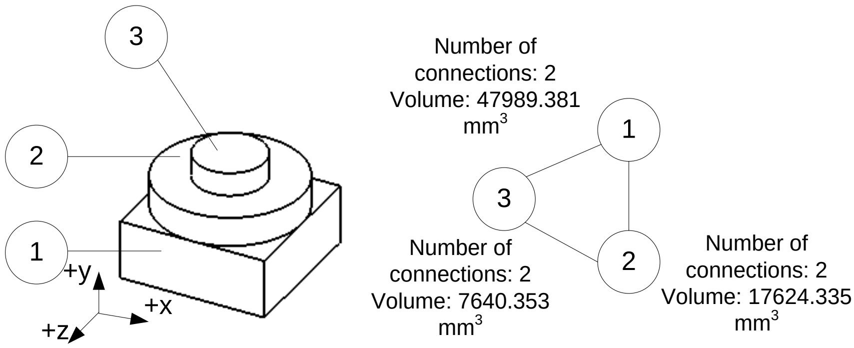


Figure 2 Number of connections and volume of components

2.4. Precedence Constraint Improvement

If components 1, 2, 3 and 4 have the following precedence relations:

a) 4 precedes 3
b) 3, 4 precedes 2
c) 3 precedes 1

then points *a* to *c* above are determined based on the particular direction for the six-axis observed. The following additional criteria are needed to obtain a complete precedence constraint:

1) For component-*i*, which is a predecessor of component-*j*, then the predecessors of component-*i* are also predecessors of component-*j*. For the above example, component-3 is a predecessor of component-1, therefore component-4, which is a predecessor of component-3, is also a predecessor of component-1. Consequently, the precedence relation of component-1 is refined into 3, 4 precede 1.
2) For all the predecessors of component-*i*, which are the predecessors of component-*j*, while component-*i* is connected with component-*j*, then component-*i* is also the predecessor of component-*j*. For the above example, all the predecessors of component-2 are predecessors of component-1, as established in step 1. Component-2 is connected to component-1, so it is also a predecessor of component-1. The precedence relation of component-1 can be refined into 2, 3, 4 precede 1.

The criteria for improving the precedence relation are formulated in Set B of rules using associative law, as follows:

B: Precedence constraint improvement.

1) If component-*i* is a predecessor of component-*j*, then all the predecessors of component-*i* are also predecessors of component-*j*.
2) If all the predecessors of component-*i* are predecessors of component-*j*, and component-*i* is connected to component-*j*, then component-*i* is also a predecessor of component-*j*.

2.5. Precedence Constraint Generation Algorithm

An automatic method for generating precedence constraints was developed in the research based on the collision-free assembly path (CFAP), the number of connections, and component volume. All this information can be determined based on the mating condition data obtained from the

CAD database. Alfadhlani et al. (2011) proposed an automatic method to define the assembly collision-free path, and this was adopted in this research.

The CFAP is determined using the component collision detection algorithm (Alfadhlani et al., 2011). The data required as input are taken from the database of the product drawing on the SolidWorks 2005 CAD system and stored in two databases, the component database and the CFAP database. The component database contains information on the list of all pairs of contacted components and their mating type; the volume of each component; the coordinates of the point of contact; and the normal vector directions. All the information was extracted using the component database algorithm (Alfadhlani at al., 2011). The CFAP database contains the same information as the component database, but with the addition of data on the number of connections and the CFAP information of each component (see Figure 3).

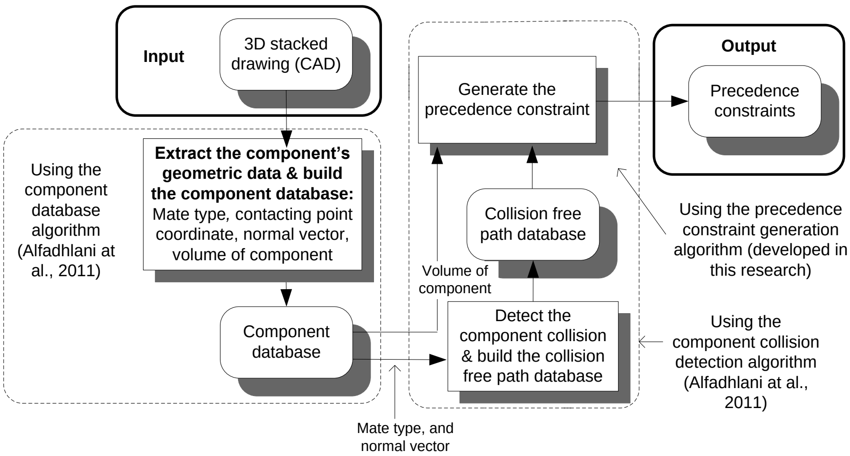


Figure 3 Framework for generating precedence constraints

Furthermore, the priority rules for selecting components, the rules for improving precedence constraints, and the precedence constraint generation steps, as described above, are used in constructing the following precedence constraint algorithm:

(1) Iteration = 0
   i. Set Selected Component List = Ø.
   ii. Identify the CFAP of each component in the CFAP database that has a value equal to 1 on at least one axis; set as a candidate, and save the candidate, its connections and its volume in the Candidate Component List.

(2) Iteration = Iteration +1.
Check the candidates and their connections in the Candidate Component List. Calculate *a* as the number of candidate connections with other components in the Selected Component list. Calculate *b* as the number of candidate connections with other components that have not been selected, *b* = total connections – *a*.

(3) Use **Set A of rules**:
Select a component to be released from the Candidate Component List; that is, the candidate which has the lowest value of *b*. If there is more than one candidate, select that which has the smallest volume. If there is still more than one candidate, choose a component arbitrarily from those which have the smallest value of *b* and the smallest volume. Save the component and the number of connections in the Selected Components List and delete from the Candidate List.

(4) Set all the previously selected components and connect to the new one as its predecessors.

(5) Delete the new selected component from the CFAP database, then re-evaluate the value of the CFAP of each component.

(6) Check all the components in the CFAP database which have a CFAP value equal to 1; set as new candidates and save in the Candidate List.
(7) Check the Candidate List. If there are components that have not been evaluated, go back to step 2; otherwise, go to the next step.
(8) Update the Predecessor List of the last selected component by adding all the components which connect to it.
(9) Update the predecessors of the component using **Set B of rules**:
For all components-*j* and *j* = 1, 2… *N*:

(a) If component-*i* is a predecessor of component-*j*, then all predecessors of component-*i* are also predecessors of components-*j*.
(b) If all predecessors of component-*j* are predecessors of component-*k*, and component-*j* is connected to component-*k*, then component-*j* is also a predecessor of component-*k*.

(10) Save the information of the components and their predecessors on the Precedence Constraint List in the Component Database, then stop the iteration.

The result of this algorithm is the precedence constraints, represented by the relationship between the components and their predecessors. The results are saved in the Component Database. At this stage, the Component Database contains information about the list of all pairs of contacted components and their mating type; the volume of each component; the coordinates of the points of contact; the normal vector directions; and a list of predecessors of each component. A flowchart of precedence constraint generation is shown in Figure 4.

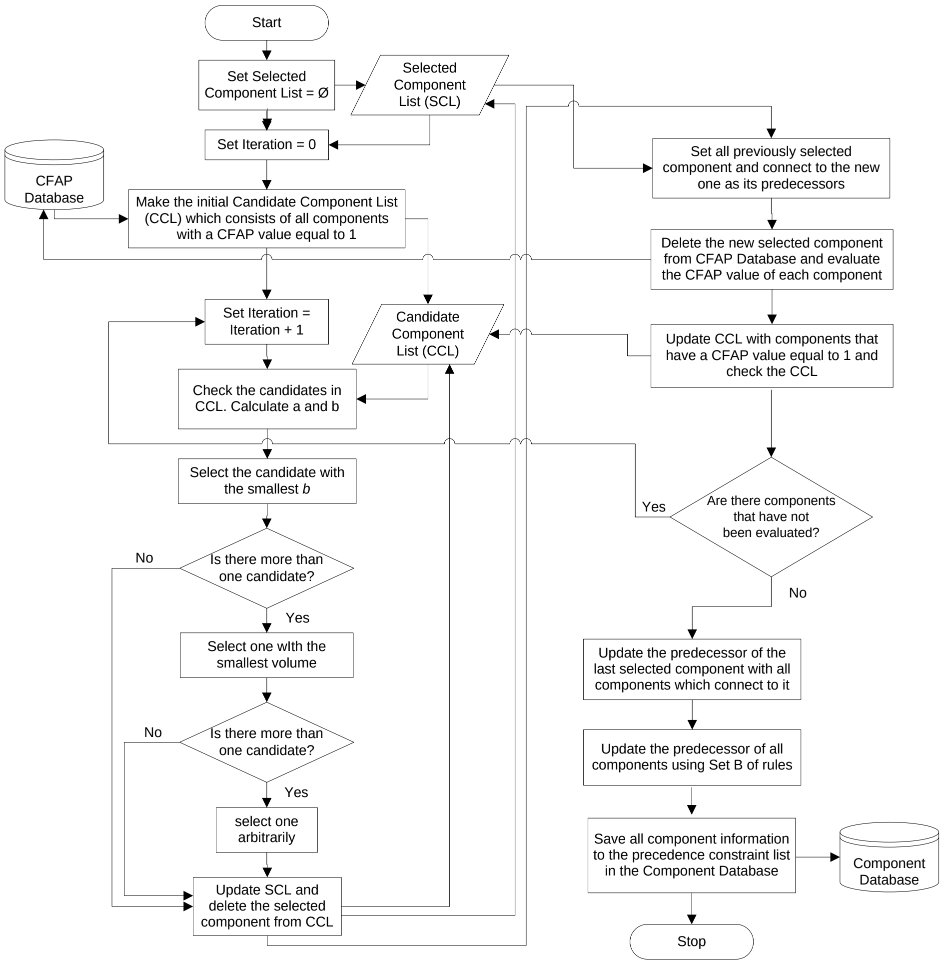


Figure 4 Precedence constraint generation flowchart

3. IMPLEMENTATION EXAMPLE

A bench vice was used to test the algorithm, a product adopted from Tickoo (2004) and redrawn for these particular requirements. Its assembly orientations are multidirectional and orthogonal to the *x*, *y*, and *z*. It consists of 13 components, as shown in Figure 5.

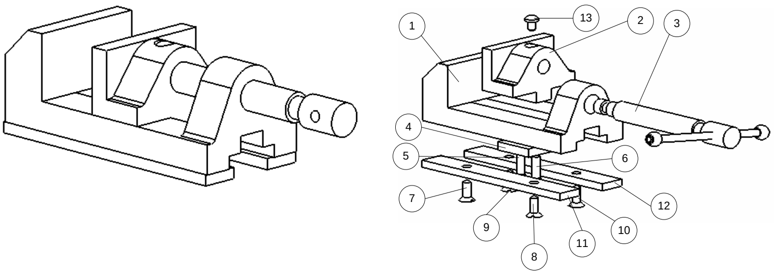


Figure 5 Bench vice assembly (Tickoo, 2004)

The CFAP information, which was determined using the CFAP algorithm of Alfadhlani et al. (2011), is shown in Table 3. This information was used to generate the precedence constraints. From it, it was established that the candidates to be released in the first iteration were components 5, 6, 7, 8, 9, 10 and 13, as these have a value equal to 1 in the collision-free assembly path.

Table 3 CFAP value of each bench vice component
(Initial data, using the method of Alfadhlani et al., 2011)

Component	+*x*	-*x*	+*y*	-*y*	+*z*	-*z*
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	0	0	0	0	0	0
5	0	0	1	0	0	0
6	0	0	1	0	0	0
7	0	0	1	0	0	0
8	0	0	1	0	0	0
9	0	0	1	0	0	0
10	0	0	1	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	1	0	0

Table 4 CFAP value of each bench vice component
(Data after component 13 selected, using the method of Alfadhlani et al., 2011)

Component	+*x*	-*x*	+*y*	-*y*	+*z*	-*z*
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	1	0	0	0	0
4	0	0	0	0	0	0
5	0	0	1	0	0	0
6	0	0	1	0	0	0
7	0	0	1	0	0	0
8	0	0	1	0	0	0
9	0	0	1	0	0	0
10	0	0	1	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0

Table 5 Precedence constraint generation of the bench vice

Iteration	Candidate	Connection components	*a*	*c*	*b*	Volume (mm^3)	Selected component	New candidate	Disassembly direction	Predecessors
1	5	2,4	0	2	2	804.08	13	3	-*y*	-
	6	2,4	0	2	2	804.08				
	7	1,11	0	2	2	545.8				
	8	1,11	0	2	2	545.8				
	9	1,12	0	2	2	545.8				
	10	1,12	0	2	2	545.8				
	13	2,3	0	2	2	483.06				
2	5	2,4	0	2	2	804.08	7	empty	-	-
	6	2,4	0	2	2	804.08				
	7	1,11	0	2	2	545.8				
	8	1,11	0	2	2	545.8				
	9	1,12	0	2	2	545.8				
	10	1,12	0	2	2	545.8				
	3	1,2,13	1	3	2	25334.69				
3	5	2,4	0	2	2	804.08	9	empty	-	-
	6	2,4	0	2	2	804.08				
	8	1,11	0	2	2	545.8				
	9	1,12	0	2	2	545.8				
	10	1,12	0	2	2	545.8				
	3	1,2,13	1	3	2	25334.69				
4	5	2,4	0	2	2	804.08	8	11	+*x*, -*x*, +*y*,+*z*, -*z*	-
	6	2,4	0	2	2	804.08				
	8	1,11	0	2	2	545.8				
	10	1,12	0	2	2	545.8				
	3	1,2,13	1	3	2	25334.69				
5	5	2,4	0	2	2	804.08	10	12	+*x*, -*x*, +*y*,+*z*, -*z*	-
	6	2,4	0	2	2	804.08				
	10	1,12	0	2	2	545.8				
	3	1,2,13	1	3	2	25334.69				
	11	1,4,7,8	2	4	2	16563.72				
6	5	2,4	0	2	2	804.08	5	empty	-	-
	6	2,4	0	2	2	804.08				
	3	1,2,13	1	3	2	25334.69				
	11	1,4,7,8	2	4	2	16563.72				
	12	1,4,9,10	2	4	2	16563.72				
7	6	2,4	0	2	2	804.08	6	4	+*x*,-*x*	-
	3	1,2,13	1	3	2	25334.69				
	11	1,4,7,8	2	4	2	16563.72				
	12	1,4,9,10	2	4	2	16563.72				
8	3	1,2,13	1	3	2	25334.69	11	empty	-	7,8
	11	1,4,7,8	2	4	2	16563.72				
	12	1,4,9,10	2	4	2	16563.72				
	4	1,2,5,6,11,12	2	6	4	6224.54				
9	3	1,2,13	1	3	2	25334.69	12	empty	-	9,10
	12	1,4,9,10	2	4	2	16563.72				
	4	1,2,5,6,11,12	3	6	3	6224.54				
10	3	1,2,13	1	3	2	25334.69	4	empty	-	5,6,11,12
	4	1,2,5,6,11,12	4	6	2	6224.54				
11	3	1,2,13	1	3	2	25334.69	3	1	+*y*	13
								2	-*y*	
12	1	2,3,4,7,8,9, 10,11,12	8	9	1	169238.64	2	empty	-	3,13,4,5,6
	2	1,3,4,5,6,13	5	6	1	57452.1				
13	1	1,3,4,5,6,13	9	9	0	169238.64	1	empty	-	2,3,4,11,7, 8,12,9,10

a = number of candidate connections with components that have yet to be disassembled; *b* = number of candidate connections with components yet to be disassembled; *c* = number of candidate connections

Table 5 shows the priority value, disassembly direction and overall iterations for generating the precedence constraints of the bench vice. Referring to iteration 1, the candidate components to be released are 5, 6, 7, 8, 9, 10 and 13. Because all the candidates have the same value of *b* (the number of candidates connected with components that have been released), then the candidate with the smallest volume, that is component-13, was selected. After component-13 was released the CFAP was then updated, and the results are shown in Table 4. Furthermore, component-3 was entered as a candidate.

This process was continued until 13 iterations. If all the components have been chosen to be released, then the iteration is stopped.

The final step was the improvement of the predecessor list of components using Set B of rules. Since components-3 and -4 are predecessors of component-1, then the predecessors of components-3 and -4 are also predecessors of component-1. Thus, the predecessors of component-1 are 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 and 13.

The predecessors of component-4 are 5, 6, 11 and 12; those of component-11 are 7 and 8; while the predecessors of component-12 are 9 and 10. Thus, the completed predecessors of component-4 are 5, 6, 7, 8, 9, 10, 11 and 12. The predecessor list, which has been improved for each bench vice component, is shown in Table 6, while the assembly precedence diagram of the bench vice components is shown in Figure 6.

Table 6 Predecessors and CFAP value of bench vice components

Component	*+x*	*-x*	*+y*	*-y*	*+z*	*-z*	*Predecessor*
1	0	0	1	0	0	0	2,3,4,5,6,7,8,9,10,11,12,13
2	0	0	0	1	0	0	3,4,5,6,7,8,9,10,11,12,13
3	0	1	0	0	0	0	13
4	1	1	1	0	0	0	5,6,7,8,9,10,11,12
5	0	0	1	0	0	0	-
6	0	0	1	0	0	0	-
7	0	0	1	0	0	0	-
8	0	0	1	0	0	0	-
9	0	0	1	0	0	0	-
10	0	0	1	0	0	0	-
11	1	1	1	0	1	1	7,8
12	1	1	1	0	1	1	9,10
13	0	0	0	1	0	0	-

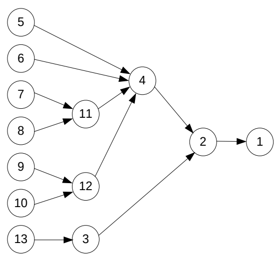


Figure 6 Assembly precedence diagram of bench vice components

4. CONCLUSION

This research was conducted as part of an effort to help product assembly planners to generate feasible assembly sequences automatically. Assembly precedence constraints were determined by the following steps: (1) extract component geometric data from the CAD system; (2) build the component database; (3) detect component collisions; (4) build the collision-free path database; and (5) generate the precedence constraints. The paper proposes a fully automated method for generating precedence constraints using the disassembly approach. A 3D stacked drawing in a solid model is used as input as it has more information than a 2D drawing. The exploded view and the assembly line to obtain connection information for each component, as proposed by Ariastuti et al. (1998) and Toha et al. (2004), and also analysis of the component motion planning, as proposed by Morato et al. (2013), did not need to be performed in this proposed method. Product designers have defined connection types between components when designing products in a CAD system. In SolidWorks, the connection type is defined by the mating type; consequently, the information on the mating type and the volume of the component is used in this method.

A CAD system was used in the development of the models and the algorithm, while the software used was SolidWorks 2005. This CAD system has relatively complete features, so are able to show how the components are assembled to build the final product. We proposed an automatic precedence constraint method in the paper and developed two rules and an algorithm. We built the algorithm using the rules and information on the assembly collision-free path as input. The proposed method was tested, correctly showing the product disassembly precedence constraints. We developed the prototype software in the SolidWorks 2005 CAD system for implementation. The research is a part of an effort which is currently being made to propose an automated method to generate assembly sequences.

5. ACKNOWLEDGEMENT

The author thanks the Institute of Research and Community Service, University of Andalas for providing financial assistance under grant number 01/UN.16/PL/API/2014.

6. REFERENCES

Alfadhlani., Toha, I.S., 2005. Product Assembly Sequence Generation using 3D Solid Drawing as Input. *In:* Proceedings of Seminar VII Production System, pp. 1–13

Alfadhlani, Toha, I.S., 2008. Assembly Sequence Generation for Multidirectional and Orthogonal of Product based on Mating Location Point and Volume of Components. *Jurnal Teknik Industri*, Volume 10(2), pp. 124–137 (in Indonesian)

Alfadhlani, Samadhi, T.M.A.A., Ma'ruf, A., Toha, I.S., 2011. Automatic Collision Detection for Assembly Sequence Planning using a Three-dimensional Solid Model. *Journal of Advanced Manufacturing Systems*, Volume 10(2), pp. 277–291

Ariastuti, R., Taroepratjeka, H., Toha, I.S., 1998. Automatic Assembly Sequence Generation of Products. *Jurnal Teknik dan Manajemen Industri*, Volume 18(1), pp. 26–36 (in Indonesian)

De Fazio, T.L., Whitney, D.E., 1987. Simplified Generation of All Mechanical Assembly Sequences. *IEEE Transactions on Robotics and Automation*, Volume 3(6), pp. 640–658

Delchambre, A., 1992. *Computer-aided Assembly Planning*. Chapman & Hall, London

Lai, H.Y., Huang, C.T., 2004. A Systematic Approach for Automatic Assembly Sequence Plan Generation. *International Journal of Advanced Manufacturing Technology*, Volume 24(9-10), pp. 752–763

Li, J.R., Wang, Q.H., Huang, P., Shen, H.Z., 2010. A Novel Connector-knowledge-based Approach for Disassembly Precedence Constraint Generation. *International Journal of Advanced Manufacturing Technology*, Volume 49(1), pp. 293–304

Morato, C., Kaipa, K.N., Gupta, S.K., 2013. Improving Assembly Precedence Constraint Generation by Utilizing Motion Planning and Part Interaction Clusters. *Computer-Aided Design*, Volume 45(11), pp. 1349–1364

Tickoo, S., 2004. Solid Works for Designers Release 2004. CADCIM Technologies, Schererville, USA

Toha, I.S., Baharuddin, F.R., Ma'ruf, A., 2004. Automatic Assembly Planning using the Assembly Line. *Jurnal Teknik dan Manajemen Industri*, Volume 24(2), pp. 15–27, (in Indonesian)

Tseng, H.E., Li, R.K., 1999. A Novel Means of Generating Assembly Sequences using the Connector Concept. *Journal of Intelligent Manufacturing*, Volume 10(5), pp. 423–435