

Integrated Synthesis of Assembly and Fixture Scheme for Properly Constrained Assembly

Byungwoo Lee and Kazuhiro Saitou, *Member, IEEE*

Abstract—This paper presents an integrated approach to design an assembly, fixture schemes, and an assembly sequence, such that the dimensional integrity of the assembly is insensitive to the dimensional variations of individual parts. The adjustability of critical dimensions and the proper constraining of parts during assembly process are the keys in achieving the dimensional integrity of the final assembly. A top-down design method is developed which recursively decomposes a lump of initial product geometry and fixture elements matching critical dimensions into parts and fixtures. At each recursion, joints are assigned to the interfaces between two subassemblies to ensure that parts and fixtures are properly constrained at every assembly step. A case study on a simple frame structure is presented to demonstrate the method.

Note to Practitioners—Achieving dimensional integrity of complex assemblies is a very demanding task due to the dimensional variations of parts and their propagation. Keys to achieving the goal are to adjust critical dimensions using proper fixtures and joint configurations and to minimize unpredictable deformations caused by overconstrained parts, such that assembled products can be robust to dimensional variations of parts. Equally important are arranging fixtures and sequencing assembly steps as these are what really define which critical dimension is adjusted and which subassemblies are put together on what fixture at each assembly step. In this paper, we propose a top-down decomposition-based approach that generates assembly design, assembly sequence, and fixture plan ensuring adjustability for critical dimensions and properly constrained subassemblies at every assembly step. The method does not only prevent costly trial and error following bottom-up approaches, but also enumerates all feasible solutions for a given product geometry and critical dimensions. As the number of solutions can be fairly large, for real-world application, it is crucial to incorporate practical constraints and to identify other important criteria to search optimal designs.

Index Terms—Assembly synthesis, design for manufacture, dimensional integrity, proper constraint.

I. INTRODUCTION

STRUCTURAL enclosures of modern mechanical products, such as ship hulls, airplanes, and automotive bodies, typically are made of hundreds or thousands of parts due to

Manuscript received June 13, 2004; revised November 17, 2004. This paper was recommended for publication by Associate Editor H. Ding and Editor M. Wang upon evaluation of the reviewers' comments. This work was supported in part by the National Science Foundation under CAREER Award (DMI-9984606) and by the Toyota Motor Corporation. This paper was presented in part at the ASME 2004 International Design Engineering Technical Conferences, Salt Lake City, UT, September 28–October 2, 2004.

B. Lee was with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109-2125 USA. He is now with GE Global Research, Niskayuna, NY 12309 USA (e-mail: byungwoo.lee@gmail.com).

K. Saitou is with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109-2125 USA (e-mail: kazu@umich.edu).

Digital Object Identifier 10.1109/TASE.2005.846288

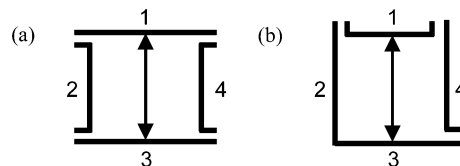


Fig. 1. Two box designs: (a) without and (b) with adjustable height during assembly [1].

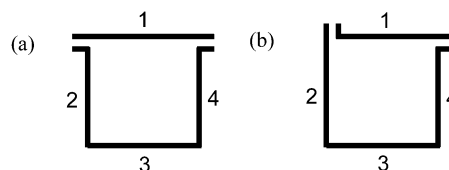


Fig. 2. Two box designs: (a) without and (b) with properly constrained parts [1].

their geometric complexity and sizes. As the number of parts increases, however, achieving the dimensional integrity of the final assembly becomes more difficult due to the inherent variations in manufacturing and assembly processes.

A solution is to adjust critical dimensions *in assembly processes* when parts or subassemblies are located and fully constrained in fixtures. This in-process dimensional adjustment is typically facilitated by slip planes, mating surfaces at joints that allow a small amount of relative motions. For example, Fig. 1 shows two designs of a rectangular box. In contrast to design in (a) with no in-process adjustability of the critical dimensions (length between sections 1 and 3), the design in (b) provides slip planes such that the relative location of parts can be adjusted along the critical dimension.

The dimensional integrity of an assembly is also affected by the postassembly distortion due to the internal stress induced by joining parts with dimensional mismatches. A solution is to ensure the proper constraining of subassemblies at each assembly step. For example, part 1 in Fig. 2(a) is not properly constrained and therefore the postassembly distortion might occur, if the length of sections 2 and 4 are slightly different due to manufacturing variation. With two slip planes perpendicular to each other, the design in (b) can absorb manufacturing variations within parts 1 and 2–4, provided that variations in angles are negligible.

In addition to the assembly design including joint types at part interfaces, the assembly sequence also influences in-process dimensional adjustability and proper part constraints. In the assembly sequence in Fig. 3(a), the critical dimension (total length) is not adjustable since there is no slip plane parallel to it when the total length is realized with the addition of part

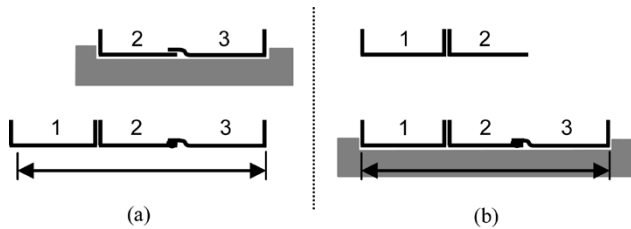


Fig. 3. Assembly sequences (a) without and (b) with in-process adjustability (modified from [2]).

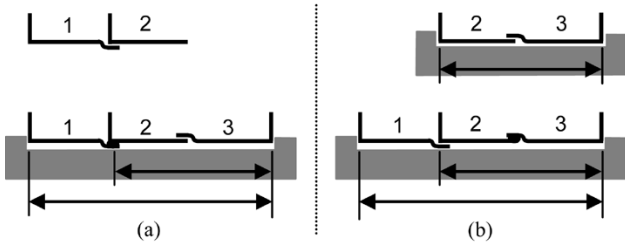


Fig. 4. Assembly sequences where two dimensions are adjusted (a) at one step and (b) independently at two steps (modified from [2]).

1. On the other hand, the sequence shown in (b) provides the slip plane at the assembly step where the critical dimension is achieved, to absorb the variation in length. As another example, the sequence in Fig. 4(b), where each critical dimension is independently adjusted at each step, is more desirable than the sequence in (a), where both dimensions are adjusted at one step, inevitably requiring a compromise between two potentially conflicting critical dimensions. Fig. 5 illustrates an effect of the assembly sequence on proper part constraints, where the sequence in (a) causes overconstraint at the second step, whereas all parts are properly constrained at all steps in (b), thus avoiding potential assembly stress.

Let us note Fig. 4 again. In Fig. 4(b), each critical dimension is realized on a separate fixture, in which case it is the only a feasible assembly sequence to realize both critical dimensions independently. However, to make the problem more complicated, other assembly sequences are feasible if fixtures are arranged differently. For example, in Fig. 6, both critical dimensions are realized independently on the only fixture, in two different assembly sequences, (a) and (b). What is different from Fig. 4(a) is that pins locating parts 1 and 2 control the location of parts 1 and 2 separately, thus enabling independent realization of the critical dimensions. The pin locating part 3 serves to realize both critical dimensions. Indeed, the fixture in Fig. 6 is the union of the two fixtures utilized in Fig. 4(b). Examining different ways of arranging fixtures for multiple critical dimensions is valuable, as using one fixture to deal with several critical dimensions is quite common for large-scale assemblies, especially when several parts constitute a flat subassembly.

As pointed out by industry practitioners and researchers, having subassemblies adjustable for critical dimensions and properly constrained are key elements in assembly design to achieve high precision and accuracy with low-cost parts [3]. Whereas it is important to carefully design and sequence the

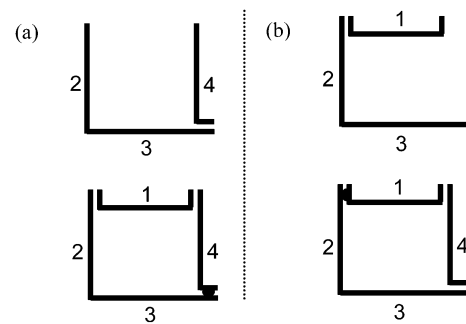


Fig. 5. Assembly sequences (a) without and (b) with proper constraints [1].

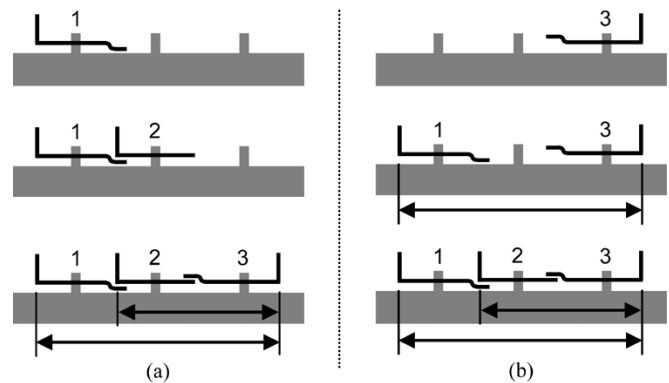


Fig. 6. Feasible assembly sequences depend on utilization of fixtures. Compare with Fig. 4(b).

assembly and fixtures in order to avoid overconstraints and the loss of desired adjustability, industry practices do not come up to systematic approaches. Despite the fact that adjustability and proper constraint should be ensured between “subassemblies” at “every assembly step,” not between parts, current design practices and CAD systems overlook this important property and mistreat joints and tolerances as the attributes of part geometry without considering assembly sequences. For complex mechanical assemblies, this causes many dimensional discrepancies at the manufacturing stage, followed by costly redesigns and reworks. To make matters worse, typical engineering countermeasures in such situations have often been to tighten part tolerances, without examining the assembly design and tolerance relationships as a whole [4].

As a remedy, we have presented a top-down decomposition-based assembly synthesis method [1] to fully enumerate all feasible sets of part decomposition, joint assignments and an assembly sequence, for two-dimensional (2-D) geometry. Assuming that assemblies can be built in the reverse sequence of decomposition, the method recursively decomposes a given product geometry into two subassemblies until parts become manufacturable. At each recursion, joints are assigned to the interfaces between two subassemblies to ensure in-process dimensional adjustability and proper constraint. The method has also been applied to three-dimensional (3-D) beam-based structure [5], where Screw Theory [6] is utilized for the evaluation of in-process adjustability and proper constraints of subassemblies at every assembly step.

However, our previous works [1], [5] are limited in exploring solutions on various *fixture schemes*,¹ by implicitly assuming one fixture to achieve every critical dimension [as shown in Fig. 4(b)]. This paper extends our previous works to design a fixture scheme as an integrated part of assembly synthesis, which enumerates all feasible “designs” (assembly designs, fixture schemes, and assembly sequences) by treating fixtures as an entity of assembly. Not only does this integration explore all feasible fixture schemes along with assembly designs, but also reveals feasible assembly sequences that were illicit in our previous methods [1], [5], such as those shown in Fig. 6. A case study on a simple space frame is presented to demonstrate the method. Considering the number of parts, the number of fixtures, the depth of the assembly tree, and the number of under-constraints as objectives to minimize, a multi-objective graph search is performed on the enumerated feasible designs in order to obtain Pareto optimal solutions. Some representative designs in the Pareto set are examined to illustrate the tradeoffs among the assembly design, fixture scheme, and assembly sequence.

II. RELATED WORKS

Since previous works in general relevance to assembly synthesis are reviewed in [1]; this section focuses on the literature directly related to the present extension of the assembly synthesis method, namely on properly constrained assembly designs and fixture designs.

The advantages of properly constrained assemblies are well known to practitioners in precision machinery design, and several methods have been proposed in the literature, including kinematic design [7], minimum constraint design [8], and exact constraint design [3], [4]. These works describe disadvantages of overconstraints and provide good practices as well as analytical methods to compute constraints. In these works, the most commonly cited merit of proper constraint design is repeatability that leads to high precision. Downey *et al.* [9] analyzed and classified elements of assemblies that absorb manufacturing variations of parts.

A universal analytical method for motion and constraint analysis dates back to Screw Theory, a pioneering work by Ball [6]. Since then, Screw Theory has been applied to wide areas of mechanism, robotics, and machine design. Among others, Waldron [10] utilized the Screw Theory to build a general method that can determine all relative degrees of freedom (DOFs) between any two rigid bodies making contacts to each other. Ohwovoriole and Roth [11] extended Screw Theory by providing a theoretical study of repelling screw systems, which became an important basis in many automatic layout design methods of machining fixtures [12]–[14]. Asada and By [12] proposed a kinematic analysis method for fixture layout design by modeling kinematic constraints of fixture locators as a Jacobian matrix, which should have full rank to locate a given work piece uniquely at a desired position. Blanding [4] showed the application of Screw Theory to assembly design. Adams and Whitney [15] also used Screw Theory to compute

¹A fixture scheme is defined as a plan showing which fixture will control what critical dimensions where in assembly sequence. More formal definition will follow in terminology section.

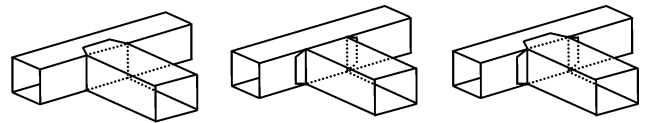


Fig. 7. An example of joint library for 3-D beam-based assemblies consisting of lap, butt, and lap-but.

the constraints on parts and applied it to rigid body assemblies with mating features such as pin-slot joint.

Whereas these works provide tools for analyzing constraints in a given assembly and design guidelines, they do not address a systematic and integrated synthesis of an assembly and fixture scheme with desired constraint characteristics such as in-process dimensional adjustability and proper part constraints, as discussed in this paper. Although the design of the fixture scheme should precede physical fixture layout design, authors could not find previous works attacking this problem in a systematic way.

III. TERMINOLOGY

Since the assembly synthesis deals with objects yet to be decomposed into an assembly of separate parts, a few terms and concepts need to be defined to avoid confusion with generic meanings used in other literatures.

- A *product geometry* is a geometric representation of a whole product as one piece before decomposition into parts.
- A *member* is a section of a product geometry allowed to be a separate part. A pair of members is *connected* when they meet at a certain point in the product geometry.
- A *configuration* is a group of members which are connected. A product geometry is a configuration, and so is a part (as defined below).
- The *Key Characteristics* (KCs) are defined by Lee and Thornton [16] as product features, manufacturing process parameters, and assembly features that significantly affect a product’s performance, function and form. In this paper, a KC refers to a critical dimension to be achieved in assemblies.
- A *decomposition* is a transition of a configuration into two subconfigurations by removing connections.
- A *part* is a configuration that is not decomposed further under given criteria, e.g., a minimum part size. A part may consist of one or more members.
- A *joint library* is a set of joint types available for a specific application domain (Fig. 7).
- A (*synthesized*) *assembly* is a set of parts and joints that connect every part in the set to at least one of other parts in the set.
- *Assembly synthesis* is a transformation of a product geometry into an assembly.
- A *fixture element* is an imaginary part of a fixture to control a KC. Physically, a KC will be controlled by a set of locators, and the fixture element is abstract representation of this set of locators. Thus, each KC will have a fixture element corresponding to it. A *fixture* is a group of fixture elements and controls corresponding KCs.

- *Fixture scheme* is partitioning the whole set of fixture elements into groups and assigning them into assembly sequence.

IV. SCREW THEORY

In Screw Theory,² a *screw* is defined as a pair of a straight line (*screw axis*) in a 3-D Cartesian space and a scalar (*pitch*). It is commonly represented by *screw coordinates*, a pair of two row vectors $\mathcal{S} = (\mathbf{s}; \mathbf{s}_0)$ in 3-D Cartesian coordinates, where \mathbf{s} is a unit vector parallel to the screw axis and \mathbf{s}_0 is given as

$$\mathbf{s}_0 = \mathbf{r} \times \mathbf{s} + p\mathbf{s} \quad (1)$$

where \mathbf{r} is the position vector of a point on the screw axis and p is the pitch. By taking dot product with \mathbf{s} , p can be expressed as

$$p = \frac{\mathbf{s} \cdot \mathbf{s}_0}{\mathbf{s} \cdot \mathbf{s}}. \quad (2)$$

A screw with an infinite pitch does not follow (1), and therefore it is denoted by \mathbf{s} being the zero vector and \mathbf{s}_0 being the unit vector parallel to the screw axis.

Two types of screws, twist and wrench, are utilized in this paper. A *twist* is a screw representing a motion of a rigid body simultaneously rotating around and translating along an axis. Using screw coordinates, it is denoted as $\mathcal{T} = (\boldsymbol{\omega}, \mathbf{v})$, where $\boldsymbol{\omega}$ is the angular velocity and \mathbf{v} is the linear velocity of a point on the body (or its extension) located at the origin of global reference frame. A *wrench* is a screw representing a force along and a moment around an axis exerted on a rigid body. Using screw coordinates, it is denoted as $\mathcal{W} = (\mathbf{f}; \mathbf{m})$, where \mathbf{f} is the force and \mathbf{m} is the moment that a point on the body (or its extension) located at the origin of global reference frame should resist.

Two screws $\mathcal{S}_1 = (\mathbf{s}_1; \mathbf{s}_{01})$ and $\mathcal{S}_2 = (\mathbf{s}_2; \mathbf{s}_{02})$ are *reciprocal* to each other, if and only if they satisfy

$$\mathbf{s}_1 \cdot \mathbf{s}_{02} + \mathbf{s}_{01} \cdot \mathbf{s}_2 = 0. \quad (3)$$

If a twist \mathcal{T} is a reciprocal of a wrench \mathcal{W} (or vice versa), \mathcal{W} does no “work” to a rigid body moving according to \mathcal{T} .

When a body can receive linear combinations of several screws (either twist or wrench), this set of screws are typically represented as a matrix where each screw in the set forms a row vector of the matrix. This matrix is called a *screw matrix*. As its row space is the screw space (a space formed by the set of screws in the matrix), the rank of a screw matrix is equal to the dimension of the screw space.

The function reciprocal (\mathbf{S}) returns a screw matrix whose row consists of the screws reciprocal to those in \mathbf{S} . It can be obtained by exchanging the former three columns and the latter three columns of the null space of \mathbf{S} .

The *union* of screw matrices represents the sum of the screw spaces defined by the matrices and can be obtained by simply “stacking” them on top of one another

$$\bigcup_{i=1}^n \mathbf{S}_i \equiv \begin{pmatrix} \mathbf{S}_1 \\ \mathbf{S}_2 \\ \dots \\ \mathbf{S}_n \end{pmatrix}. \quad (4)$$

²The terminology and formalization in this section are summarized from [6], [15], [17]–[19].

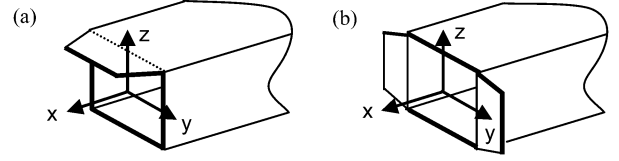


Fig. 8. (a) Lap and (b) lap-butt joint of a beam-based model and the local coordinate frames for twists.

The *intersection* of screw matrices is the set of screws common to the screw matrices and can be computed through double reciprocals

$$\bigcap_{i=1}^n \mathbf{S}_i \equiv \text{reciprocal} \left(\bigcup_{i=1}^n \text{reciprocal}(\mathbf{S}_i) \right). \quad (5)$$

Since a twist and a wrench are also screws, the definitions of reciprocal, union, and intersection hold.

Woo and Freudenstein [17] studied kinematic properties of various joint types in screw coordinates, which are adopted to build twist matrices of beam joint types. Fig. 8(a) shows a typical lap joint found in beam-based structures. When it is attached to another beam, the tab allows planar motion parallel to the x - y plane. Also, if we assume that the length of the tab is very small compared to the length of the beam, it can be treated as a line contact along y axis, allowing the rotation about the y axis. Thus, the lap joint, with respect to the local coordinate frame shown in the figure, can be modeled as a twist matrix

$$\mathbf{T}_{\text{lap}} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \quad (6)$$

Similarly, a butt joint in Fig. 8(b) allows the motion parallel to the y - z plane to be modeled as

$$\mathbf{T}_{\text{butt}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (7)$$

In twist matrices in (6) and (7), each row represents an independent motion, and each nonzero number represents rotation or translation along a corresponding axis— ω_x , ω_y , ω_z , v_x , v_y , or v_z . For example, the first row in (6) has 1 at the second column, which means the lap joint allows rotational motion about y axis. In the third row, it has 1 at the fourth column, meaning translation along the x axis is allowed. Since these matrices are used only to give information on which DOFs are not constrained for a joint type, the magnitude of each twist (row) of these twist matrices (i.e., the magnitudes of the angular and linear velocities in the twist) is not significant in this paper.

Once the twist matrix is obtained for a joint type, the reciprocal wrench matrix can be computed as described above. For instance, the wrench matrices corresponding to twist matrices in (6) and (7) are

$$\mathbf{W}_{\text{lap}} = \text{reciprocal}(\mathbf{T}_{\text{lap}}) = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad (8)$$

$$\mathbf{W}_{\text{butt}} = \text{reciprocal}(\mathbf{T}_{\text{butt}})$$

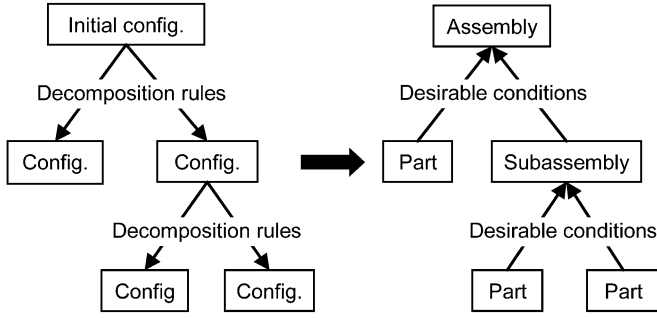


Fig. 9. Assembly synthesis by top-down hierarchical decomposition. Assembly sequence is the reverse of the decomposition sequence.

$$= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (9)$$

Each nonzero number now represents force or moment along a corresponding axis— f_x , f_y , f_z , m_x , m_y , or m_z . Since a wrench that is a reciprocal of a twist does no “work” to a rigid body moving according to the twist, these are the forces and moments the joint supports (hence resulting in no work). For example, the first row in (8) has 1 at the third column, which means the lap joint can support a force along the z axis.

V. ASSEMBLY AND FIXTURE SCHEME SYNTHESIS

A. Assembly Synthesis via Recursive Decomposition

There are numerous issues related to assembly design. Among others, adjustability and proper constraint are the key conditions for dimensional integrity. Dissimilar to other issues such as structural stiffness and product function, these two conditions should be satisfied *at every assembly step*, as illustrated in Figs. 1–5. By taking advantage of this fact, one can hierarchically decompose a given product geometry such that (sub)geometries at each decomposition step satisfy the above desired conditions when they are assembled back together in the reverse order (see Fig. 9). Our previous works [1], [5] suggested the framework of assembly synthesis via such hierarchical decomposition, which was successfully applied to simple 2-D [1] and 3-D [5] geometries.

B. Generation of Fixture Elements

A KC, in this paper, is assumed to be a critical dimension between parts to be achieved by the adjustment during the assembly process. Thus, the dimension noted as a KC will be constrained by a fixture, according to which parts being assembled will be located. In this context, we know the fixture would have to constrain *at least* the DOFs specified by the KC, regardless of its physical embodiment. Provided a KC is controlled by a fixture, the assembly of two subassemblies connected by a KC can be viewed as two assembly steps, involving two subassemblies and a fixture, such as $\{\{\text{part1, fixture}\}, \text{part2}\}$. As depicted in Fig. 10, this allows each KC to be replaced by a fixture element connecting the same members. The graph representation shown

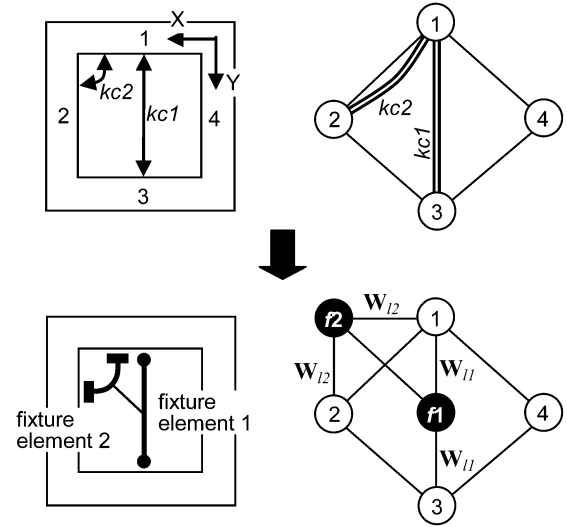


Fig. 10. Replacement of KCs with fixture elements whose locators constrain the same DOFs.

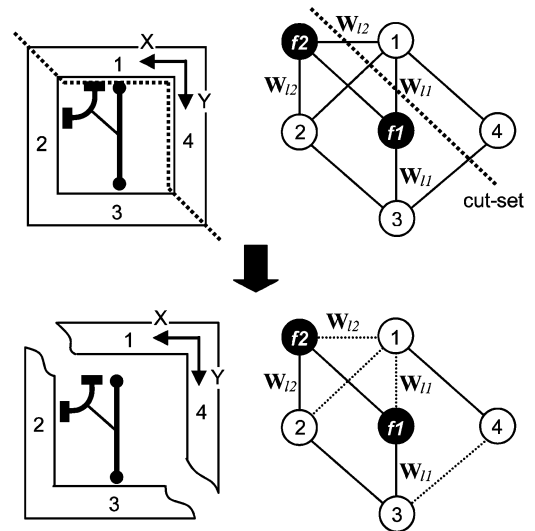


Fig. 11. A feasible decomposition.

in Fig. 11 is what we call *configuration graph*. After replacing KCs with fixture elements, the configuration graph is a pair

$$C = (M, E) \quad (10)$$

where M is the set of nodes representing members and fixture elements, and E is the set of edges representing connections. Each node in M is associated with its type (members are in white and fixture element are in black in Fig. 10), and each edge in E between a member node and a fixture element node is associated with a wrench matrix representing the DOFs to be constrained by the replaced KC. For example, if $kc1$ in Fig. 10 is the distance between members 1 and 3 in the y direction measured at $x = 1.5$ in the global reference frame, then the wrench matrix associated with edges $\{1, f1\}$ and $\{3, f1\}$ is

$$W_{11} = (0 \ 1 \ 0 \ 0 \ 0 \ 1.5) \quad (11)$$

where subscript l indicates “locator.” Similarly, the wrench matrix associated with for edges $\{1, f2\}$ and $\{2, f2\}$ are

$$\mathbf{W}_{l2} = (0 \ 0 \ 0 \ 0 \ 0 \ 1). \quad (12)$$

Although seemingly subtle, this replacement of KCs with the fixture elements enables an elegant integration of fixture scheme synthesis into the assembly synthesis process. Initially, each fixture element is connected to all of the other fixture elements, in order to allow the exploration of all possible fixture schemes. The connections between a fixture element and a member represent *minimum locators* that constrain at least the DOFs specified by the replaced KC. Any additional DOFs needed to uniquely locate the part will be computed during assembly and fixture scheme synthesis as described in Sections V-C-F. Further, the configurations after the replacement of KCs with fixture elements are classified to three classes:

- *Incomplete configuration*: configuration with disconnected members (if without fixture elements) or with a fixture element connected to less than two members. For example, the second step of Fig. 6(b) is an incomplete configuration since, members are not connected and the fixture element controlling the distance between members 2 and 3 has only one connection (to part 3) due to the absence of member 2.
- *Fixture*: configuration consisting of only fixture elements.
- *Complete configuration*: configuration that is neither an incomplete configuration nor a fixture.

C. Feasible Binary Decomposition

The assembly synthesis algorithm [1] adopted in this paper assumes that every assembly step combines a pair of subassemblies. Conversely, the algorithm decomposes a configuration into two (sub)configurations by removing some connections, which is equivalent of finding a *cut-set* [20] of the configuration graph. Decomposition is made only when the reverse of it yields a feasible assembly step, for which there are two criteria. First, the assembly step is binary—only two subassemblies are joined at the assembly step. This is justified by the fact that a nonbinary assembly step (e.g., assembly of multiple parts on a fixture in one step) can be broken down to an equivalent sequence of binary assemblies. Second, at least one of two subassemblies joined at the assembly step is a complete configuration, which is justified in the next paragraph.

When a configuration is incomplete, subassemblies should remain on the fixture because subassemblies are either not connected or the fixture has at least one assigned KC yet to realize [such as the status shown in Fig. 6(b)]. Since fixtures are usually heavy or grounded, it would be very rare that a subassembly attached to a fixture is assembled to another subassembly in the same situation or to another fixture. For the same reason, assembly of two fixtures is considered infeasible. On the other hand, when a configuration is complete, it has only one connected subassembly and, if any, a fixture with all assigned KCs realized. Therefore, it is ready to leave the fixture for further assembly with any configuration including a fixture.

For example, Fig. 11 shows a feasible decomposition yielding one complete and one incomplete subconfiguration. In Fig. 12,

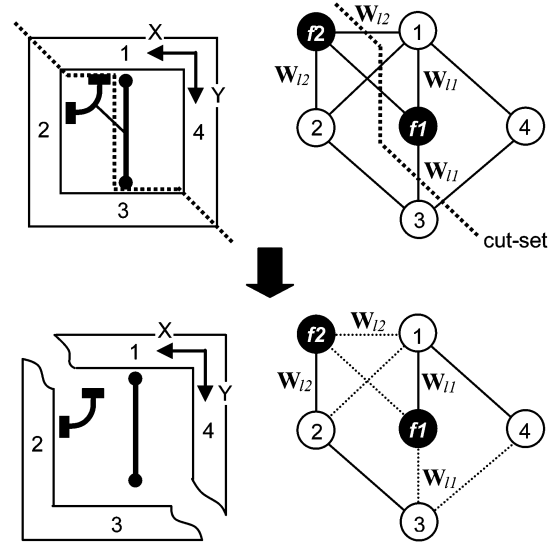


Fig. 12. Infeasible decomposition that results in two incomplete subconfigurations.

both subconfigurations are incomplete and thus will not be considered to be a feasible decomposition.

More formally, decomposition from configuration $C_a = (M_a, E_a)$ to two subconfigurations $C_b = (M_b, E_b)$ and $C_c = (M_c, E_c)$ is feasible if the following conditions are satisfied:

- $M_b \neq \emptyset$ and $M_c \neq \emptyset$.
 - C_b and C_c are connected.
 - At least one of C_b and C_c is a complete configuration.
 - $M_a = M_b \cup M_c$.
 - $M_b \cap M_c = \emptyset$.
- (13)

The first condition states that subconfigurations should be nonempty. The second condition states that the subconfigurations must be connected. The fourth and fifth conditions state that the configuration must split into a pair of disjoint subconfigurations.

D. Decomposition Rule for Dimensional Integrity

Once a decomposition satisfying conditions in (13) is found, feasible joint types are assigned to broken connections, which is represented as mapping $\gamma_d : CS_d \mapsto JL$, where CS_d is the cut-set broken by decomposition d and JL is a library of joint and locator types. With the joint assignment, decomposition d can be uniquely specified as $d = (M_a, \gamma_d, (M_b, M_c))$. See Fig. 13 for an example. Note that feasible joint types may depend on the local geometry near the joint location. For example, feasible joint types between two perpendicular beams would be different from those for two coaxial beams. The broken connections with the assigned joints are associated with the wrench matrices computed according to the assigned joint types and orientations. Every connection between a member and a fixture element already has a wrench matrix computed in the previous step.

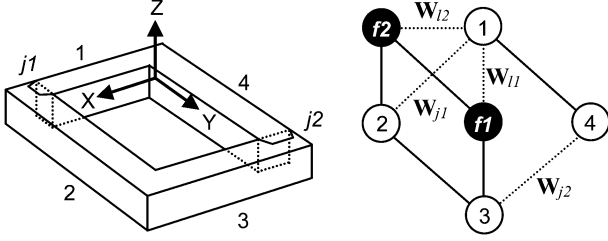


Fig. 13. Joints assigned to broken connections, for which wrench matrices are computed accordingly (fixture elements omitted in the left figure).

Having replaced all KCs with the corresponding fixture elements, the only criterion to consider for assigning joint types to broken connections is the proper constraint of the mixture of subassemblies and fixtures at every assembly step. In particular, there is no need to explicitly consider the adjustability for KCs as required in our previous work [1] since the proper constraint including the DOF constrained by KCs implies the assigned joints do not interfere with the DOFs constrained by KCs, automatically ensuring their individual adjustability.

In order for subassemblies being assembled to be properly constrained, the joints should not constrain the same DOF more than once, as illustrated in Fig. 5. This assembly rule is inversely stated as the decomposition rule for proper constraint in our previous work [1], which only allows the combination of joints yielding no overconstraint. Although, when one subassembly is located on an empty fixture, the fixture should constrain all six DOFs [12], and when the next subassembly is put together contacting the other subassembly on the fixture, it would be constrained by both the other subassembly and the fixture. Therefore, joints must be selected in such a way that no DOF is constrained twice not only among joints but also with locators. In such cases, the intersection of the wrench matrix corresponding to any subset of CS_d and the wrench matrix of any other disjoint subset must result in the zero matrix

$$\forall C_1, C_2 \subseteq CS_d, \quad C_1 \cap C_2 = \emptyset, \quad \left(\bigcup_{e \in C_1} \mathbf{W}_{\gamma_d(e)} \right) \cap \left(\bigcup_{e \in C_2} \mathbf{W}_{\gamma_d(e)} \right) = \mathbf{O} \quad (14)$$

which is also equivalent to

$$\text{rank} \left(\bigcup_{e \in CS_d} \mathbf{W}_{\gamma_d(e)} \right) = \sum_{e \in CS_d} \text{rank}(\mathbf{W}_{\gamma_d(e)}). \quad (15)$$

Further, in order to have all six DOFs constrained,

$$\text{rank} \left(\bigcup_{e \in CS_d} \mathbf{W}_{\gamma_d(e)} \right) = \sum_{e \in CS_d} \text{rank}(\mathbf{W}_{\gamma_d(e)}) = 6. \quad (16)$$

When any set of joint types and fixture elements satisfies (15) with the total rank less than six, it is considered to be feasible, assuming that additional fixtures or locators on existing fixtures will be arranged. Six less the number of DOFs constrained is counted as the number of underconstraints for each feasible joint assignment and recorded as $uc(\gamma_d)$. A predicate of a decomposition $d = (M_a, \gamma_d, (M_b, M_c))$ for the rule is given as

$$\text{de} : 2^{M_0} \times (2^{E_0} \mapsto JL) \times (2^{M_0} \times 2^{M_0}) \mapsto \{\text{true}, \text{false}\} \quad (17)$$

where $\text{de}(M_a, \gamma_d, (M_b, M_c))$ is *true* if and only if the conditions in (13) and (15) are satisfied.

There is an important exception to (15), for which compensation should be made before it is checked against (15); when there are connections in CS_d , from multiple fixture element to one member, the wrench matrices associated with the connections should be combined as a union such that the intersection among them could be ignored. This is based on a basic assumption that there would be no overconstraint between a fixture and a member. Suppose there is a set of fixture elements that are connected to a member and more than one of these connections are broken by a decomposition. In this case, even if there is a DOF constrained by more than one fixture elements, the DOF will be constrained by one locator in actual implementation. For example, see the first step in Fig. 6(b). When part 3 is placed on the fixture, the fixture is constraining two KCs of the same DOF, the distances to 1 and 2. However no one will use one locator for each KC, which will certainly yield overconstraint. When this step is generated through decomposition, the decomposition would break connections between the member 3 and each of the two fixture elements transformed from the two KCs. In order to match the assumption of one locator for a DOF, the wrench matrices for these connections should be combined as a union.

Consider the product geometry decomposed in Fig. 11 and the joint assignment shown in Fig. 13, which has two lap joints j_1 and j_2 , and two locators l_1 and l_2 , for edges cut by the decomposition. Because the decomposition is breaking multiple connections from member 1 to fixture elements, wrench matrices for these connections should be combined as a union. From (12) and (13), we can compute

$$\mathbf{W}_{l_1} \cup \mathbf{W}_{l_2} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1.5 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (18)$$

Suppose that the location of j_1 and j_2 in global reference frame X - Y - Z are $(3, 0, 0)$ and $(0, 4, 0)$. Then, based on the local coordinate frame of lap joint shown in Fig. 8 and orientation of j_1 and j_2 , \mathbf{W}_{lap} [in (8)] can be transformed into \mathbf{W}_{j_1} and \mathbf{W}_{j_2} in global reference frame

$$\mathbf{W}_{j_1} = \begin{pmatrix} 0 & 0 & 1 & 0 & -3 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \quad (19)$$

$$\mathbf{W}_{j_2} = \begin{pmatrix} 0 & 0 & 1 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \quad (20)$$

The union³ of the three matrices in (18)–(20) is then

$$\bigcup_{e \in CS_d} \mathbf{W}_{\gamma_d(e)} \sim \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (21)$$

whereas the summation of the ranks of individual matrices is six, and the rank of the union is only five, which implies that this combination of joints yield an overconstraint of one DOF. In fact, the intersection of \mathbf{W}_{j_1} and \mathbf{W}_{j_2} is not a zero matrix. As this joint assignment does not satisfy the decomposition rule (15), the assembly synthesis process will discard it.

³The matrix is in the Row Reduced Echelon Form for easy interpretation.

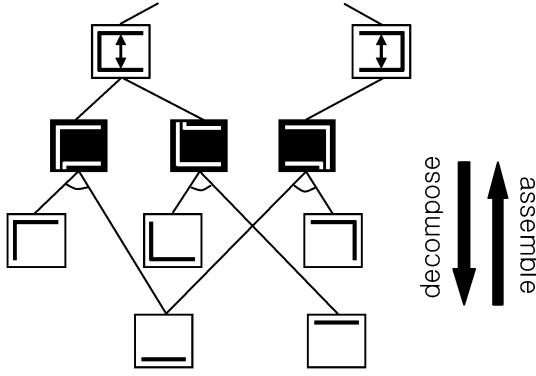


Fig. 14. Part of the AND/OR graph for the 2-D rectangular box in Fig. 1.

E. Part Manufacturability

The decomposition stops when the resulting subconfigurations become manufacturable by a chosen manufacturing process. In the following case study on frame structures, components are assumed to be extruded and bent. Therefore, a predicate of a configuration M_a for stopping decomposition is given as

$$\text{stop_de} : 2^{M_0} \mapsto \{\text{true}, \text{false}\} \quad (22)$$

where $\text{stop_de}(M_a)$ is true (i.e., decomposition continues) if and only if none of the first four conditions are satisfied or the fifth condition is satisfied.

- 1) M_a contains both member(s) and fixture member(s).
- 2) The induced subgraph on members in M_a has a closed loop (cannot extrude such parts).
- 3) Three or more members in M_a are connected to each other at a single point (cannot extrude such parts).
- 4) Members in M_a lie on more than one plane (difficult to handle/fixture).
- 5) M_a consists of only fixture elements (assembly of two fixtures is not considered).

See Fig. 11 for an example. The configuration $\{2, 3, f1, f2\}$ satisfies the first condition, thus stop_de returns *false*, subject to further decomposition. On the other hand, the other configuration $\{1, 2\}$ satisfies none of the first four conditions, the decomposition is stopped for this configuration.⁴

F. AND/OR Graph of Assembly Synthesis

A series of decompositions can be typically represented in a tree as shown in Fig. 9. However, the aim of the presented method is to enumerate all such trees, and an AND/OR graph [21] is adopted to facilitate the assembly synthesis, in which multiple trees share common nodes. Although the AND/OR has been previously used to enumerate assembly sequences for a given assembly design [22], it is augmented in this paper in order to embody joint assignments. Fig. 14 shows a partial AND/OR graph of assembly synthesis [1] for the 2-D rectangular box shown in Fig. 1. Each node in white background contains a configuration ($M_a \subseteq M_0$), and each node in black background contains joint assignment $\gamma_i CS_i \mapsto JL$. A set of three lines which connects a configuration M_a , joint assignment γ_i , and

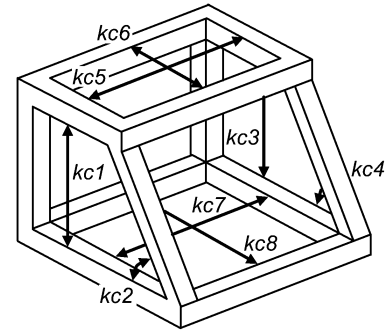


Fig. 15. Frame structure with eight KCs.

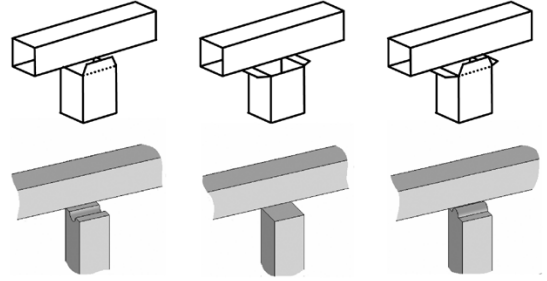


Fig. 16. Top: joint types for frame structure. Bottom: their graphical representation used in results.

TABLE I
NONDOMINATED COST VECTORS AND THE NUMBER OF CORRESPONDING NDSTs FOR THE FRAME STRUCTURE IN FIG. 15

Objectives				
No. of parts	No. of fixtures	Depth of tree	No. of under-constraints	No. of solution trees
7	2	7	6	16
8	4	5	24	24
8	1	8	3	46
6	3	6	11	4
Total				90

two subconfigurations (M_b, M_c) is a hyper-edge, represented as $(M_a, \gamma_i, (M_b, M_c))$ which is also the representation of a decomposition defined earlier. The AND/OR graph of assembly synthesis is then represented as a triple

$$AO = (S, J, F) \quad (23)$$

where S is a set of nodes representing configurations, J is a set of nodes representing joint assignments, and F is a set of hyper-edges $(M_a, \gamma_i, (M_b, M_c))$ satisfying the following necessary conditions:

- 1) $\text{stop_de}(M_a) = \text{false}$
- 2) $\text{de}(M_a, \gamma_i, (M_b, M_c)) = \text{true}$.

Then $AO = (S, J, F)$ is recursively defined as

- 1) If $\text{stop_de}(M_0) = \text{false}$, $M_0 \in S$.
- 2) For $\forall M_a \in S$, if $\exists \gamma_i, M_b, M_c$ such that $f = (M_a, \gamma_i, (M_b, M_c))$ satisfies necessary conditions (24), then $\gamma_i \in J, M_b, M_c \in S$ and $f \in F$.
- 3) No element is in S, J , and F , unless it can be obtained by using rules 1 and 2.

⁴Test runs were conducted on a 3.2-GHz Pentium 4 PC running Windows.

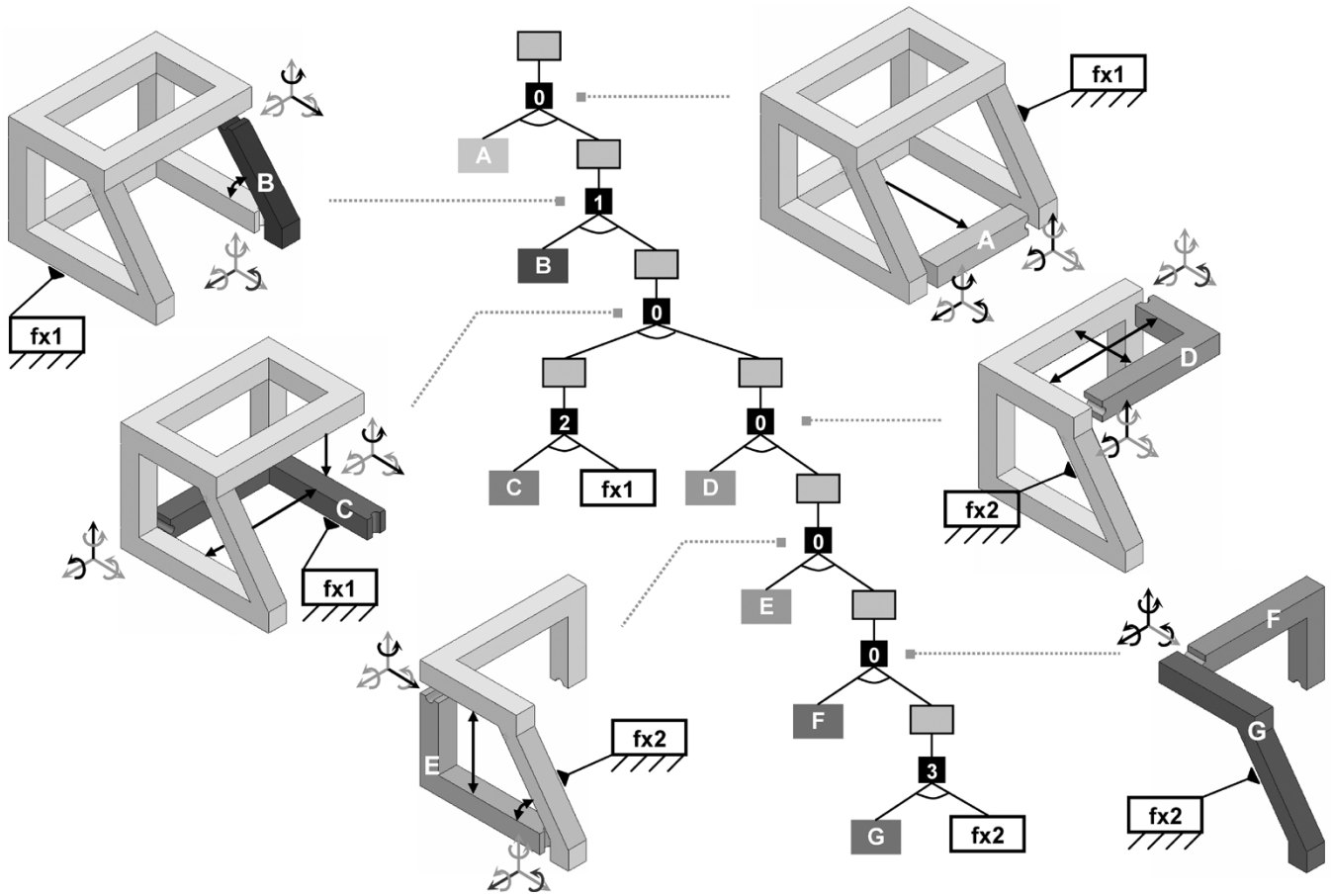


Fig. 17. One of 24 NDSTs whose cost vector is (7, 2, 7, 6).

The recursive definition in (25) can be easily transformed to an algorithm *build_AO* that generates AO from initial configuration and joint library by recursively decomposing a configuration into two subconfigurations [1], whose details are omitted due to space limitation. Using *stop_de* and *de* as defined earlier, one can run *build_AO* with any 3-D configurations to enumerate all possible assemblies (decompositions and joint assignments), fixture schemes, and accompanying assembly sequences that satisfy the in-process dimensional adjustability and proper part constraint.

VI. CASE STUDY

A frame structure in Fig. 15 is decomposed based on (25). Only joint types in Fig. 16 are assigned to broken connections as required by the decomposition rule. In order to reduce the size of the AND/OR graph, when several joint assignments satisfy the decomposition rule, one with minimum under-constraint is included in the AND/OR graph. Still, the decomposition rule produced a large AND/OR graph with 19 962 nodes representing configurations and 143 269 hyper-edges, which contains about 8.4 billion trees. While there would be many other potential objectives in designing assemblies, we have chosen the number of parts, the number of fixtures, the depth of tree (related to cycle time of assembly processes) and total underconstraints (the number of DOFs that fixtures should take care of, in addition to those that must be controlled for KCs) as objectives, which form a four-element cost vector. Using simple brute search

starting from terminal nodes (either part or fixture that satisfies *stop_de*), we have found only 90 trees are nondominated. Associated cost vectors for these nondominated solution trees (NDSTs) are listed in Table I.

The number of fixtures and underconstraints shows a strong correlation, because the more fixtures that are used, the more DOFs should be constrained when initially placing a part on each fixture. From Figs. 17–22, some of the NDSTs and their corresponding assembly designs are presented. In solution trees, a node with a capital letter represents a part (marked with the same letter in the following assembly design), and a node marked with “fx” with a number represents a fixture. A black node represents a joint assignment and the number within the node represents, $uc(\gamma_d)$, the number of underconstraints for the joint assignment.

Fig. 17 shows an NDST and corresponding assembly design and sequence, which has seven parts, two fixtures, the depth of 7, and six underconstraints. The coordinate frame shown by each joint shows DOFs constrained by the joint in black and unconstrained DOFs in gray. In the figure, fx1 controls kc_3 , kc_4 , kc_7 , and kc_8 , and fx2 controls kc_1 , kc_2 , kc_5 , and kc_6 . The assembly sequence is as follows.

- 1) Locate G on fx2. Three KCs related to G; kc_1 , kc_2 , and kc_5 are constrained by fx2. In order to uniquely locate G, fx2 should constrain the other three DOFs (the number of underconstraints) in addition to those required by the KCs.

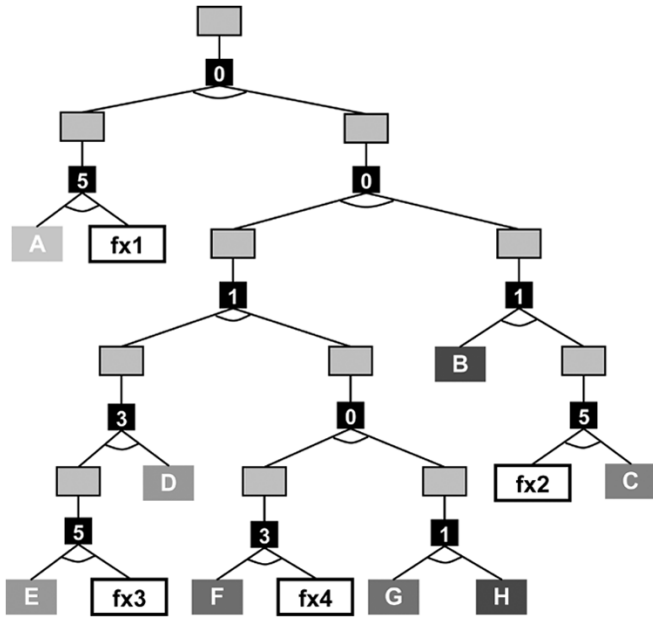


Fig. 18. One of 24 NDSTs whose cost vector is (8, 4, 5, 24).

- 2) Assemble F on G-fx2. Only one KC, $kc6$ is required for F, which is fixed by fx2. The other five DOFs are constrained by the lap-butt joints with G, thus there is no underconstraint to be controlled additionally.
- 3) Assemble E on F-G-fx2. Two KCs related to E; $kc1$ and $kc2$ are fixed by fx2. The other four DOFs are exactly constrained by the two lap joints from E.
- 4) Assemble D on E-F-G-fx2. Two KCs, $kc5$ and $kc6$ are fixed by fx2. The other four DOFs are exactly constrained by the two lap joints from D.
- 5) In parallel with step 4, place C on fx1. All four KCs related to C are fixed on fx1. The other two DOFs should be constrained additionally.
- 6) Assemble D-E-F-G on C-fx1, where $kc3$ and $kc7$ are realized by fx1. The other four DOFs are constrained by the lap joint from F to C and another lap joint from C to E.
- 7) Assemble B on C-D-E-F-G-fx1, where $kc4$ is realized. The lap joint from B to D and the lap joint from C to B constrains four DOFs, thus one DOF should be constrained additionally.
- 8) Assemble A on B-C-D-E-F-G-fx1, realizing $kc8$. The other five DOFs are fully constrained by one lap joint and one butt joint of A. All assembly steps are now completed.

Fig. 18 shows an NDST, which has eight parts, four fixtures, the depth of 5, and 24 underconstraints. This tree has the minimum depth and has a few parallel steps. For this reason, the tree is suitable for parallelized and short cycle time production. The price it pays is the many fixtures required to realize KCs in parallel.

On the other hand, NDSTs of eight parts, one fixture, the depth of 8, and three underconstraints (not shown), are completely serial, using only one fixture to control all the KCs. Because there are less fixtures, many DOFs are constrained by joints between parts, thus yielding mere three underconstraints

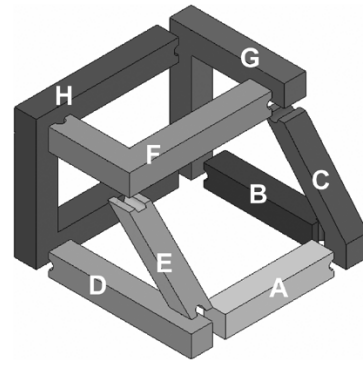


Fig. 19. Assembly design matching the NDST shown in Fig. 18.

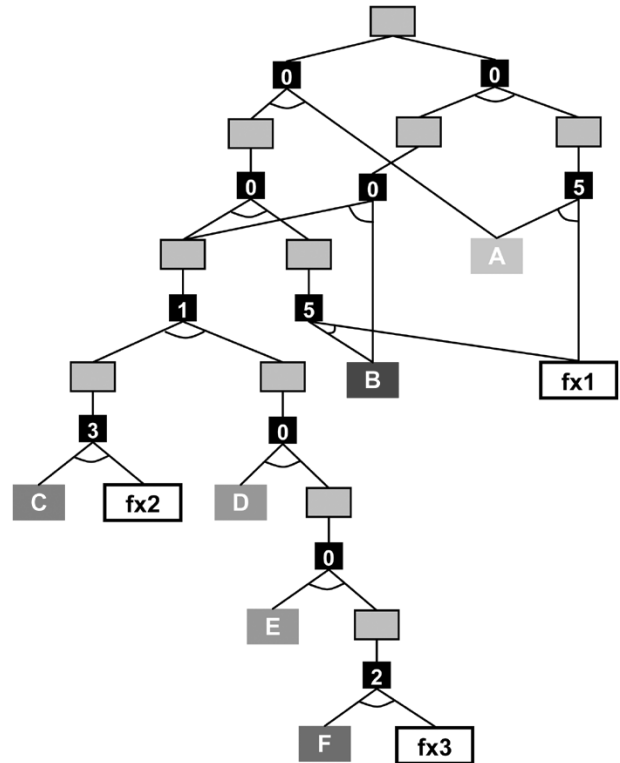


Fig. 20. Two of four NDSTs whose cost vector is (6, 3, 6, 11).

throughout the assembly. Instead, the production would require a longer cycle time. An assembly with many connections among parts and with many KCs is likely to have less parallel assembly sequence, because it is likely that the assembling two large sub-assemblies at the later stage would have many joints between the two subassemblies and realize many KCs at one step, thus more likely to have overconstraints.

There are only four trees that have the nondominated cost vector of (6, 3, 6, 11). Two of these are shown as an AND/OR graph in Fig. 20, which contains two different assembly sequences to build the assembly design shown in Fig. 21 (note the OR relation between the two hyper-edges from the top node). The other two trees for the same cost vector are mirror images of ones shown in Fig. 20, which has a corresponding assembly design that is also the mirror image of the one shown in Fig. 21.

Whereas other NDSTs have one or more decomposition(s) solely to remove an unplanar part, a closed loop, or a T-joint

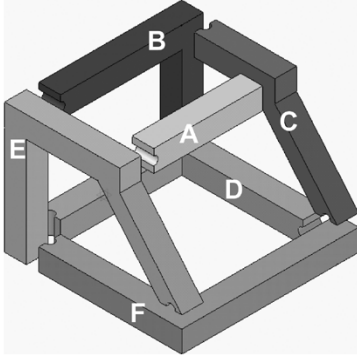


Fig. 21. Assembly design matching both trees in Fig. 22.

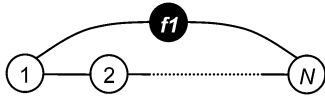


Fig. 22. Linearly connected members with one fixture element.

required by `stop_de`, without breaking a KC, all of the decompositions in these trees have been made to remove KCs. In other words, these trees show the most efficient way to remove KCs in terms of the number of decompositions. As a result, these solutions have minimum number of parts, six.

VII. DISCUSSION

Due to the enumerative nature of the presented approach, the amount of computation would grow rapidly as the product geometry gets complex. Although the amount of computation would be largely dependent on the number of members and KCs, how they are connected, joint types, and manufacturability criteria, two extreme models are presented to estimate bounds of computational complexity. In the following analyses, the number of prospective decompositions that must be analyzed is used as a measure of the amount of computation [23].

- 1) Let us consider a configuration with linearly connected members, of which two at the ends are connected to a fixture element (Fig. 22). In addition, suppose any subset of members satisfies the manufacturability criteria and, for every feasible decomposition, at least one joint assignment satisfies the decomposition rule. Because of the fixture element, the configuration does not satisfy the manufacturability criteria, thus subject to decomposition. The initial decomposition will result in a pair of subconfigurations $\{f1, 1, 2, \dots, n\}$ and $\{n+1, n+2, \dots, N\}$ for $1 \leq n \leq N-1$, as well as $\{1, 2, \dots, n\}$ and $\{n+1, n+2, \dots, N, f1\}$ for $1 \leq n \leq N-1$. Thus, the number of initial decompositions is $2(N-1)$. All configurations without $f1$ now satisfy the manufacturability criteria, thus no further decomposition is conducted. Configurations with n members and $f1$ (either $\{f1, 1, 2, \dots, n\}$ or $\{N-n+1, N-n+2, \dots, N, f1\}$) are now linearly connected graph with $n+1$ nodes, thus there are n ways to

TABLE II

AMOUNT OF COMPUTATION FOR: 1) THE EXAMPLE IN FIG. 15; 2) WITHOUT KCs; AND 3) WITH FIVE MEMBERS AND FOUR KCs REMOVED. TEST RUNS WERE CONDUCTED ON A 3.2-GHZ PENTIUM 4 PC RUNNING WINDOWS

	1	2	3
No. of members	12	12	7
No. of KCs	8	0	4
No. of hyper-edges	143269	32240	2621
No. of solution trees	8.38×10^9	1.03×10^8	9.92×10^3
computation time [sec]	2482	293.0	8.125

decompose each of them. Therefore, the total number of decompositions, D is

$$D = 2(N-1) + 2 \cdot \sum_{n=1}^{N-1} n = N^2 + N - 2. \quad (26)$$

- 2) As the other extremity, consider a configuration in which every member or fixture element is connected to every other member or fixture element, rendering a complete graph. Also, suppose only single member or fixture element satisfies the manufacturability criteria, and at least one joint assignment satisfies the decomposition rule at decomposition. As any subset of a complete graph is a feasible vertex cut, a configuration of n members and fixture elements has $2^{n-1} - 1$ decompositions, for $2 \leq n \leq N$. As we have ${}^N C_n$ configurations of size n , then

$$D = \sum_{n=2}^N \binom{N}{n} \cdot (2^{n-1} - 1) = \frac{3^N + 1}{2} - 2^N. \quad (27)$$

Some of the actual computation times are shown in Table II, which shows exponential property of computational complex. Although the enumerative approach is meaningful in that it shows all possibilities to the designer, especially when the number of solutions is small (as shown in [1] for 2-D cases), a multi-objective AO* algorithm [24], [25] will prune a large part of an AND/OR graph and show optimum solutions faster, which will be implemented in future works.

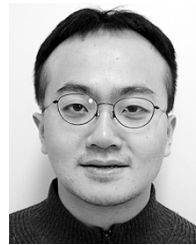
VIII. CONCLUSION

This paper shows that assembly design, sequence, and fixture scheme ensuring in-process dimensional adjustability and proper constraint can be enumerated by a decomposition-based procedure. It is applicable to synthesis of any kind of structural assemblies, provided that the geometry and available joint types are known. For more flexible assemblies, it would be practical to incorporate finite-element-method-based variation analysis, instead of Screw Theory. In all cases, careful definition of members based on manufacturing constraints and careful definition of KCs based on customers' needs should precede, in order to avoid a large number of unpractical solutions. As the enumerative search presented does not only require a significant amount of computation, but conflicts with many practical constraints, it is not desirable to entrust the whole product geometry to the assembly synthesis method, especially when the product geometry is fairly complex. The present method, therefore, would most effectively be integrated in the design process if it were applied

to subassemblies of a product first decomposed by a human designer. An alternative would be an initial run with coarse definition of members and global KCs, followed by subsequent runs for subassemblies selected at the previous runs with fine definition of members and local KCs.

REFERENCES

- [1] B. Lee and K. Saitou, "Decomposition-based assembly synthesis for in-process dimensional adjustability," *ASME J. Mech. Design*, vol. 125, no. 3, pp. 464–473, 2003.
- [2] D. E. Whitney, R. Mantripragada, J. D. Adams, and S. J. Rhee, "Designing assemblies," *Res. Eng. Design*, vol. 11, pp. 229–253, 1999.
- [3] D. L. Blanding, *Exact Constraint: Machine Design Using Kinematic Principles*. New York: ASME Press, 1999.
- [4] J. M. Kriegel, "Exact constraint design," *Mech. Eng.*, vol. 117, no. 5, pp. 88–90, 1995.
- [5] B. Lee and K. Saitou, "Three-dimensional synthesis for robust dimensional integrity based on screw theory," in *Proc. 5th Int. Symp. Tools and Methods of Competitive Engineering*, vol. 2, Lausanne, Switzerland, Apr. 2004, pp. 585–596.
- [6] R. S. Ball, *A Treatise on the Theory of Screws*. Cambridge, U.K.: Cambridge Univ. Press, 1900.
- [7] T. N. Whitehead, *The Design and Use of Instruments and Accurate Mechanism*. New York: Dover, 1954.
- [8] L. J. Kamm, *Designing Cost-Effective Mechanisms*. New York: McGraw-Hill, 1990.
- [9] K. Downey, A. R. Parkinson, and K. W. Chase, "An introduction to smart assemblies for robust design," *Res. Eng. Design*, vol. 14, no. 4, pp. 236–246, 2003.
- [10] K. J. Waldron, "The constraint analysis of mechanisms," *J. Mechanisms*, vol. 1, no. 2, pp. 101–114, 1966.
- [11] M. S. Ohwovoriole and B. Roth, "An extension of screw theory," *ASME J. Mech. Design*, vol. 103, pp. 725–735, 1981.
- [12] H. Asada and A. B. By, "Kinematic analysis of workpart fixturing for flexible assembly with automatically reconfigurable fixtures," *IEEE J. Robot. Automat.*, vol. RA-1, no. 2, pp. 86–94, Apr. 1985.
- [13] Y.-C. Chou, V. Chandru, and M. M. Barash, "A mathematical approach to automatic configuration of machining fixtures: Analysis and synthesis," *ASME J. Eng. Industry*, vol. 111, pp. 299–306, 1989.
- [14] E. C. DeMeter, "Restraint analysis of fixtures which rely on surface contact," *ASME J. Eng. Industry*, vol. 116, pp. 207–215, 1994.
- [15] J. D. Adams and D. E. Whitney, "Application of screw theory to constraint analysis of assemblies joined by features," *ASME J. Mech. Design*, vol. 123, no. 1, pp. 26–32, 2001.
- [16] D. J. Lee and A. C. Thornton, "The identification and use of key characteristics in the product development process," in *Proc. ASME Design Engineering Technical Conf.*, Irvine, CA, Aug. 1996, Paper no. 96-DETC/DTM-1506.
- [17] L. Woo and F. Freudenstein, "Application of line geometry to theoretical kinematics and the kinematic analysis of mechanical systems," *J. Mechanisms*, vol. 5, pp. 417–460, 1970.
- [18] K. H. Hunt, *Kinematic Geometry of Mechanisms*. Oxford, U.K.: Oxford Univ. Press, 1978.
- [19] B. Roth, "Screws, motors, and wrenches that can not be bought in a hardware store," in *Proc. Robotics Research, 1st Symp.*, Cambridge, MA, 1983, pp. 679–735.
- [20] L. R. Foulds, *Graph Theory Applications*. New York: Springer-Verlag, 1991.
- [21] N. J. Nilsson, *Principles of Artificial Intelligence*. Palo Alto, CA: Tioga, 1980.
- [22] L. S. H. de Mello and A. C. Sanderson, "AND/OR graph representation of assembly plans," *IEEE Trans. Robot. Automat.*, vol. 6, no. 2, pp. 188–199, Apr. 1990.
- [23] —, "A correct and complete algorithm for the generation of mechanical assembly sequences," *IEEE Trans. Robot. Automat.*, vol. 7, no. 2, pp. 228–240, Apr. 1991.
- [24] C.-F. Liaw, B. S. Stewart, and C. C. White III, "Multiobjective heuristic search in AND/OR graphs," *IEEE Trans. Syst., Man, Cybernet.*, vol. 28, no. 11, pp. 1513–1521, Nov. 1995.
- [25] P. Dasgupta, P. P. Chakrabarti, and S. C. DeSarkar, "Multiobjective heuristic search in AND/OR graphs," *J. Algorithms*, vol. 20, pp. 282–311, 1996.



Byungwoo Lee received the B.S. degree from Seoul National University, Seoul, Korea, in 1996, and the M.S. and Ph.D. degrees from the University of Michigan, Ann Arbor, in 2001 and 2004, respectively.

He is currently with GE Global Research, Niskayuna, NY. His research interests are in design automation and optimization, design for manufacture, and assembly modeling.

Dr. Lee is a member of the American Society of Mechanical Engineers. He was the co-recipient (with Dr. Saitou) of the Best Paper Award at the 5th International Symposium on Tools and Methods of Competitive Engineering, in 2004.



Kazuhiro Saitou (M'97) received the Ph.D. degree in mechanical engineering from the Massachusetts Institute of Technology (MIT), Cambridge, in 1996.

From 1997 to 2003, he was an Assistant Professor with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, where he is currently an Associate Professor. His research interests include design automation and optimization, design for X , where X = manufacture, assembly, robustness, and environment, modeling and optimization of micro electro-mechanical systems, and evolutionary computation in mechanical design.

Dr. Saitou is a member of American Society of Mechanical Engineers, society of manufacturing engineers (SME), the Association for Computing Machinery, and Sigma Xi. He currently serves as an Associate Editor for the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING and an editorial board member of the *International Journal of CAD/CAM* and *Genetic Programming and Evolvable Machines*. He was the recipient of the 1999 CAREER Award from the National Science Foundation.