

# Decomposition-Based Assembly Synthesis for In-Process Dimensional Adjustability

**Byungwoo Lee**

Graduate Student Research Assistant,  
e-mail: byungwoo@engin.umich.edu

**Kazuhiro Saitou**

Assistant Professor,  
e-mail: kazu@umich.edu

Department of Mechanical Engineering,  
University of Michigan,  
Ann Arbor, MI 48109-2125

*This paper presents a method of assembly synthesis focused on the in-process adjustability, where assembly synthesis is defined as the decomposition of the end product design prior to the detailed component design phase. Focusing on the effect of joint configurations on dimensional integrity of complex assemblies, the method recursively decomposes a product configuration and assigns joint configurations according to simple rules, in order to achieve a designed dimensional adjustability and non-forced fit. The rules employed during the decomposition process are drawn from the previous works of assembly design. An augmented AND/OR graph is utilized to represent a process of assembly synthesis with the corresponding assembly sequences, and the algorithm for generating the AND/OR graph is discussed. The method is applied to two dimensional skeletons of products without moving parts at very early stage of the design process. The relation of the assembly synthesis to Datum Flow Chain [1] is discussed. It is also shown that each final design from the assembly synthesis defines its own Datum Flow Chain.*

[DOI: 10.1115/1.1587746]

## Introduction

Body frames of most mechanical products such as ships, airplanes, and automobiles are fairly complex, hence it is very expensive to manufacture them from a single piece material if it is not impossible. Typically, human designers would decompose a complex body structure into parts such as panels and beams so that each part could be manufactured with reasonable cost while satisfying its structural and functional requirements.

As the number of parts increases, however, achieving the dimensional integrity of the final assembly becomes more demanding work due to the inherent manufacturing variations in fabrication and assembly operations. For body structures or frames in which parts are typically forged or bent, it is not economical to manufacture every part with tight tolerance such that tolerance stack-up could be compatible with required dimensional integrity of the final product. Hence, in this type of assemblies, while relative dimensions among parts are specified, the locations of joints are not specified at the part design. Instead, during assembly operations, parts are located and fully constrained in fixtures and they are welded or stamped or drilled for fasteners. In order to adjust relative locations, the contact areas where joints will be placed should be designed in such a way that a small amount of relative motion is allowed, which is why those contact areas are called *slip planes*.

Designers still have to decide how to decompose the product and how to orient slip planes so that they could provide adjustability during the assembly operations. See Fig. 1, for example, where two candidate designs of a rectangular box are shown. Suppose the distance between section 1 and 3 is critical for some reason and parts are assembled on a fixture. Then, it is obvious that the design shown in (a) is not proper due to its lack of adjustability along the critical dimension. On the other hand, the one shown in (b) provides slip planes such that relative location of parts can be adjusted along the critical dimension. Provided that the geometry of the product is fairly complex with a number of critical dimensions to achieve, decomposing the whole piece into parts, configuring slip planes, defining datums, assigning/

analyzing tolerances and planning assembly operations, would be a very tedious process and require several iterations.

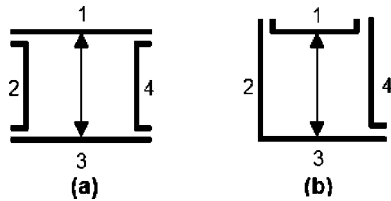
The decomposition and joint configuration have another important effect on designing this type of assembly. Consider two designs of a rectangular box without a critical dimension shown in Fig. 2. Note that, in Fig. 2(a), two slip planes are parallel while those in Figure 2(b) are perpendicular. The design shown in Fig. 2(a) can be assembled effortlessly if the section 3 is the only part that has a manufacturing variation in its length. However, if the lengths of section 2 and 4 are slightly different due to manufacturing variation, some amount of force would be required to clamp two parts together before the joining process. This force required to fit two parts together would result in the residual stress after joining is finished. The residual stress does not only cause excessive stress where fasteners or welding spots are located, but also alters the dimensions of the final product especially when the parts are relatively flexible. By configuring two joints in perpendicular, on the other hand, the design shown in (b) can absorb manufacturing variations that section 1, 2, 3, or 4 may have, provided that variations in angles are negligible. Briefly, the decomposition and joint configuration have effects on the residual stress caused by the manufacturing variations each part has as well as the manufacturing variations the assembly operations introduce.

The decision of which components to assemble together to achieve the end product has been termed *assembly synthesis* in our previous work [2], and the assembly synthesis is done by decomposition of the end product design prior to the detailed component design phase. Among several effects that the assembly synthesis has on the end product, in this paper we focus on the assembly synthesis to achieve dimensional integrity and a method which generates all possible decompositions accompanied with proper joint configurations and assembly sequences that provide in-process dimensional adjustability. For this reason, in the rest of the paper, combined process of decomposition, joint configuration and assembly sequence planning will be referred to as assembly synthesis process.

## Related Work and Background

**Representation of Mechanical Assembly and the Key Characteristic.** The “graphe de liaisons fonctionnelles” [3] is a simple graph devised to represent an assembly. The graph has a node for

Contributed by the Design Automation Committee for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received April 2002. Associate Technical Editor: J. E. Renaud.



**Fig. 1** Examples of decomposition and joint configuration for dimensional adjustment. The design in (b) provides adjustability along the critical dimension, while the design in (a) lacks proper slip planes.

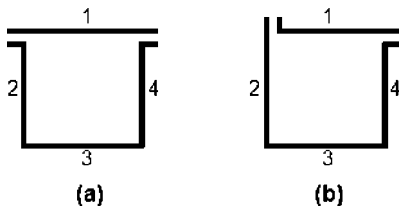
each part in the assembly and an edge for each physical contact that a pair of parts have between them. We shall call it the liaison diagram (or liaison graph) from now on, after De Fazio and Whitney [4].

The Key Characteristics (KCs) are defined by Lee and Thornton [5] as product features, manufacturing process parameters, and assembly features that significantly effect a product's performance, function, and form. As designing a complex assembly in such a way that KCs could be delivered is very crucial, Mantripragada and Whitney [1] devised an augmented liaison diagram called Datum Flow Chain (DFC). The DFC is an efficient tool to analyze how geometric KCs are delivered through datum relationships (represented as directed edges in DFC) among parts and the degrees of freedom joints carry. The critical dimensions mentioned earlier will be referred to as KCs in the rest of the paper.

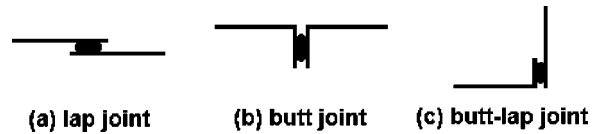
**Assembly Sequence Generation.** Bourjault [3] and De Fazio and Whitney [4] are among the earliest researchers in assembly sequence generation. In both works, a user is required to answer a series of questions such that the software system could set the priorities among the liaisons defined in the assembly. Then the system generates all feasible assembly sequences based on a series of rules built from the user's answers.

Homem de Mello and Sanderson have developed AND/OR graph of assembly sequence and algorithms to generate all feasible assembly sequences in a series of works [6–8]. A decomposition approach is utilized in their method under the assumption that the disassembly sequence is the reverse of assembly sequence no matter the assembly operation is reversible or not.

**Decomposition Methods in Mechanical Design.** In most of aforementioned works on assembly sequence, the use of decomposition was only for generating assembly sequences between physically separate parts, not for designing products. On the other hand, Wang et al. [9,10] utilizes decomposition techniques to physically decompose a given product geometry to separate pieces. Wang developed a system which, at first, unfolds a sheet metal product by searching spanning trees of the face-adjacency graph of the product, then decomposes the unfolded product into several parts by enumerating cut-sets on the spanning trees. The goal of product decomposition, in his research, is optimal manufacturability of resulting parts.



**Fig. 2** Examples of decomposition and joint configuration for non-forced fit. The design in (b) provides slip planes that can absorb manufacturing variation of each part, while the design in (a) lacks the proper configuration of slip planes.



**Fig. 3** The basic joint configurations for sheet metal assemblies [11]

Yetis and Saitou [2] also developed a system that decomposes a given product geometry into parts for minimum reductions in structural stiffness. However, the employed method to search design space differs from most of similar work in that it utilizes the Genetic Algorithm (GA), which randomly generates possible decompositions and evaluate each of them in terms of structural strength, so that the system could find the optimal solution as the GA proceeds.

**Joint Configuration for Sheet Metal Assembly.** Recently, a few researchers have pointed out the influence of joint configurations on the manufacturing variations of sheet metal assemblies. Noting the flexibility of sheet metal assemblies, Liu and Hu [11] have utilized FEM to simulate the manufacturing variations of simple rectangular box constructed from three basic joints shown in Fig. 3. They have shown that variation characteristics widely vary according to the joint configuration. Ceglarek and Shi [12] have developed a method which evaluates the ability of absorbing part variations for a given joint configuration. For a given sheet metal assembly and its joint configuration, the method generates an index which tells how closely parts are related in terms of dimension. Although these works show the effect of joint configuration on the dimension of final assembly using tolerance analysis skills, they lack the ability to *generate* all feasible decomposition and joint configuration from the geometry of initial product design.

## Assembly Synthesis Process

In this section, the assembly synthesis process with the rules that guide the process will be discussed in detail. These rules do not only ensure the dimensional integrity of the end product but also provide the accompanying assembly sequences to achieve the intended dimensional integrity. Starting with the initial product geometry, the assembly synthesis process with these rules will be applied recursively to each decomposed part until the decomposed part is considered to be manufacturable in one piece.

**Representation of Product Geometry.** Since the object under consideration before and during the assembly synthesis process is not an assembly of separate parts yet, a few terms need to be defined to avoid confusion of the object with assembly. A *member* is defined as any section of a product geometry which is allowed to be a separate part after the decomposition process and we state a pair of members are *connected*, when they meet at a certain point in the product geometry.

By the reason that a product geometry (a group of members and connections) is similar in structure to assembly (a group of parts and joints) and, ultimately, a product will be transformed into an assembly after an assembly synthesis process, it is very convenient to use liaison diagram to represent a product geometry and to employ the existing methods developed for decomposition techniques. In a liaison diagram of a product geometry, however, each node represents a member and each edge represents a connection between a pair of members (see Fig. 4). Although, in Fig. 4, every segment with constant tangent and no intersection with other segments is shown as a member, the specification of members might vary according to the area of application or according to a designer's preference. For example, in Fig. 4(a), a designer

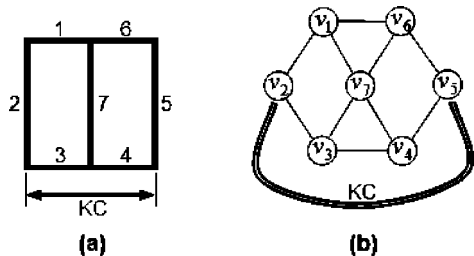


Fig. 4 A product geometry and its liaison diagram

might specify section 1 and 2 as a single member, if he or she want section 1 and 2 to be a single part during the assembly synthesis process.

The liaison diagram of a product geometry can be defined as a five-tuple:

$$L_0 = (V_0, E_0, A_0, id, nv) \quad (1)$$

where  $V_0 = \{v_1, v_2, \dots, v_l\}$  is the set of nodes representing members,  $E_0 = \{e_1, e_2, \dots, e_m\}$  is the set of edges representing connections,  $A_0 = \{a_1, a_2, \dots, a_n\}$  is the set of edges representing KCs,  $id: E \rightarrow Z$  is a mapping which returns an integer for a connection, and  $nv: A \rightarrow R^{nsd}$  is a mapping which returns a normal vector for an KC, where the superscript  $nsd$  on the real number set  $R$ , represents the number of space dimension.

The integer which  $id$  returns is an identification number, to count how many edges in the liaison diagram belongs to a physical connection point. For example, in Fig. 4(a), member 1 has two physical connection points, one with member 2 and the other with member 6 and 7, hence  $v_1$  is adjacent to three nodes,  $v_2$ ,  $v_6$ , and  $v_7$ . Since the liaison diagram without  $id$  does not tell which two of three edges  $((v_1, v_2), (v_1, v_6)$  and  $(v_1, v_7))$  belong to a same connection point, we need an identification number which tells the physical connection point for each edge. In this example, the edge  $(v_1, v_6)$  and  $(v_1, v_7)$  are supposed to have an identical number which should be different from that of  $(v_1, v_2)$ . On the other hand, the  $nv$  returns a vector which tells the orientation of a KC. Both  $id$  and  $nv$  are necessary when the system assigns joint configurations, which will be explained later.

A *configuration* is defined as a group of members which are connected to at least one of members within the group. Once the liaison diagram of a product geometry is given as  $L_0$ , we can represent a configuration as a set of nodes,  $V \subseteq V_0$ , and the liaison diagram on the configuration will be the induced subgraph<sup>1</sup> of  $L_0$  on  $V$ . Since a configuration is a group of connected members, the liaison diagram of a configuration excluding the edges representing KCs should be a connected graph.<sup>2</sup>

**Binary Decomposition of a Configuration.** In the decomposition process, we will allow only such decompositions that divide a configuration into a pair of pieces which we will call *subconfigurations*, as we assume every assembly plan is a series of assembly operations of a pair of parts or subassemblies. In order to decompose a configuration into a pair of subconfigurations, we have to select a group of connections to 'break' such that without those selected connections, the configuration could split into two subconfigurations. Same reasoning can be applied to the liaison diagram of the configuration, hence we will have to select a set of edges to 'cut' in order to split a liaison diagram into two connected graphs. This set of edges is called a *cut-set*<sup>3</sup> (edges representing KCs are not counted in a cut-set). Refer to Fig. 5 for

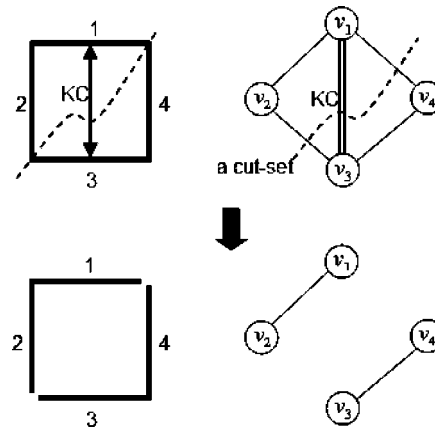


Fig. 5 Binary decomposition is shown both on product geometry (left) and its liaison diagram (right). Removing a cut-set from the liaison diagram resulted in a pair of connected graphs.

an example.

Let us say  $L_a = (V_a, E_a, A_a)$ ,  $L_b = (V_b, E_b, A_b)$  and  $L_c = (V_c, E_c, A_c)$  are liaison diagrams on configuration  $V_a$ ,  $V_b$  and  $V_c$ , respectively. Then the validity of a decomposition from  $V_a$  to its subconfiguration  $V_b$  and  $V_c$  can be decided by predicate  $de$ :

$$de: 2^{V_0} \times (2^{V_0} \times 2^{V_0}) \rightarrow \{true, false\}^4 \quad (2)$$

where  $2^{V_0}$  represents the set of all subsets of  $V_0$  including  $\emptyset$ , and  $de(V_a, (V_b, V_c)) = true$ ,<sup>5</sup> if and only if all of the following conditions are satisfied:

1.  $V_b \neq \emptyset$  and  $V_c \neq \emptyset$
2.  $(V_a, E_a)$ ,  $(V_b, E_b)$  and  $(V_c, E_c)$  are connected.
3.  $V_a = V_b \cup V_c$ .
4.  $V_b \cap V_c = \emptyset$ .

The first condition states that neither of subconfigurations should be empty. And the second condition states that the liaison graphs of the configurations and a pair of subconfigurations should be connected, which means each of them should be a single part. The third and fourth condition specify that the configuration should be divided to a pair of subconfiguration without any common member between them.

In order to obtain all feasible decompositions for a configuration, we have to enumerate all possible cut-sets. Among several methods to enumerate cut-sets [15], we will use the method implemented in the work done by Homem de Mello and Sander-son [7] due to its efficiency on problems of moderate size.

**The First and the Second Decomposition Rules for Dimensional Adjustment.** Once we obtain a binary decomposition of a configuration, we need to assign a proper joint configuration to the broken connections in the decomposition. A *joint configuration* will be represented as normal vectors assigned to broken connections (see Fig. 6). More formerly, a joint configuration can be defined as a mapping  $\gamma: E \rightarrow R^{nsd}$ , which returns a normal vector for each edge which belongs to the cut-set of the decomposition. For edges that do not belong to the cut-set (such as  $(v_1, v_2)$  and  $(v_3, v_4)$  in Fig. 6), the  $\gamma$  will simply return a zero vector. Although  $\gamma$  is associated to  $E$ , it is not unique for a product geometry, as each decomposition will have its own cut-set and a different joint configuration. Also it should be noted that the nor-

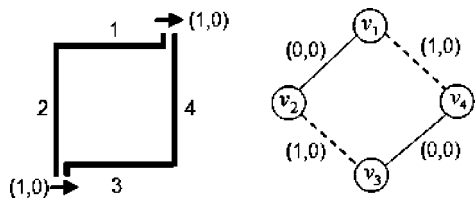
<sup>1</sup>Given a graph  $G=(V,E)$ , let  $U$  be a nonempty subset of  $V$ . The graph whose vertex set is  $U$  and whose edge set comprises exactly the edges of  $E$  which join vertices in  $U$  is termed an *induced subgraph* of  $G$  [13].

<sup>2</sup>A graph  $G$  is termed *connected* if every pair of vertices in  $G$  are joined by a path [13].

<sup>3</sup>A *cut-set* in a connected graph  $G=(V,E)$ , is a minimal set of edges of  $E$  whose removal from  $G$ , renders  $G$  disconnected [13].

<sup>4</sup> $A \times B = \{\binom{a}{b} | a \in A, b \in B\}$  [14].

<sup>5</sup>A parenthesized pair of objects usually have an order between the objects. However, in this context, a pair of subconfigurations obtain the same result from the predicate regardless of the order. Thus,  $(V_b, V_c)$  and  $(V_c, V_b)$  are not distinguished in the rest of the paper when they represent a pair of subconfigurations.



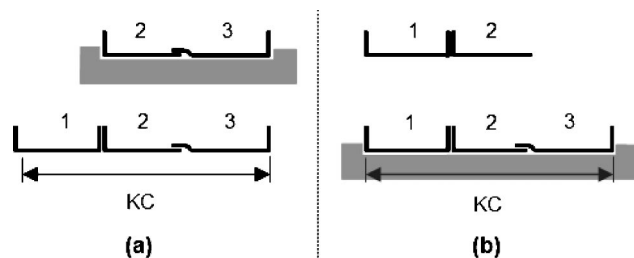
**Fig. 6** A decomposition and its joint configuration is depicted. The joint configuration is a set of normal vectors associated with the cut-set of the decomposition.

mal vectors in the opposite directions are treated as identical. For example,  $(-1,0)$  and  $(1,0)$  are not differentiated in the problem.

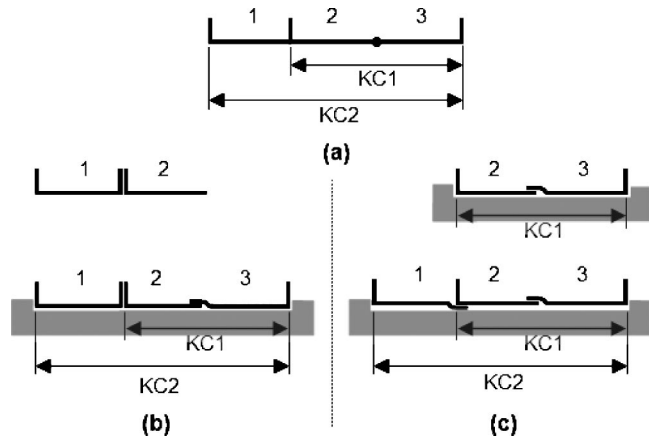
To begin with, let us consider how to decide joint configurations for those decompositions which have at least one broken KC. Provided that fixtures are used to achieve KCs due to part variations, at least one slip plane should be laid along each KC's direction. Figure 7 [16] shows an simplified car floor pan design which has a slip plane between parts 2 and 3 such that the KC can be delivered. However, as pointed out by Whitney et al. [16], assembly sequence planning plays a significant role in KC's delivery. The assembly sequences in Fig. 7 show that it is desirable that a slip plane is provided at the very assembly operation where KC is realized (Fig. 7 (b)). In Fig. 7 (a), although there is a slip plane in the KC's direction, the adjustability which the slip plane provides will be used up by the time KC is realized. Thus, there will be no means left to absorb the variation of part 1.

Let us consider this relationship among slip planes, assembly sequences and delivering KCs in the reverse direction, so that we could decompose a configuration in such a way that KCs are delivered when parts are assembled. As shown in the example in Fig. 7 (b), no matter what joint configuration the part 1 and 2 have between them, it is important that joint configuration between  $\{1,2\}$  and 3 should be parallel to the KC's direction. This can be stated in the reverse course as follows: no matter how a sub-configuration is decomposed further, when a KC is broken by a decomposition, slip planes should be oriented parallel to the KC's direction at all the broken connections. We will refer to this as the *1st decomposition rule for dimensional adjustability*.

As typical complex assemblies have more than one KC involved in several parts, it is common that several KCs cannot be realized *independently*, meaning that "each KC may not be able to have its own DFC that can be built, inspected, and/or adjusted into compliance separately from others [16]." This situation is called *KC conflict* by the same authors. Regarding the KC conflict as undesirable but inevitable, they also addressed that "lacking the ability to adjust each KC independently, one is forced to accord



**Fig. 7** Two assembly sequences for the car floor pan [16]. In figure (a) part 2 and 3 are assembled first and then part 1 is assembled. It is possible that the manufacturing variation of part 1 makes the KC unattainable. However, the sequence shown in figure (b) provides the slip plane at the moment the KC is achieved, so that the slip plane can absorb any variation in length involved with the KC.



**Fig. 8** The car floor pan with two KCs is shown in (a). Two assembly sequences with different joint configurations are shown in (b) and (c) [16]. While the assembly sequence in (b) achieves two KCs at a time with only one slip plane (between part 2 and 3), that of (c) achieves two KCs in sequence with one slip plane for each KC.

one or the other looser tolerances, a process called *prioritizing the KCs*," which warns us of the possibility that a KC can be delivered less precisely under a KC conflict.

In Fig. 8(a), a product geometry with two KCs is shown. While the assembly sequence shown in Fig. 8(b) realizes both KCs at the second operation, the planning shown in Fig. 8(c) achieves one KC at a time with one slip plane or adjustability for each KC. Also note that joint configurations in (b) and (c) are different due to the number of slip planes that they provide. We can program our assembly synthesis method to come up with design (c), by allowing only such decompositions that break at most one KC at a time (*2nd decomposition rule for dimensional adjustability*). However, it is possible that the decomposition process encounter a certain configuration where breaking only one KC is not possible by any means. In this case, our method will leave the configuration undecomposed, as a systematic way to prioritizing KCs is not incorporated in the current implementation.

The predicate  $da$  defined below will decide whether a joint configuration satisfies both 1st and 2nd decomposition rules for dimensional adjustability:

$$da: 2^{V_0} \times (E \rightarrow R^{nsd}) \times (2^{V_0} \times 2^{V_0}) \rightarrow \{true, false\} \quad (3)$$

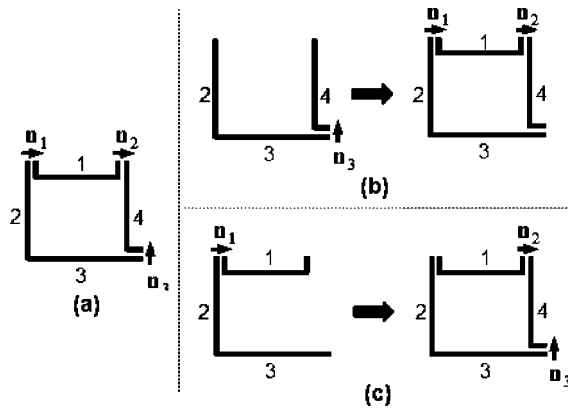
where  $da(V_a, \gamma_i, (V_b, V_c)) = true$ , if and only if all of the following conditions are satisfied:

1.  $|A_a \setminus (A_b \cup A_c)| \leq 1$ .
2. For  $a \in (A_a \setminus (A_b \cup A_c))^6$  and  $\forall e \in (E_a \setminus (E_b \cup E_c))$ ,  $nv(a) \cdot \gamma_i(e) = 1$ .
3.  $\forall e \in (E_b \cup E_c), \gamma_i(e) = \mathbf{0}$ .

The first condition states that the number of broken KC should be 1. The second condition states that the joints at broken edges (edges belong to the cut-set) should be parallel to the broken KC. The third condition states that normal vectors for unbroken edges should be the zero vector.

**Decomposition Rule for Nonforced Fit.** Figure 9(a) depicts a product geometry decomposed with normal vectors represented as  $\mathbf{n}_1$ ,  $\mathbf{n}_2$ , and  $\mathbf{n}_3$  for three joints. Two candidate assembly sequences are shown in Figure 9(b) and (c). In the figure (b),  $\{2,3\}$  and 4 are assembled first and  $\{\{2,3\},4\}$  and 1 are assembled at the second operation. Because there is no KC defined between member 2 and member 4, a fixture is not used at the first operation; hence some level of manufacturing variation might occur

<sup>6</sup> $A \setminus B = \{x | x \in A \text{ and } b \notin B\}$  [14]



**Fig. 9** Two different assembly sequences for a decomposition (a) are shown in (b) and (c). While, in figure (b), the joint configuration at the second assembly operation does not absorb possible manufacturing variations from the first assembly operation and part 1, those in figure (b) does absorb the manufacturing variations, thus enables non-forced fit.

during the first operation. Also, there might be manufacturing variation in the length of the part 1. Since the joint configuration at the following operation does not allow any absorption of manufacturing variations (as both  $\mathbf{n}_1$  and  $\mathbf{n}_2$  are parallel to member 1), it is possible that excessive force is required to assemble part 1 to the rest of the assembly. This is not desirable because such excessive force to fit one part with the others might result in residual stress after a joining operation. On the other hand, the assembly sequence shown in Fig. 9(c) does not exhibit such problem. At the second operation, joints are oriented perpendicular to each other such that they could absorb manufacturing variations from both the previous assembly operation and part 1. Therefore, when a decomposition breaks two or more connections, we will allow only the joints whose normal vectors are perpendicular to each other. We will refer to this rule as the *decomposition rule for nonforced fit*. When a problem is defined in two-dimensional space, any decomposition with more than two joints will not satisfy the rule, since, in two-dimensional space, we can not have three or more vectors perpendicular to each other. Also, it should be noted that it is a very simplified rule for non-forced fit, especially when a product under consideration is very flexible so that variations in straightness and angle of assembly tabs are not negligible compared to those in length.

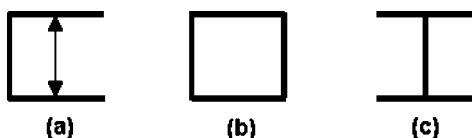
A predicate describing the decomposition rule for non-forced fit can be defined as follows:

$$\text{nff}: 2^{V_0} \times (E \rightarrow R^{nsd}) \times (2^{V_0} \times 2^{V_0}) \rightarrow \{true, false\} \quad (4)$$

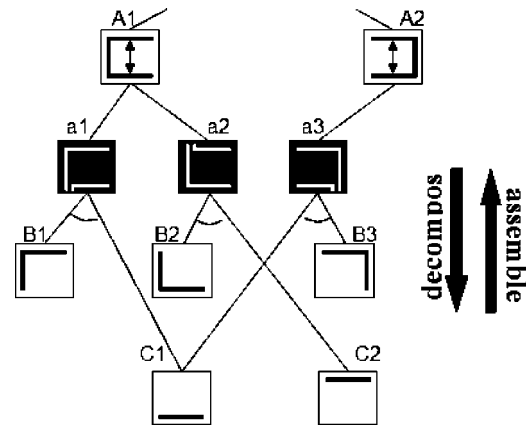
where  $\text{nff}(V_a, \gamma_i, (V_b, V_c)) = true$ , if and only if all of the following conditions are satisfied:

1.  $|\{\text{id}(e) | e \in E_a \setminus (E_b \cup E_c)\}| \leq nsd$ .
2. For  $\forall e_p, e_q \in (E_a \setminus (E_b \cup E_c))$  such that  $\text{id}(e_p) \neq \text{id}(e_q)$ ,  $\gamma_i(e_p) \cdot \gamma_i(e_q) = 0$ .
3. For  $\forall e \in (E_b \cup E_c)$ ,  $\gamma_i(e) = \mathbf{0}$ .

The first condition states that the number of edges in the cut-set should be less than  $nsd$  since it will not satisfy the second con-



**Fig. 10** Examples of configurations that need further decompositions



**Fig. 11** A partial AND/OR graph of the simple rectangular box (Fig. 5). Note that the graph is constructed from the top to the bottom as the assembly synthesis is being conducted and it reads from the bottom to the top when an assembly sequence is extracted.

dition otherwise. The second condition states that the joints at broken edges (edges belong to the cut-set) should be perpendicular to each other. The third condition states that normal vectors for unbroken edges should be the zero vector.

**Criteria for Stopping Decomposition.** Since the decomposition is motivated by the difficulties in manufacturing the final product in a single piece, it is natural to stop decomposing a configuration when the configuration does not have such difficulties anymore. Although the criteria for stopping decomposition will vary depending on the chosen fabrication and assembly processes, we have employed following three simple criteria which define predicate  $\text{stop\_de}$ . Assuming components are to be manufactured via sheet metal stamping, any configuration that falls under any one of following criteria (predicate  $\text{stop\_de}$  returns *false* for this configuration) will be decomposed further.

1. Any configuration that has a KC (Fig. 10 (a)) since we have assumed that KCs can not be achieved with the tolerance level of part fabrication.
2. Any configuration that has a closed section (Fig. 10 (b)).
3. Any configuration that has a connection point where three or more members meet (Fig. 10 (c)).

**Representation of Decomposition and Assembly Sequence.** The AND/OR graph of assembly plans [6] is employed to represent the assembly synthesis with accompanying assembly sequences. Since the original AND/OR graph of assembly plans was devised to show all feasible assembly sequences of a given assembly with fixed joint configurations, it did not have to show joint configurations. However, since our assembly synthesis process assigns joint configurations as the product geometry is being decomposed, they should be represented in the *AND/OR graph of assembly synthesis*. A partial AND/OR graph of the simple rectangular box (Fig. 5) is depicted in Fig. 11, where the joint configurations are represented in nodes with black background. The nodes in white background represent configurations. In the figure, for example, the configuration A1 is decomposed into B1 and C1 with the joint configuration of a1. Note that A1 can be decomposed into either (B1 'AND' C1 through a1), 'OR' (B2 'AND' C2 through a2).

As we recall that a configuration is a set of members  $V_i \subseteq V_0$  and a joint configuration is a mapping  $\gamma: E \rightarrow R^{nsd}$ , each node in white background contains a subset of all members and each node in black background contains a mapping. A set of three lines which connects a configuration  $V_a$ , a joint configuration  $\gamma_i$ , and a pair of subconfigurations  $(V_b, V_c)$  is a hyper-edge, represented

```

BUILD_AO( $L_0$ )
1  global  $S, J, F \leftarrow \phi$ 
2  if STOP_DCMP( $V_0$ ) = FALSE then DECOMPOSE( $V_0$ )
3  AO  $\leftarrow (S, J, F)$ 
4  return AO

DECOMPOSE( $V_i$ )
5  SubConfig  $\leftarrow$  GET_FSBL_CON( $V_i$ )
6  ( $v_i$  is_v_new)  $\leftarrow$  FIND_OR_MAKE_NODE( $V_i$ )
7  while SubConfig  $\neq \phi$ 
8     $V_s$   $\leftarrow$  POP(SubConfig)
9     $V_c$   $\leftarrow V_s \setminus V_i$ 
10   JointConfig  $\leftarrow$  GET_FSBL_JOINTS( $V_s, (V_s, V_c)$ )
11   if JointConfig  $\neq \phi$  then
12     ( $x_i$  is_x_new)  $\leftarrow$  FIND_OR_MAKE_NODE( $V_s$ )
13     ( $y_i$  is_y_new)  $\leftarrow$  FIND_OR_MAKE_NODE( $V_c$ )
14     while JointConfig  $\neq \phi$ 
15        $\gamma$   $\leftarrow$  POP(JointConfig)
16        $J \leftarrow J \cup \{w \leftarrow$  MAKE_NEW_NODE( $\gamma$ )\}
17        $F \leftarrow F \cup \{\text{MAKE\_NEW\_HYPER\_EDGE}(v_i, w_i, (x_i, y_i))\}$ 
18       if STOP_DCMP( $V_i$ ) = FALSE and is_x_new = TRUE
19         then DECOMPOSE( $V_s$ )
20       if STOP_DCMP( $V_i$ ) = FALSE and is_y_new = TRUE
21         then DECOMPOSE( $V_c$ )

```

Fig. 12 Algorithm BUILD\_AO to generate the AND/OR graph of assembly synthesis

as  $(V_a, \gamma_i, (V_b, V_c))$  which is also the representation of a decomposition with a joint configuration. The AND/OR graph of assembly synthesis is then represented as a three-tuple:

$$AO = (S, J, F) \quad (5)$$

where  $S$  is a set of nodes representing configurations,  $J$  is a set of nodes representing joint configurations, and  $F$  is a set of hyper-edges. The necessary conditions for a hyper-edge  $(V_a, \gamma_i, (V_b, V_c))$  in  $F$  are as follows:

1. stop\_de( $V_a$ ) = false
2. de( $V_a, (V_b, V_c)$ ) = true.
3. da( $V_a, \gamma_i, (V_b, V_c)$ ) = true.
4. nff( $V_a, \gamma_i, (V_b, V_c)$ ) = true. (6)

Then  $AO = (S, J, F)$  can be defined by following rules:

1. If stop\_de( $V_0$ ) = false,  $V_0 \in S$ .
2. For  $\forall V_a \in S$ , if there exist  $\exists \gamma_i, V_b, V_c$  such that  $f = (V_a, \gamma_i, (V_b, V_c))$  satisfies necessary conditions (6), then  $\gamma_i \in J, V_b \in S, V_c \in S$  and  $f \in F$ .
3. No element is in  $S, J$  and  $F$ , unless it can be obtained by using rules 1 and 2. (7)

### Decomposition Algorithm

Figure 12 shows the algorithm used for the generation of the AND/OR graph of assembly synthesis. The algorithm consists of the main procedure BUILD\_AO and DECOMPOSE, which recursively decomposes configurations. The basic recursive structure of DECOMPOSE borrows from Lee [17]. For a given product geom-

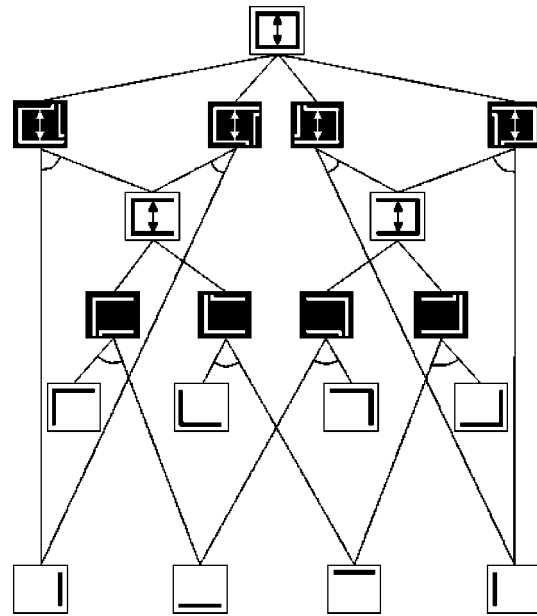


Fig. 13 The AND/OR graph for the simple rectangular box in Fig. 5

etry  $L_0$ , the main procedure BUILD\_AO initializes global variables  $S, J$  and  $F$  (line 1) which will be updated during recursive decomposition process to become the AND/OR graph of assembly synthesis.

The BUILD\_AO then calls DECOMPOSE with the initial configuration  $V_0$ , if STOP\_DCMP( $V_0$ ) returns FALSE (line 2). The function STOP\_DCMP returns FALSE if and only if  $V_0$  does not satisfy the predicate stop\_de. The DECOMPOSE in turn calls GET\_FSBL\_CON to get all feasible pairs of subconfigurations that satisfy predicate de (line 5). The GET\_FSBL\_CON actually returns only one of the pair with less elements, and the other subconfiguration is obtained later at line 9. If SubConfig returned by GET\_FSBL\_CON is empty, DECOMPOSE terminates without updating  $S, J$ , and  $F$  (line 7). If the SubConfig is not empty, DECOMPOSE calls GET\_FSBL\_JOINTS for each pair of subconfigurations to obtain feasible joint configurations that satisfy predicates da and nff (line 10). If JointConfig is not empty, DECOMPOSE locates an existing node or makes a new node that represents the subconfiguration and updates  $S$  within FIND\_OR\_MAKE\_NODE (lines 12 and 13).<sup>7</sup> Then DECOMPOSE makes a new node representing the joint configuration and updates  $J$  in line 16 and draw a new hyper edge and update  $F$  in line 17, which starts from the configuration, passes the joint configuration and reaches the binary subconfigurations newly made or found. After updating  $S, J$ , and  $F$ , DECOMPOSE calls STOP\_DCMP to check if the pair of subconfigurations need further decomposing (line 18 and 20). If a subconfiguration fails STOP\_DCMP and if the subconfiguration was newly generated at line 12 or 13, it calls DECOMPOSE recursively to decompose it further (in line 19 and 20). If the subconfiguration was 'found' instead of 'made' (the boolean flag is\_x\_new in line 12 or is\_y\_new in line 13 is false in this case), it means the decomposition was attempted on the subconfiguration once when it was updated in  $S$ , thus DECOMPOSE will not be called for this subconfiguration. Theorem 1 tells the correctness and the completeness of BUILD\_AO in the general case of initial liaison diagram  $L_0$ .

<sup>7</sup>For example, in Fig. 11, suppose node B1 and C1 have already created by decomposing A1. Therefore, when A2 is decomposed, FIND\_OR\_MAKE\_NODE will find C1 existing instead of making the same node.

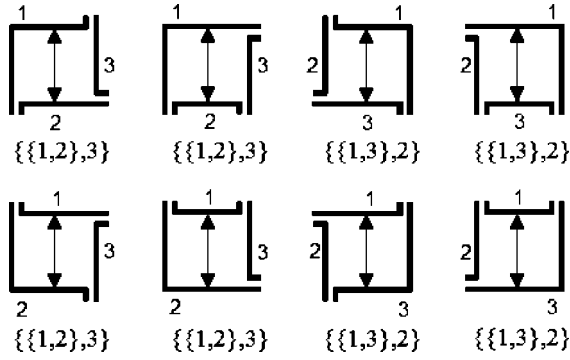


Fig. 14 All final designs with accompanying assembly sequences interpreted from the AND/OR graph in Fig. 13

**Theorem 1.** BUILD\_AO is correct and complete given that GET\_FSBL\_CON, GET\_FSBL\_JOINTS and STOP\_DCMP are correct and complete.

*Proof.* Let  $AO=(S,J,F)$  be the AND/OR graph defined by the rules in (7) and  $AO'=(S',J',F')$  be the AND/OR graph generated by BUILD\_AO. At first, we will prove the correctness by showing that  $S' \subseteq S$ ,  $J' \subseteq J$  and  $F' \subseteq F$ . Then the completeness can be proved by showing that  $S \subseteq S'$ ,  $J \subseteq J'$  and  $F \subseteq F'$ . The proof is done using mathematical induction on the  $n$  which is the number of decompositions. As a matter of convenience, any local object obtained within DECOMPOSE( $V_{ia}$ ) will be denoted by *ObjectName*<sup>*i*</sup>.

• **Proof of correctness:**  $(S' \subseteq S) \wedge (J' \subseteq J) \wedge (F' \subseteq F)$ .

Let any path (a series of hyper-edges) starting from the top in  $AO'=(S',J',F')$  be  $f_1, f_2, \dots, f_n$ , where  $f_i = (V_{ia}, \gamma_i, (V_{ib}, V_{ic}))$ ,  $V_{1a} = V_0$ ,  $|V_{ib}| \leq |V_{ic}|$  and, either  $V_{ia} = V_{(i-1)b}$  or  $V_{ia} = V_{(i-1)c}$  for  $i \geq 1$ . We will show that  $f_1, f_2, \dots, f_n$  and their components exist also in  $AO=(S,J,F)$ .

1. Suppose the statement is false when  $n$  is 1. In other words,  $V_{1a} \notin S$ ,  $V_{1b} \notin S$ ,  $V_{1c} \notin S$ ,  $\gamma_1 \notin J$  or  $f_1 \notin F$ . When  $V_{1a} \notin S$ ,  $\text{stop\_de}(V_{1a}) = \text{true}$  by the first rule of (7). Thus,  $\text{STOP\_DCMP}(V_{1a}) = \text{TRUE}$  at line 2 since STOP\_DCMP is correct. In this case, BUILD\_AO terminates leaving  $S'$ ,  $J'$  and  $F'$  empty, which contradicts  $V_{1a} \in S'$ . Therefore,  $V_{1a} \in S$ . In case  $V_{1b} \notin S$ ,  $V_{1c} \notin S$ ,  $\gamma_1 \notin J$  or  $f_1 \notin F$ , by rule 2 in (7), it is obvious that  $f_1$  or some of its components does not satisfy one of necessary conditions in (6). When  $\text{de}(V_{1a}, (V_{1b}, V_{1c})) = \text{false}$ , then  $V_{1b} \notin \text{SubConfig}^1$  at line 5. Therefore,  $V_{1b}$  can never be assigned to  $V_b^1$  at line 8, and, as a result,  $V_{1b}$ ,  $V_{1c}$ ,  $\gamma_1$  and  $f_1$  cannot have a chance to be updated in  $S'$ ,  $J'$  and  $F'$  respectively, which contradicts the assumption. If  $\text{da}(f_1) = \text{false}$  or  $\text{nff}(f_1) = \text{false}$ , then  $\gamma_1 \notin \text{JointConfig}^1$  at line 10. Thus,  $V_{1b}$ ,  $V_{1c}$ ,  $\gamma_1$  and  $f_1$  can not be updated in  $S'$ ,  $J'$  and  $F'$ , which contradicts the assumption. Hence, the statement is true when  $n$  is 1.

2. Suppose the statement is true if  $n=k$  but false if  $n=k+1$ . If the statement is true when  $n=k$ ,  $V_{kb}, V_{kc} \in S$ . Since one of  $V_{kb}$  and  $V_{kc}$  is equivalent to  $V_{(k+1)a}$ ,  $V_{(k+1)a} \in S$ . Therefore, when the statement is false for  $n=k+1$ ,  $V_{(k+1)b} \notin S$ ,  $V_{(k+1)c} \notin S$ ,  $\gamma_{k+1} \notin J$  or  $f_{k+1} \notin F$ . Then, by rule 2 in (7),  $f_{k+1}$  or some of its components does not satisfy one of necessary conditions in (6). Suppose  $\text{stop\_de}(V_{(k+1)a}) = \text{true}$ . Then,  $\text{STOP\_DCMP}(V_{(k+1)a}) = \text{TRUE}$  at line 18 or 20 in the DECOMPOSE where  $V_{(k+1)a}$  is updated in  $S'$ . Hence, DECOMPOSE can never be called for  $V_{(k+1)a}$ , which contradicts  $\gamma_{k+1} \in J'$  and  $f_{k+1} \in F'$ . When  $\text{de}(V_{(k+1)a}, (V_{(k+1)b}, V_{(k+1)c})) = \text{false}$ ,  $V_{(k+1)b} \notin \text{SubConfig}^{k+1}$  at line 5. Therefore,  $V_{(k+1)b}$  can never be assigned to  $V_b^{k+1}$  at line 8, and, as a result,  $\gamma_{k+1}$  and  $f_{k+1}$  cannot have a chance to be updated in  $J'$  and  $F'$  respectively,

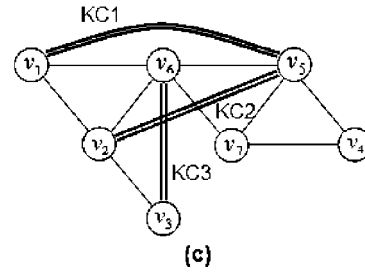
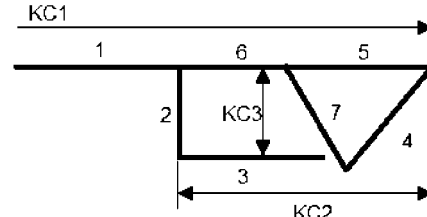
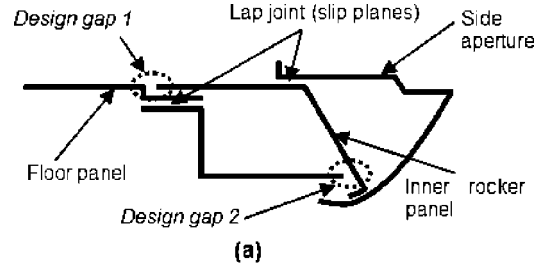


Fig. 15 An automotive side aperture design (a) borrowed from [12] is simplified in (b) with a few KCs assumed. Figure (c) shows the liaison diagram of the initial geometry.

which contradicts the assumption. If  $\text{da}(f_{k+1}) = \text{false}$  or  $\text{nff}(f_{k+1}) = \text{false}$ , then  $\gamma_{k+1} \notin \text{JointConfig}^{k+1}$  at line 10. Thus,  $\gamma_{k+1}$  and  $f_{k+1}$  cannot be updated in  $J'$  and  $F'$ , which contradicts the assumption. Hence, the statement is true when  $n$  is 1. Hence the statement is true when  $n=k+1$ , if it is true  $n=k$ .

• **Proof of completeness:**  $(S \subseteq S') \wedge (J \subseteq J') \wedge (F \subseteq F')$ .

Let any path (a series of hyper-edges) starting from the top in  $AO=(S,J,F)$  be  $f_1, f_2, \dots, f_n$ , where  $f_i = (V_{ia}, \gamma_i, (V_{ib}, V_{ic}))$ ,  $V_{1a} = V_0$ ,  $|V_{ib}| \leq |V_{ic}|$  and, either  $V_{ia} = V_{(i-1)b}$  or  $V_{ia} = V_{(i-1)c}$  for  $i \geq 1$ . We will show that  $f_1, f_2, \dots, f_n$  and their components exist also in  $AO'=(S',J',F')$ .

1. If  $n$  is 1,  $\text{STOP\_DCMP}(V_{1a}) = \text{FALSE}$  and DECOMPOSE is called for  $V_{1a} (= V_0)$  at line 2, because  $V_{1a} \in S$  which means  $\text{stop\_de}(V_{1a}) = \text{false}$  by the rule 1 in (7). Also, since  $\text{de}(V_{1a}, (V_{1b}, V_{1c})) = \text{true}$  by the rule 2 in (7),  $V_{1b} \in \text{SubConfig}^1$  returned by GET\_FSBL\_CON( $V_{0a}$ ) (line 5) and  $V_{1a}$  is updated in  $S$  (line 6). Therefore, at a certain iteration in the while loop at line 7,  $V_{1b}$  is popped from  $\text{SubConfig}^1$  and assigned to  $V_b^1$  (line 7).  $V_{1c}$  is drawn and assigned to  $V_c^1$  at line 8. Since both  $\text{da}(f_1)$  and  $\text{nff}(f_1)$  are true by the rule 2 in (8),  $\gamma_1 \in \text{JointConfig}^1$  returned at line 10. Thus  $V_{1b}$  and  $V_{1c}$  are updated in  $S'$  (line 12 and 13) and  $\gamma^1$  is recorded in  $J'$  (line 16). And,  $(V_a^1, \gamma^1, (V_b^1, V_c^1))$  is updated in  $F'$  (line 17).
2. Suppose the statement is true for  $n=k$ . Since  $f_k \in F$ , it satisfies the necessary conditions in (6). Thus  $\text{stop\_de}(V_{(k+1)a}) = \text{false}$ . Also, since  $f_k \in F'$  by the assumption,  $V_{(k+1)a}$  must have been added in  $S'$  as a sub-configuration of some  $V_{pa} \in S'$ . Since  $\text{STOP\_DCMP}(V_{(k+1)a})$

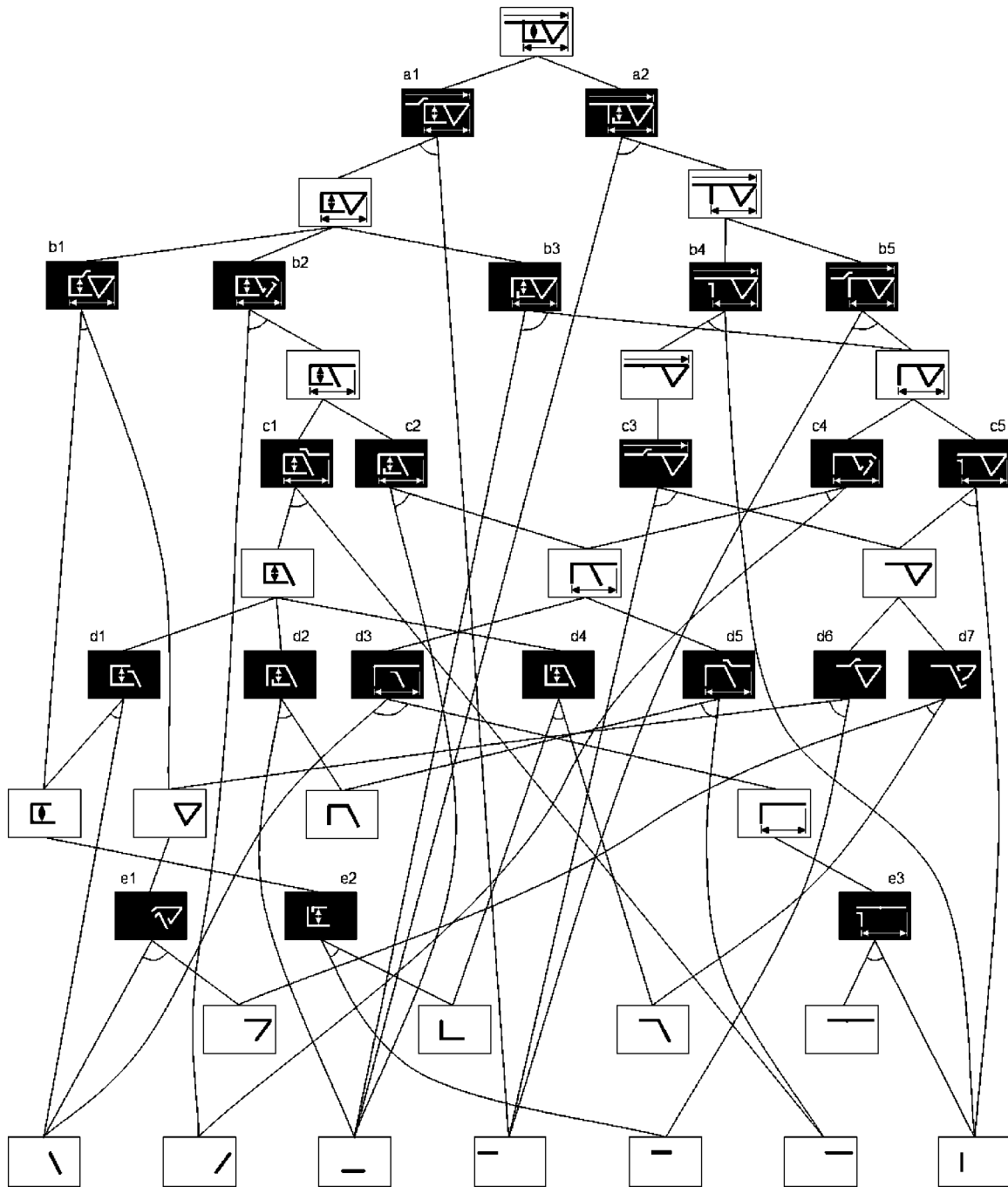


Fig. 16 A partial AND/OR graph of assembly synthesis for the aperture design in Fig. 15(b)

$=FALSE$  by the fact that  $stop\_de(V_{(k+1)a})=false$  and  $is\_x\_new^p=true$  (or  $is\_y\_new^p=true$ ), DECOMPOSE is called for  $V_{(k+1)a}$  at line 19 (or 20). Following the same logic utilized in the step 1, once DECOMPOSE( $V_{(k+1)a}$ ) is called, it is obvious that  $V_{(k+1)b}$ ,  $V_{(k+1)c}$ ,  $\gamma_{k+1}$  and  $f_{k+1}$  are updated in  $S'$ ,  $J'$  and  $F'$ , respectively. Hence the statement is true when  $n=k+1$ , if it is true  $n=k$ .

According to Theorem 1, DECOMPOSE is correct and complete given that GET\_FSBL\_CON, GET\_FSBL\_JOINTS and STOP\_DCMP are correct. However, we have to limit the completeness of GET\_FSBL\_JOINTS due to a practical reason, since there can be infinite number of joint configurations (as long as the decomposition does not break a KC and two normal vectors of

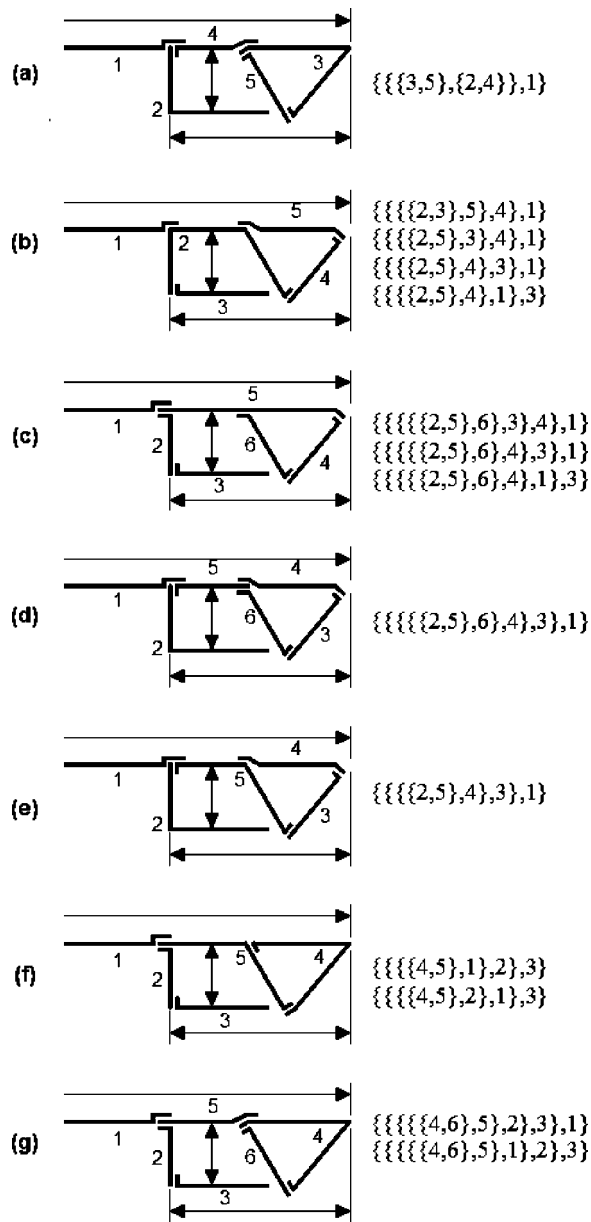
two joints are perpendicular) that satisfy predicates  $da$  and  $nff$ . A user needs to confine each joint to a few angles. In our examples, we only allow a joint to be oriented either parallel or perpendicular to one of members it connects.

The computer software embodying the algorithm in Fig. 12 is written in C++ with the intense use of the data structures and algorithms of LEDA (Library of Efficient Data Types and Algorithms) developed at Max-Planck-Institute für Informatik, Saarbrücken, Germany.

### Examples

**Four-Member Rectangular Box.** The complete AND/OR graph for the simple rectangular box from Fig. 5 is shown in Fig.



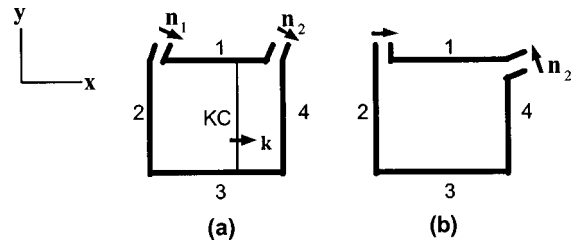


**Fig. 17 All final designs with accompanying assembly sequences interpreted from the AND/OR graph in Fig. 18. Since the AND/OR graph in Fig. 16 is partial, neither final designs nor assembly sequences for each final design are listed completely.**

13. All feasible decompositions with joint configurations are interpreted in Fig. 14. The numbers 1, 2 and 3 designated at each final design in Fig. 14 indicate parts in the synthesized assembly.

**Automotive Side Aperture Design.** Figure 15(a) depicts a typical automotive side aperture design borrowed from Ceglarek and Shi [12]. The design gaps shown in the figure are not KCs, but to avoid potential interference caused by manufacturing variations. The initial geometry of the aperture design might have before the assembly synthesis is shown in Fig. 15(b) with a few KCs assumed. Figure 15(c) shows the liaison diagram of the initial product geometry shown in Fig. 15 (b).

Due to the space limit, only a part of the complete AND/OR graph of the aperture design (Fig. 15 (b)) is shown in Fig. 16. The partial AND/OR graph in Fig. 16 contains 24 configurations and 20 joint configurations while the complete graph has 44 and 123,



**Fig. 18 Nonstrict cases for the decomposition rules for dimensional adjustability (a) and the decomposition rule for nonforced fit (b).**

respectively. Figure 17 lists all final designs and assembly sequences contained in the partial AND/OR graph in Figure 16. Note that only the assembly sequences that contained in the partial AND/OR graph are listed for each final design in Fig. 17, and they are not the complete list of feasible assembly sequences for the particular final design. Each assembly sequence of a final design corresponds to a unique combination of hyper edges or nodes representing joint configurations in the AND/OR graph (i.e. a 'tree' in the AND/OR graph). For instance, the sequence  $\{\{3,5\},\{2,4\},1\}$  of the final design in Figure 17(a) corresponds to the set of nodes,  $\{a1, b1, e1, e2\}$  in Fig. 16.

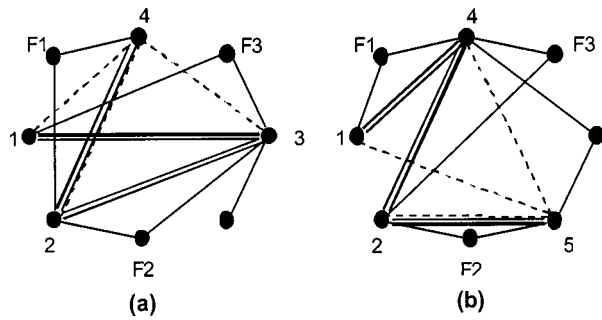
## Discussion

**Nonstrict Cases.** Figure 18(a) shows a decomposition that does not satisfy the decomposition rules for dimensional adjustability, although joints provide adjustability to achieve given KC in a slightly tilted angle. While member 1 can be adjusted along the direction of the given KC, this adjustment also brings about the movement of member 1 in the  $x$  direction. When  $n_1 \cdot k, n_2 \cdot k \geq T_1$ , where  $0 \leq T_1 < 1$ , we can say those joints can provide adjustability for the KC with negligible variation in other direction. Likewise, Fig. 18(b) shows a decomposition that does not satisfy the decomposition rule for nonforced fit, but it is still better than those with parallel joints. In this case,  $n_1 \cdot n_2 \leq T_2$ , where  $0 < T_2 \leq 1$ .

We will characterize these decompositions as nonstrict cases in contrast to the strict case we have considered so far. Since the nonstrict case involves tolerance issues and increases the size of problem significantly, we have not considered such nonstrict adjustability at this point and shall leave it as one of the future works.

**Relation to Datum Flow Chain (DFC).** In the DFC [1] of an assembly, every node represents a part or fixture and every directed edge defines a joint called "mate" which transfers a dimensional constraint from the part at the source node to the part at the target node. Every joint that does not transfer any dimensional constraint is called 'contact' which is represented as a dashed edge. Each KC is represented as a double edge. In our assembly synthesis method, because it is assumed that every KC is achieved through a slip plane with a fixture, there is no base part (a part without any incoming edge) other than fixtures if the assembly has at least one KC. Every slip plane used to achieve a KC becomes a contact, as they are just joined once parts are located in the fixture. The joints involved in an assembly operation not delivering a KC become mates because the joints themselves fix the dimension.

In the assembly synthesis method presented, whenever a KC is broken by decomposition, a parallel slip plane has been assigned with the assumption of existence of fixture which decides relative location of parts connected by the KC. Every distinctive assembly design from the AND/OR graph, thus, defines a unique DFC where every part connected by a KC are located by a corresponding fixture. Furthermore, there are several trees (assembly se-



**Fig. 19 (a) The DFC of the final design in Fig. 17(a) and (b) the DFC of the final design in Fig. 17(e).**

quences) in the AND/OR graph for each final design, and each of these trees represents a desirable assembly sequence to achieve the DFC that its final design defines. As BUILD\_AO generate all feasible assembly designs with accompanying assembly sequences for in-process dimensional adjustability and proper constraint, it can be concluded that BUILD\_AO presents all feasible DFCs with compatible assembly sequences. Figures 19(a) and (b) show DFCs defined from the final designs in Figs. 17(a) and 17(e) respectively. As described above, assembly sequences listed in Figs. 17(a) and (e) can be used to achieve these DFCs satisfying in-process adjustability and proper constraint criteria. Note that, in each DFC, there are three fixtures and three contacts for three corresponding KCs. Part 5 in Fig. 19(a) and part 3 in Fig. 19(b) are the only parts in each DFC that have mates coming from a nonfixture part, as they are not related to delivering any KC.

### Summary and Future Work

This paper presented a method of assembly synthesis focused on the in-process dimensional adjustability. The method recursively decomposed product geometry and assigned joint configurations based on the simple rules in order to achieve dimensional adjustability and nonforced fit. The rules employed in the decomposition process were drawn from a few theories of assembly design. The AND/OR graph of assembly plans was augmented to represent assembly synthesis with matching assembly sequences. The algorithm generating the AND/OR graph of assembly synthesis was given. The method was applied to two dimensional skeletons of product designs at very early stage of the design process. It was also shown that each final design from the assembly synthesis defines its own DFC.

Among others, applying the method to three dimensional products would be the first step in future works. Because typical three dimensional product such as automotive bodies have beams and panels with curved plane, it would be impractical to apply strict rules discussed in this paper. Instead, we would have to develop relieved rules for nonstrict case to successfully implement the method to three dimensional products. In order to make the method more practical, we have to deal with cases in which a number of KCs are closely related that they can not be achieved independently. Also it is possible that a certain part of the product

cannot be decomposed without breaking one of the rules guiding decomposition process. In these cases, we should either transfer the KC to the part level by abandoning the assumption that all KCs are delivered by assembly operations on fixtures, or carefully prioritize rules case by case. Building cost evaluation of having a certain number of components and a certain number of assembly operations will also enrich the method. Based on the evaluation scheme, we could search the AND/OR tree to find the optimal assembly synthesis.

### Acknowledgments

This work was supported by the National Science Foundation with a CAREER Award (DMI-9984606). A matchable fund for this grant has been provided by Toyota Motor Company. These sources of support are gratefully acknowledged. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

### References

- [1] Mantripragada, R., and Whitney, D. E., 1998, "The Datum Flow Chain," *Res. Eng. Des.*, **10**, pp. 150–165.
- [2] Yetis, F. A., and Saitou, K., 2000, "Decomposition-based Assembly Synthesis Based on Structural Considerations," *2000 ASME Design Engineering Technical Conferences*, Baltimore, Maryland, Paper no. DETC2000/DAC-14228.
- [3] Bourjault, A., 1984, "Contribution a une Approche Methodologique de L'assemblage Automate: Elaboration Automatique des Sequences Operatoires," Ph.D. Thesis, Universite de Franche-Comte.
- [4] De Fazio, T. L., and Whitney, D. E., 1987, "Simplified Generation of All Mechanical Assembly Sequences," *IEEE J. Rob. Auto.*, **RA-3**(6), pp. 640–658.
- [5] Lee, D. J., and Thornton, A. C., 1996, "The Identification and Use of Key Characteristics in the Product Development Process," *1996 ASME Design Engineering Technical Conference*, Irvine, California, Paper no. 96-DETC/DTM-1506.
- [6] Homem de Mello, L. S., and Sanderson, A. C., 1990, "AND/OR Graph Representation of Assembly Plans," *IEEE Trans. Rob. Autom.*, **6**(2), pp. 188–199.
- [7] Homem de Mello, L. S., and Sanderson, A. C., 1991, "A Correct and Complete Algorithm for the Generation of Mechanical Assembly Sequences," *IEEE Trans. Rob. Autom.*, **7**(2), pp. 228–240.
- [8] Homem de Mello, L. S., and Sanderson, A. C., 1991, "Representations of Mechanical Assembly Sequences," *IEEE Trans. Rob. Autom.*, **7**(2), pp. 211–227.
- [9] Wang, C.-H., 1997, "Manufacturability-driven Decomposition of Sheet Metal Products," Ph.D. Thesis, Carnegie Mellon University.
- [10] Wang, C.-H., and Bourne, D., 1997, "Concurrent Decomposition for Sheet-metal Products," *1997 ASME Design Engineering Technical Conferences*, Sacramento, California, Paper no. DETC97/DFM-4328.
- [11] Liu, S. C., and Hu, S. J., 1998, "Sheet Metal Joint Configurations and Their Variation Characteristics," *ASME J. Manuf. Sci. Eng.*, **120**(2), pp. 461–467.
- [12] Ceglarek, D., and Shi, J., 1998, "Design Evaluation of Sheet Metal Joints for Dimensional Integrity," *ASME J. Manuf. Sci. Eng.*, **120**(2), pp. 452–460.
- [13] Foulds, L. R., 1991, *Graph Theory Applications*, Springer-Verlag, New York, New York.
- [14] Rosen, K., 1999, *Discrete Mathematics and Its Applications*, Fourth edition, McGraw Hill.
- [15] Suh, H., and Chang, C. K., 2000, "Algorithms for the Minimal Cutsets Enumeration of Networks by Graph Search and Branch Addition," *Proceedings 25th Annual IEEE Conference on Local Computer Networks*, Tampa, Florida, pp. 100–107.
- [16] Whitney, D. E., Mantripragada, R., Adams, J. D., and Rhee, S. J., 1999, "Designing Assemblies," *Res. Eng. Des.*, **11**, pp. 229–253.
- [17] Lee, S., 1991, "Backward Assembly Planning," *Proceedings of the 1991 IEEE International Conference on Tools for AI*, San Jose, California, pp. 2382–2391.