Int J Adv Manuf Technol (2006) 30: 507–520
DOI 10.1007/s00170-005-0036-7

ORIGINAL ARTICLE

Tianyang Dong · Ling Zhang ·
Ruofeng Tong · Jinxiang Dong

A hierarchical approach to disassembly sequence planning for mechanical product

Received: 30 June 2004 / Accepted: 17 March 2005 / Published online: 20 December 2005

Abstract An important aspect of design for the life cycle is assessing the disassemblability of products. This paper presents a novel approach to automatic generation of disassembly sequence from hierarchical attributed liaison graph (HALG) representation of an assembly through recursively decomposing the assembly into subassemblies. In order to increase the planning efficiency, the HALG is built according to the knowledge in engineering, design and demanufacturing domains. In this method, the boundary representation (B-Rep) models are simplified by removing the hidden surfaces to reduce the computational complexity of disassembly planning. For each layer of HALG, the subassembly selection indices defined in terms of mobility, stability, and parallelism are proposed to evaluate the extracted tentative subassemblies and select the preferred subassemblies. To verify the validity and efficiency of the approach, a variety of assemblies including some complicated products are tested, and the corresponding results are presented.

Keywords Disassembly sequence planning · Hierarchical structure · Geometric reasoning · Assembly model

1 Introduction

The increased public consciousness of environmental issues has led to a growing concern with the environmental implications of product design, manufacturing processes, and recycling. Concerns about the environment have spurred interest in design for life cycle (DFLC). In DFLC, the designer considers the entire life of the product, from the initial conceptual design, through normal product use, to the eventual disposal of the product. DFLC includes two major components: design for disassembly and design for service [1]. In the present era of environmental awareness, the objectives such as component reuse, remanufacture and recycling constitute some of the most important reasons for disassembling products.

According to the method of disassembly, the process may be clearly classified into two categories: destructive disassembly and non-destructive disassembly. In the case of non-destructive disassembly, if a fastener is screwed in, then it is screwed out. If two parts are snapped together, then they are snapped out. In the case of destructive disassembly, brute force is used to remove the parts, so parts are just pulled or cut [2]. This paper focuses on non-destructive disassembly for mechanical products as a part of environment friendly manufacturing. Usually, disassembly planning consists of two major activities: assembly modeling and disassembly sequence planning. The input of the assembly representation is important to the effectiveness of every disassembly planning system. Thus, it is essential to develop a good assembly modeling method.

Generally, a detailed assembly model should consist of geometric information, component information, information about the different types of contacts, and information about the production environment of the product [3, 4]. The most commonly used method for assembly modeling is graph-based and is known as part mating graph. It represents the topological relation between components of a design, where the nodes represent the components and the arcs establish the relation between the components. Eastman [5] presented the location graph to describe a chain of part location relations by a set of transformation matrix operations. De Mello et al. [6] proposed the relational model graph to represent the assembly that includes parts, contacts and attachments relation in a model. Ko and Lee [7] developed the virtual link mating graph to capture the mating conditions between two components provided by a designer.

T. Dong (✉)
College of Software Engineering,
Zhejiang University of Technology,
Hangzhou 310032, China
e-mail: dtysun@hotmail.com
Tel.: +86-0571-85290034

L. Zhang
School of Economics and Management,
Zhejiang University of Science and Technology,
Hangzhou, China

R. Tong · J. Dong
Institute of Artificial Intelligence, Zhejiang University,
Hangzhou, China

Using the approaches to establish the graphs mentioned above requires more information than what is available directly from most CAD systems such as Pro/E or UG. The disassembly sequence, generated by mental analysis, guarantees a correct and exact sequence, but becomes progressively more complex when the number of components increases. Other approaches of assembly modeling based on the geometric reasoning technique have been proposed. However, there are cases that can not be solved by geometric reasoning alone. Therefore, it is desirable to integrate geometric reasoning with interactive approaches in assembly modeling. The computation complexity of assembly modeling can be reduced drastically and the disassembly sequences obtained are more feasible and practical.

The disassembly of a product can be total or partial. For example, in a recycling application, the disassembly of the entire product is necessary, whereas in a maintenance application, only the components to be replaced and that prevent their dismounting have to be disassembled. In the case of disassembling a product completely, the problems in disassembly sequence planning are related to those in assembly sequence planning. In summary, the existing approaches to generation of assembly plans can be roughly classified into three main approaches: human-interaction [8, 9], geometry-based reasoning [10–12], and knowledge-based reasoning [13–15]. The inverse of an assembly sequence can yield the disassembly sequence.

Disassembly sequence can be generated using interactive or automated technologies. The interactive approach mainly focuses on each user's query either on the connection between a pair of parts or the feasibility of a single disassembly operation. Interactive approaches can be used to deal with very complicated assemblies because information is not gathered from the geometry of parts, but instead the information is gathered from asking the designer questions. Clearly, this method is far from automation. Therefore, a number of automated approaches of disassembly sequence planning have been proposed.

Hari Srinivasan et al. [16] performed design for selective disassembly analysis on the CAD model of an assembly. Their methodology involves the following three steps:

1. Identifying the components to be selectively disassembled for demanufacturing by a software program or designer
2. Determining an optimal disassembly sequence for the selected components that involves a computationally efficient two-level reduction procedure

    a. Determination of a set of sequences with an objective of minimal component removals via a wave propagation approach
    b. Evaluation of resulting sequences based on an objective function to identify an optimal sequence

3. Performing disassembly design decisions based on the evaluated optimal sequence

Zahed and David [1] developed an approach to generating complete disassembly processes by combining interactive and automated approaches using virtual prototyping. They were interested in generating disassembly processes of product designs during early design stage. They used the automated approaches only to generate the partial disassembly sequences and then used designer's judgment or knowledge to complete the process.

Hu et al. [17] proposed an approach to disassembly sequence and path planning based on knowledge and geometric reasoning for mechanical products. This method uses the information provided by the mating features of parts in the product to find the candidate parts for disassembly and to carry out disassembly path planning. Additionally, a set of criteria and heuristic rules based on knowledge, constraints, relationships among parts, and quantitative disassemblability assessment are employed.

Anoop Desai and Anil Mital [18] addressed the issue of disassembly evaluation in mass production. A systematic methodology to incorporate disassembly considerations in product design and enable quantitative evaluation of the design has been developed. The methodology assigns time-based numeric indices to each design factor, which makes for easy and quick determination of disassembly time. A higher score indicates anomalies in product design from the disassembly perspective. Addressing these anomalies can result in significant design modifications rendering an overall increase in disassemblability of the product.

Torres et al. [19] offered a representation based on assemblies of components grouped by hierarchical levels that form an assembly. These can be considered as new components in other groups or assemblies. This type of representation, in addition to being intuitive, affords us a connection among operations or tasks to be done for the real disassembly, by automatically generating the sequence of operations to achieve the disassembly of a component or of a subassembly of the product.

Tsai C. Kuo [20] provided the disassembly sequence and cost analysis for the electromechanical products during the design stage. The disassembly planning is divided into four stages: geometric assembly representation, cut-vertex search analysis, disassembly precedence matrix analysis, and disassembly sequences and plan generation. The disassembly cost is categorized into three types: target disassembly, full disassembly, and optimal disassembly.

Kendra E. Moore et al. [21, 22] proposed an algorithm that automatically generates a disassembly Petri net (DPN) from a geometrically-based precedence matrix. The resulting DPN can be analyzed to generate all feasible disassembly process plans, and cost functions can be used to determine the optimal disassembly process plans. This approach can be used for products containing AND, OR, and complex AND/OR disassembly precedence relationships.

Askiner Gungor et al. [23] addressed the uncertainty related difficulties in disassembly planning, and presented an approach to disassembly sequence planning for products with defective parts in product recovery.

These methods and algorithms are not very interactive. For a complex assembly, a large amount of space is needed to store the representation of disassembly sequences. They are also difficult to generate detailed disassembly plans automatically [24]. A large amount of planning is needed to determine the more feasible and practical sequences for the product. Lee [25] presented a method for the automatic determination of preferred assembly partial orders from attributed liaison graph representation of a product, with the direct connection to assembly/disassembly cost. The approach identifies and avoids those decompositions that incur physically infeasible assembly operation. To generate the disassembly sequences automatically, the algorithm of subassembly identification and evaluation is used as the foundation.

In this paper, a novel method is proposed to automatically generate disassembly sequences from the CAD model by extracting preferred subassemblies layer upon layer. In automatically detecting the subassembly, this method makes use of less information obtained interactively from the user and takes into consideration recyclability of the product. In order to reduce the computational complexity in finding all the feasible subassemblies, the knowledge in engineering, design and demanufacturing domains is employed to build the HALG, and the simplified B-Rep models are used in disassembly sequence planning and collision detection. In this paper, we develop algorithms to hierarchically build HALG and generate the disassembly precedence graphs for each layer simultaneously. This implementation assists in significantly reducing the complexity and amount of planning to determine the more feasible and practical disassembly sequence for the product.

The remainder of this paper is organized as follows. Section 2 considers the representation problem of the assembly model, Sect. 3 describes the strategy of disassembly sequence planning, Sect. 4 gives an example to illustrate the proposed approach, while conclusions and areas for future research are finally discussed in Sect. 5.

2 Representation of assembly model

The representation problem for an assembly has received some attention due to the requirement of less storage and easy user understanding. A product is made up of a group of interconnected parts and/or subassemblies. A subassembly is also composed of a group of interconnected parts and/or subassemblies. An example of the hierarchical structure model for mechanical product is shown in Fig. 1. In the model, each of the colored nodes is a terminal node that does not have to be decomposed further. In the disassembly planning in the end-of-life phases, the following problems are the terminal nodes [26]:

(a) One part
(b) A subassembly to be reused
(c) A subassembly that is made of the same or compatible recyclable materials and does not include (b)
(d) A subassembly that is made of the materials creating the major sources of pollution and does not include (b)
(e) A subassembly that needs to be removed before shredding because of its hardness, etc. and does not include (b), (c) or (d)
(f) A subassembly that is made of the unrecyclable materials and does not include (b), (d) or (e)

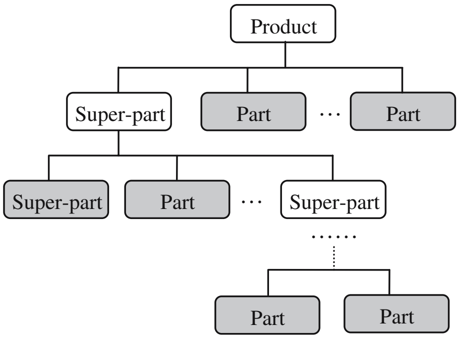


Fig. 1 Hierarchical structure model fassembly

The assembly hierarchy can be built on the basis of a spatial or functional decomposition of the product, and the bill of material (BOM) represents the hierarchy of the product. According to the eventual disposal of the product, the terminal nodes are defined. The subassemblies identified are called super-parts. When a single computer or network deals with a large or complex assembly, a special method to compress or simplify the assembly is needed. Therefore, we propose a method of simplifying B-Rep models, in which the surfaces that can't be seen from outside the model are removed. There are two models for each super-part, one is the simplified model and the other is the original model that has all detailed information for its parts or super-parts.

In our approach, an assembly is represented by HALG, which is extended from an attributed liaison graph [25]. A liaison is said to exist between a pair of parts or super-parts if one part or super-part constrains the freedom of motion of the other by a direct contact or a near contact.

A HALG is a connected graph, H=(N, E), where N represents a set of nodes and E represents a set of edges. A node n, $n \in N$, is assigned to each part or super-part of the assembly, and an edge e, $e \in E$, is assigned to the liaison between a pair of parts or super-parts. The super-parts, except the terminal nodes, have a respective child HALG in the next layer. In HALG, frames are attached to individual nodes and edges to describe the attributes associated with a node or edge. To enable automated disassembly sequence planning, all the related information should be organized and represented as an assembly model. Therefore, the HALG contains not only the geometric features involved in part mating and the interconnection mechanism between a pair of parts or super-parts associated with the liaison, but also the knowledge in engineering, design and demanufacturing domains. For the HALG, H=(N, E), where:

1. N is a set of nodes, each of which corresponds to a part or a super-part in the assembly. The attributes associated with a part or a super-part include:

    a. The part/super-part geometry (feature list or face list)
    b. The contact surfaces as part features
    c. The assembly/disassembly tools, such as screwdriver, spanner and gripper, etc.
    d. The node type (such as part node and super-part node)
    e. The child HALG
    f. The physical properties of a part, e.g. weight

2. E is a set of liaisons between two nodes, each of which corresponds to the connection between pairs of elements of N. The attributes of a liaison are described by:

    a. The connection type
    b. The mating type (such as *against*, *fit*, *screw-fit*, etc.)
    c. The male–female pairs of mating entities
    d. The contact surfaces
    e. The mating directions corresponding to the pairs of contact surfaces
    f. The relative stability of a liaison after the corresponding interconnection is completed

Figure 2 illustrates an example of HALG, where nodes corresponding to parts are ellipses and nodes corresponding to super-parts are rectangles. For each super-part, except the terminal nodes, there is a child HALG in the next layer.

3 Disassembly planning strategy

Weighted hierarchical attributed liaison graph (WHALG) is a HALG with a weight assigned to each of the edges for all layers. The weight of an edge is determined by the net strength of the edge in terms of physical stability and structural connectivity. In this research, WHALG is simplified into an abstract WHALG by merging those parts or super-parts that can’t be separable at the current stage of disassembly planning due to the interconnection feasibility. The extraction of subassemblies is based on the recursive decomposition of each child HALG into a set of subgraphs.

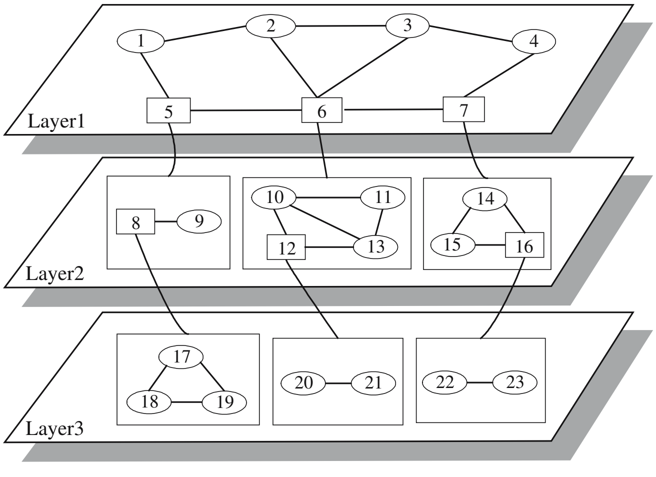


Fig. 2 An example of HALG

In order to reduce the complexity of search space in decomposition and generate cost-effective and feasible disassembly sequences, we present a hierarchical approach to disassembly sequence planning for large and complicated assemblies. The basic operating procedure of the algorithm is illustrated in Fig. 3.

In this algorithm, the planner first builds the super-parts according to the BOM of the product and the definition of terminal nodes and simplifies the models of the super-parts. Next, the HALG is built according to the hierarchical structure model of the mechanical assembly, and the WHALG is obtained by assigning weights to individual liaisons in HALG based on the stability and structural connectivity associated with liaisons. Then, each child WHALG in all layers of WHALG is decomposed and generates a disassembly sequence respectively. Last, all local disassembly sequences are merged into a global disassembly sequence.

3.1 Simplification of super-part models

Various ways has been proposed to simplify the models. Hoppe [27] and Hussain et al. [28] proposed some approaches of triangular mesh compression for storage and transmittance of triangular meshes, which are inapplicable to our research. Other researchers have considered features and topologies in the simplification of the models. Zhu [29] proposed an approach to simplify B-Rep models by automatic fillet/round suppressing, which utilizes an incremental knitting process to handle various topological structures of fillets and rounds. Koo et al. [30] adopted the wrap-around operation to make a multi-resolution model of parts and assemblies. In this method, the wrapping of products using kitchen plastic wrap was imitated.

In this research, we propose a method for simplifying B-Rep models, in which the surfaces invisible from outside the model are removed. There are two models for each super-part, one is the simplified model while the other is the original model that has all detailed information for its parts or super-parts.

In order to obtain a complete simplified B-Rep model by the simplification of super-part, a systematic approach is developed to guide the information processing. We also adopt the wrap-around operation to make a multi-resolution model of the super-part [30]. A convex edge is defined between two faces in a solid body, whereby if the angle made by two faces in the solid body lies between 0° and 180°, the edge is convex and if the angle lies between 180° and 360°, the edge is concave. A convex inner loop is a loop composed of only convex edges. This method is composed of two steps. The first step is the part level wrap-around

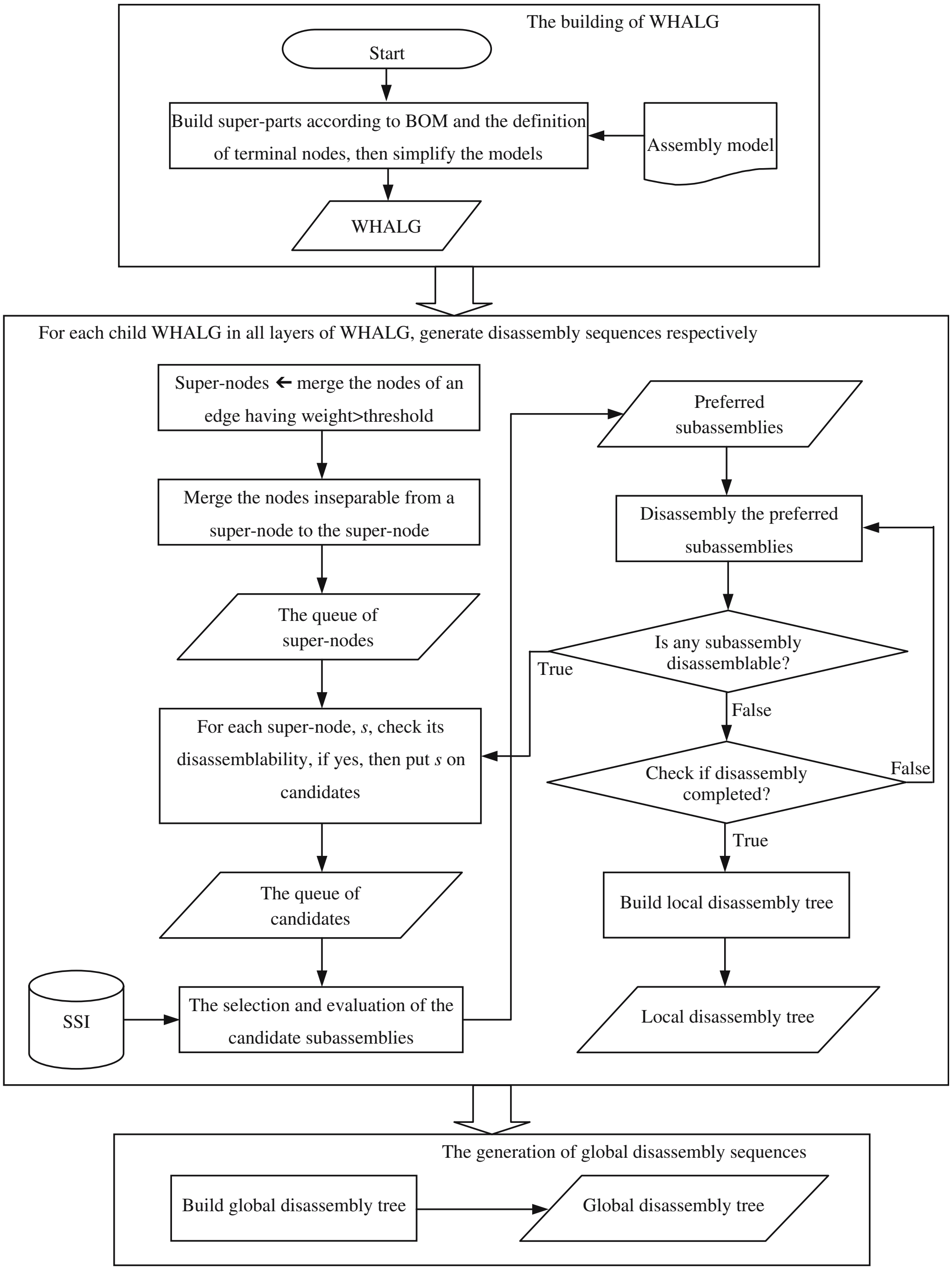

The building of WHALG
Start
Build super-parts according to BOM and the definition of terminal nodes, then simplify the models
Assembly model
WHALG
For each child WHALG in all layers of WHALG, generate disassembly sequences respectively
Super-nodes ← merge the nodes of an edge having weight>threshold
Merge the nodes inseparable from a super-node to the super-node
The queue of super-nodes
For each super-node, s, check its disassemblability, if yes, then put s on candidates
The queue of candidates
SSI
The selection and evaluation of the candidate subassemblies
Preferred subassemblies
Disassembly the preferred subassemblies
Is any subassembly disassemblable?
True
False
Check if disassembly completed?
False
True
Build local disassembly tree
Local disassembly tree
The generation of global disassembly sequences
Build global disassembly tree
Global disassembly tree

Fig. 3 The procedure of disassembly sequence planning

operation for the parts or super-parts that compose the simplified super-part. In this step, a convex inner loop is used as a clue to find a concave space and fill this space by removing the convex inner loop. After filling the concave space, the surfaces that can’t be seen from outside the model are removed. The level of detail in our model is defined using a set comprised of a convex inner loop and surfaces that are removed with the convex inner loop. The second step is an assembly level wrap-around operation. As a result of the first step, an overlap between parts exists and surfaces that can’t be seen from outside the model exist. These surfaces are deleted in the second step.

For the super-part shown in Fig. 4, there are 17 surfaces in the original model, but only six surfaces exist after removing the invisible surfaces. Only the simplified representations of the details for the super-part are used in disassembly planning, therefore, the algorithm of disassembly planning will be more efficient.

3.2 Construction of WHALG

WHALG is a HALG with weight assigned to each of the edges for all layers and the weight of an edge is determined by the net strength of the edge in terms of physical stability and structural connectivity. The net strength of an edge e_i, $S_N(e_i)$, is defined by the relative stability of e_i, $X_s(e_i)$, and

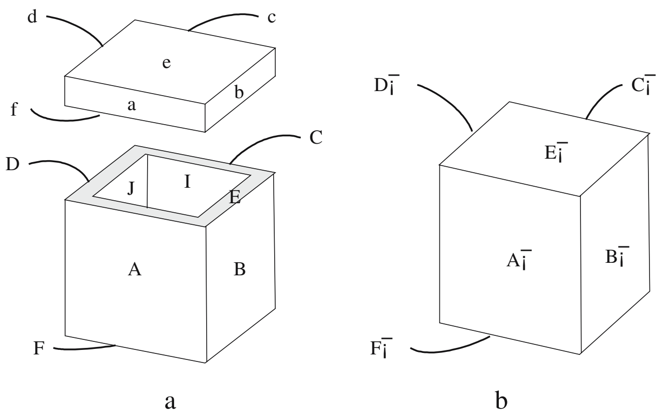


Fig. 4 An example of simplification of super-part. **(a)** Original model; **(b)** Simplified model

the directional constraints of a motion during part separation, $X_d(e_i)$, as follows [25]:

$$S_N(e_i) = \alpha X_s(e_i) + (1-\alpha) X_d(e_i)$$

where *alpha* is the weighting coefficient, $0 \le \alpha \le 1$.

In our system, the relative stability of e_i, $X_s(e_i)$ where $0 < X_s(e_i) \le 1$, is specified in the liaison frame based on the interconnection type of the edge. The relative stability for each interconnection type is illustrated in Table 1.

The directional constraint of e_i, $X_d(e_i)$ where $0 < X_d(e_i) \le 1$, represents the degree of how much the motion of the node n_{i1} is restricted by the node n_{i2} during the separation of e_i, or vice versa. $X_d(e_i)$ is obtained by $X_d(e_i) = 1 - \mathrm{DFS}(n_{i1},\ n_{i2})/6$. $\mathrm{DFS}(n_{i1}, n_{i2})$ represents the degree of freedom of separation between the two nodes n_{i1} and n_{i2}. The larger value of $X_d(e_i)$ implies that the freedom of separation is constrained by a larger degree and correspondingly has the effect of increasing the net strength of the edge. Usually, the approach to decide whether the node has the freedom of separation is geometric reasoning. It is very time consuming. In order to obtain the value of $X_d(e_i)$ efficiently, we propose a novel approach to calculate the degrees of freedom of separation. The mating conditions between the pair of nodes restrict the freedoms of separation. The freedoms of separation for nodes n_{i1} and n_{i2} satisfy the following conditions:

R1: The mating type between the nodes n_{i1} and n_{i2} is only against mating condition, no fit or screw-fit mating condition. Each against mating restricts a freedom of separation. If there are two or more assembly constraints that have the same mating direction, these assembly constraints only restrict a freedom of separation.

Table 1 Relative stability of interconnection

Interconnection Type	Relative stability
Mate	0.2
Insert	0.4
Bolt, bolt-nut, screw, pin	0.6
Key, roll-fit, gear, belt-mesh, bearing	0.8
Rivet, welding	0.9

R2: The mating type between the nodes n_{i1} and n_{i2} is only fit or screw-fit mating condition, no against mating condition. Each fit or screw-fit mating condition restricts four freedoms of separation. If there are two or more fit or screw-fit mating conditions, and their axes are parallel, these constraints restrict four freedoms of separation, else they restrict six freedoms of separation.

R3: If against and fit or screw-fit mating conditions coexist, each fit or screw-fit mating condition restricts four freedoms of separation. If there are two or more fit or screw-fit mating conditions and their axes are parallel, these constraints restrict four freedoms of separation, else they restrict six freedoms of separation. For each against mating condition, if its mating direction is perpendicular to the axes of fit or screw-fit mating, the against mating does not restrict more freedoms of separation, else one more freedom is restricted.

The proposed algorithm to calculate the degrees of freedom of separation is depicted in Fig. 5.

The weight of an edge is now determined by reinforcing $S_N(e_i)$ with the structural connectivity. The structural connectivity evaluates the strength of an edge in the context of its surrounding structure. The structural connectivity between the nodes n_{i1} and n_{i2} is due to the indirect as well as direct paths between n_{i1} and n_{i2}. The algorithm of reinforcing $S_N(e_i)$ with the structural connectivity is described as follows [25]:

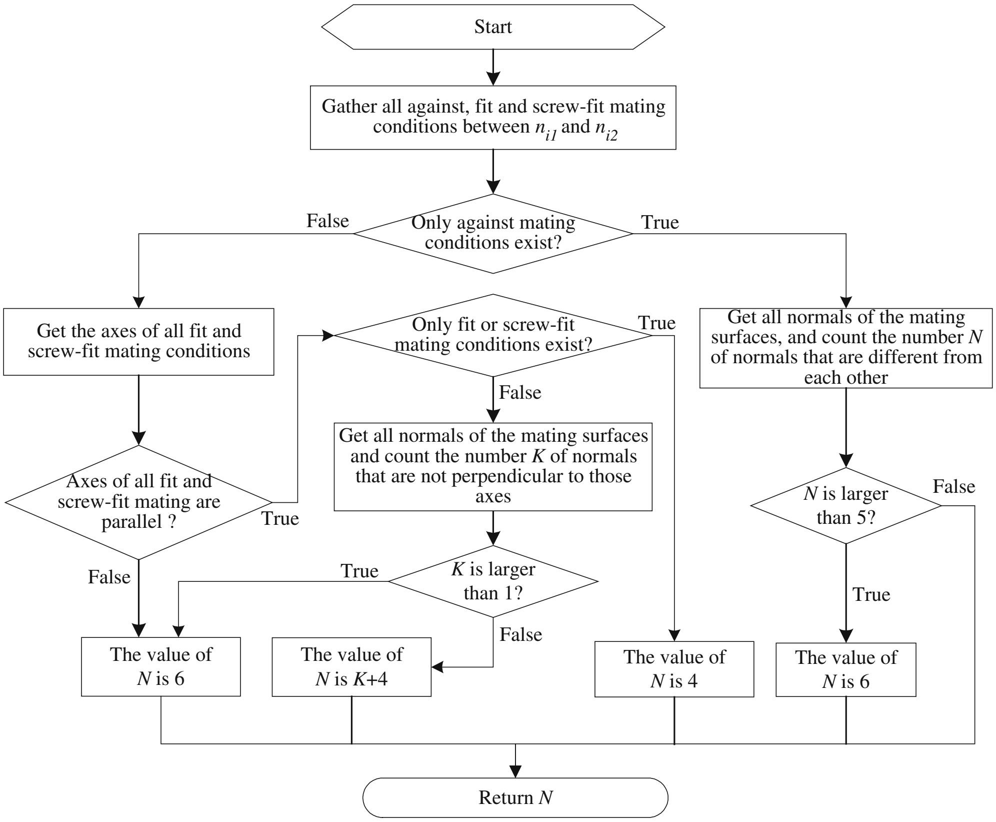
1. Find all the indirect paths between n_{i1} and n_{i2} of e_i, $\left\{P_j^i \middle| j = 1, \ldots, n\right\}$, sharing no common edges. Assume that P_j^i consists of a set of edges $\{e_{jl} | l = 1, \ldots, m_j\}$, ($j$=1,...,$n$).
2. Define the net strength of a path P_j^i, $S_N\left(P_j^i\right)$, by $S_N(P_j^i) = \prod_{l=1}^{m_j} S_N(e_{jl})$, for $j = 1,\ldots,n$.
3. Reinforce SN(e_i) by adding $\sum_{j=1}^{n} S_N(P_j^i)$ to SN(e_i). Therefore, $S_N'(e_i) = S_N(e_i) + \sum_{j=1}^{n} S_N(P_j^i)$.
4. The weight of the edge W(e_i) is achieved by normalizing $S_N'(e_i)$ such that the maximum value of W(e_i) is 1. Due to the normalization, each W(e_i) has a value between 0 and 1.

3.3 Disassemblability of subassemblies

3.3.1 Representation of all feasible removal directions

A set of tentative subassemblies is generated by successively merging nodes in WHALG. A disassemblable tentative subassembly has at least one nonzero linear translation element in the current configuration of the assembly. A nonzero linear translation indicates that a part or super-part can be moved away from its initial position along the

Fig. 5 Algorithm to calculate the degrees of freedom of separation


Start
Gather all against, fit and screw-fit mating conditions between n_{i1} and n_{i2}
Only against mating conditions exist?
False
True
Get the axes of all fit and screw-fit mating conditions
Only fit or screw-fit mating conditions exist?
True
Get all normals of the mating surfaces, and count the number N of normals that are different from each other
False
Get all normals of the mating surfaces and count the number K of normals that are not perpendicular to those axes
Axes of all fit and screw-fit mating are parallel ?
True
N is larger than 5?
False
False
True
K is larger than 1?
True
False
The value of N is 6
The value of N is K+4
The value of N is 4
The value of N is 6
Return N

corresponding removal direction. The disassemblability and one of the feasible removal directions of the components can be computed from the normals of the mating surfaces. Mating surfaces are a subset of boundaries shared by a pair of parts or super-parts. Usually, the local removal directions can be considered as a space in the unit sphere. Considering a sphere of unit radius, a mating surface effectively divides the sphere into two hemispheres. As shown in Fig. 6, the hemisphere, H_1, that has the pole of $\overrightarrow{C^i_{jk}}$ which corresponds to the inward pointing unit normal of the mating surface in the unit sphere, contains all feasible removal directions.

Figure 6 shows an example of the *against* mating condition for the part P_k which is to be separated from the remainder of assembly P_j. $\overrightarrow{SU^i_{jk}}$ is one of the feasible removal directions. $\overrightarrow{C^i_{jk}}$ and $\overrightarrow{SU^i_{jk}}$ can be represented as $\overrightarrow{C^i_{jk}} = (x_c, y_c, z_c)$ and $\overrightarrow{SU^i_{jk}} = (x_s, y_s, z_s)$, respectively. For H_1, $\overrightarrow{C^i_{jk}}\ \overrightarrow{SU^i_{jk}} \geq 0$ or $x_c x_s + y_c y_s + z_c z_s \geq 0$ .The space containing all removal directions for a tentative subassembly can be computed by intersecting all the hemispheres corresponding to their mating surfaces. For a vector d (x_s, y_s, z_s) to be a local removal direction, it must satisfy the condition of the space in Eq. (1).

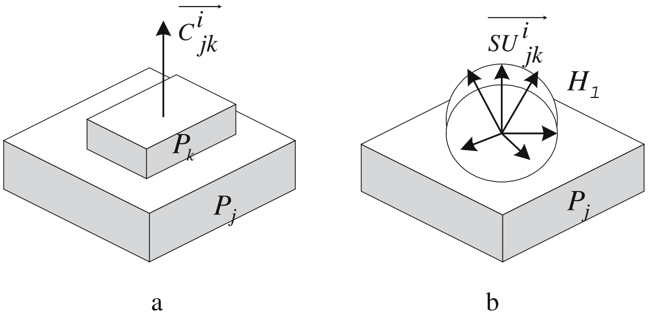


Fig. 6 The feasible removal directions of against mating condition. **(a)** Against mating condition; **(b)** Its feasible removal directions

$$D = H_1 \cap H_2 \cap H_3 \cdots \cap H_n$$

or

$$D = \left\{ \begin{matrix} x_{c1}x_s + y_{c1}y_s + z_{c1}z_s \geq 0 \\ x_s^2 + y_s^2 + z_s^2 = 1 \end{matrix} \right\} \cap \left\{ \begin{matrix} x_{c2}x_s + y_{c2}y_s + z_{c2}z_s \geq 0 \\ x_s^2 + y_s^2 + z_s^2 = 1 \end{matrix} \right\} \cdots \cap \left\{ \begin{matrix} x_{cn}x_s + y_{cn}y_s + z_{cn}z_s \geq 0 \\ x_s^2 + y_s^2 + z_s^2 = 1 \end{matrix} \right\} \quad (1)$$

3.3.2 Compute the local feasible direction using the discrete unit sphere

Because the feasible space containing all local removal directions is defined as a set of n inequalities and an equality, it is difficult to choose the optimum local removal direction in the space. Usually, the direction corresponding to the center of gravity for the feasible spherical surface is selected as an optimum local removal direction. However, obtaining the center of gravity for the feasible spherical surface is the problem of a multiple integral, for which it is difficult to determine the boundaries. Thus, a novel approach to find the optimum local removal direction is presented by applying the discrete unit sphere algorithm to analyzing assembly constraints. This approach computes the range of all feasible local directions of removal for the part and considers their mating conditions in calculating disassemblability and feasible removal direction. In our approach, we discretize the *XOY* plane in nonuniform size (Fig. 7). The *XOY* plane is divided into $n \times n$ grids and the sizes of the grids near the origin *O* of the sphere are larger than those of the grids far away from *O*. If the width of the immediate grid of *O* is *a*, that of the next grid is $at(0 < t \leq 1)$, and the rest may be deduced by analogy. So the width of the farthest grid is $at^{\frac{n}{2}-1}$, and there is $a + at + \cdots + at^{\frac{n}{2}-1} = 1$.

For any point P_i within a radius of one unit in the spherical surface, x_i and y_i are the values of the *X* and *Y* axes, respectively, which satisfy $x_i^2 + y_i^2 \leq 1$. Then we can calculate the values of z_i by applying the condition of $x_i^2 + y_i^2 + z_i^2 = 1$. If the disassembly vector (x_i, y_i, z_i) satisfies Eq. 1, the direction corresponding to the vector is one of the feasible local removal directions. Therefore, we can obtain an optimum local removal direction by computing the average value of x_i, y_i and z_i of all feasible local removal directions. If there are *k* vectors feasible to disassemble the part or super-part, the optimum local removal direction is:

$$\overline{x} = \sum_{i=1}^{k} x_i \Big/ k \; \overline{y} = \sum_{i=1}^{k} y_i \Big/ k \; \overline{z} = \sum_{i=1}^{k} z_i \Big/ k.$$

The algorithm implemented to compute the local feasible direction by using the discrete unit sphere is described as follows:

Step 1
Gather all against, *fit* and *screw-fit* mating conditions between the part or super-part to be disassembled and the remainder of the assembly, if there are only against mating conditions, then go to step 2, else go to step 3.

Step 2
If there is only one *against* mating condition, the inward pointing normal of the mating surface is a local removal direction. If there are more than one against mating conditions, the local removal direction can be computed by using the discrete unit sphere algorithm, thus go to step 6.

Step 3
If there are only fit or screw-fit mating conditions, then go to step 4, else go to step 5.

Step 4
If there is one *fit* or *screw-fit* mating condition, the axis of the fit or *screw-fit* mating is a local removal direction. If there are more than one *fit* or *screw-fit* mating conditions, check if all axes are parallel, if so, the vector corresponding to these axes is a local removal direction; else the subassembly is not disassemblable, thus go to step 6.

Step 5
If the against, *fit* and *screw-fit* mating conditions coexist, check if all axes of *fit* and *screw-fit* mating conditions are parallel and lie in the feasible removal space determined by the against mating conditions. If yes, the vector corresponding to these axes is a local removal direction for the subassembly; else the subassembly is not disassemblable.

Step 6
Check if the subassembly can be removed along the local removal direction without interference with the remainder of the assembly. If yes, the local removal direction is feasible; else the subassembly is not disassemblable.

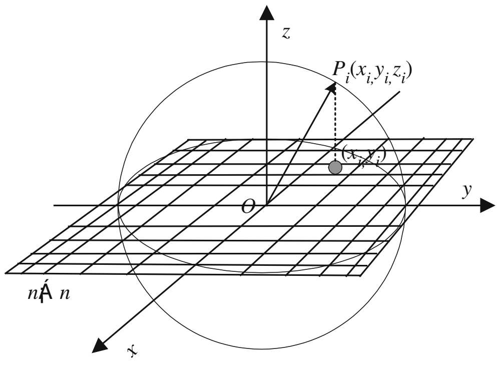


Fig. 7 Discrete unit sphere

3.3.3 Global collision free test

If a part or super-part moves in a linear trajectory along the candidate local removal direction, any subset of the remainder components other than the mating components involved in the intersection iteration may interfere with the part or super-part somewhere along the path. Therefore, a global interference check is needed to avoid such an occurrence. The global disassembly direction is finally determined by verifying that the disassembled part is free from interference on its removal path.

In order to check global interference efficiently, a simple approach uses a bounding box to envelope the object. Therefore the approach can simplify the object geometry for checking interference. However, this test does not always provide the correct answer. The imaginary boxes for non-contact objects of the assembly may report intersection. Thus, in this approach, we project all surfaces of the assembly onto a specified plane, and use the bounding boxes of the parts or super-parts to check interference. Three conditions exist between two boxes: non-intersection, partial intersection and complete enclosure, as shown in Fig. 8. The non-intersection case indicates the absence of global

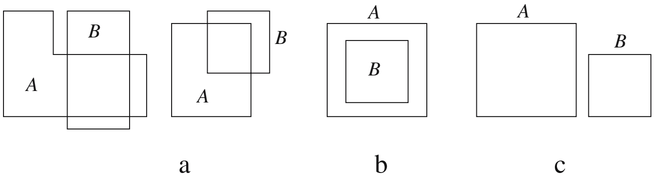


Fig. 8 Three conditions exist between two boxes. **(a)** Partial intersection; **(b)** Complete enclosure; **(c)** Non-intersection

collision. However, for the partial intersection and complete enclosure case, the result of an interference check depends on the moving object. The procedure on how to check global interference for the tentative subassemblies can be easily implemented [3].

3.4 Selection for optimal solutions

In most cases, the user wants to obtain disassembly plans containing as many disassembly sequences as possible. On the other hand, if we can quickly determine how many different operation sequences exist, then, based on the number of possible sequences, we can decide to evaluate all or only a portion of all possible operation sequences to choose the most appropriate evaluation and search techniques for finding an optimal, or at least efficient, operation sequence. Therefore, it is necessary to select a preferred subassembly out of the multiple tentative subassemblies that can not be extracted simultaneously due to the sharing of common nodes. The preferred subassemblies can be selected by evaluating tentative subassemblies based on subassembly selection indices (*SSI*). In our approach, the *SSI* evaluates a cluster of parts in the subassembly based on the following criteria:

1. ***Mobility index (MI):*** The mobility index, *MI(S)*, of a subassembly *S* represents how loosely the parts inside the subassembly *S* are connected to each other. *MI(S)* is determined by *DFS* and the relative stability of interconnections of individual parts and part clusters embedded in the subassembly *S* [25].

$$MI(S) = \sum_{l} DFS(P_i/\overline{P_i}) \times (1 - \text{maximum relative stability of the interconnections between } P_i \text{ and } \overline{P_i}) \tag{2}$$

Either P_i or $\overline{P_i}$ is selected for disassembly. A higher *MI* value implies that a subassembly *S* can be easily broken during disassembly operation.

2. ***Structural preference index (SPI):*** The structural preference index, SPI(S), represents how easily the subassembly *S* can be separated from the remainder of the assembly. *SPI(S)* is determined by *DFS* and the weights of edges between *S* and $\overline{S}$ [25].

$$SPI(S) = DFS(S_i/\overline{S_i}) \times (1 - \text{the maximum weight of the edges between } S_i \text{ and } \overline{S_i}) \tag{3}$$

where $S_i \cup \overline{S_i}$ represents the original assembly.

3. ***Operation continuity index (CNI)***: The operation continuity index specifies how much the operator must make the extra motion by exchanging mating directions and tools. The index *CNI*(S) is defined by Eq. 4:

$$CNI(S) = a_1 \times DRI + a_2 \times TLI \tag{4}$$

where $a_1$and a_2 are the coefficients and $a_1 + a_2 = 1$. *DRI* indicates the change of the disassembly direction. If the disassembly direction of the subassembly is different from that of the subassembly removed the previous time, the value of *DRI*=(*the angle of direction change*)/360. Thus, if the disassembly direction is the same as that of the subassembly removed the previous time, the value of *DRI* is 0. *TLI* depends on whether its own tools are the same as those of the subassembly removed the previous time. If the tools are different, the value of *TLI* is 1, else the value is 0.

4. ***Parallelism index (PI):*** Parallelism can be measured approximately by the depth of a disassembly partial order graph, relative to the number of parts to be disassembled. Although the exact depth can be determined only by generating the whole disassembly order, we can assign a parallelism index, *PI(S)*, to a subassembly *S* to evaluate the local contribution of *S* to the overall parallelism. The index *PI* is defined by Eq. 5:

$$PI(\mathrm{S}) = |N_s - N_{\bar{s}}| / |N_s + N_{\bar{s}}| \tag{5}$$

where N_S and $N_{\bar{s}}$ represent the number of parts in *S* and $\overline{S}$. A lower value of *PI* implies the decomposition of an assembly into two subassemblies of similar number of parts. It is clear that if the assembly can be recursively

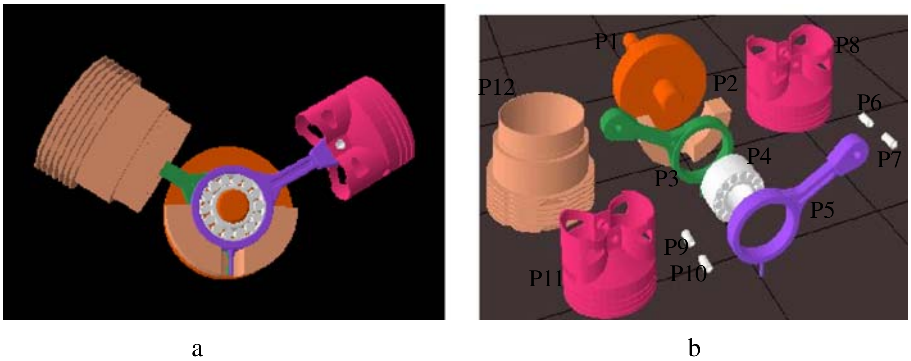


Fig. 9 The model of Wave-Hand. **(a)** Solid model of Wave-hand; **(b)** Its exploded view

decomposed into two subassemblies of equal number of parts, the resulting disassembly partial order graph should have the minimum depth [25].

5. ***Subassembly selection index SSI:*** The *SSI* is given as follows:

$$SSI(\mathrm{S}) = e^{-k_1 \cdot MI + k_2 \cdot SPI - k_3 \cdot CNI - k_4 \cdot PI} \tag{6}$$

where k_1, k_2, k_3 and k_4 are the disassembly coefficients and $\sum_{i=1}^{4} k_i = 1$. The subassembly that has the highest *SSI* value is selected as the next subassembly to disassemble. Note that the system prefers to select the subassembly with smaller M*I*, *CNI*, and *PI* values, but larger *SPI*. The coefficients can be assigned by the designer based on the relative significance of each selection index on the overall disassembly cost [25].

4 Case study

In order to illustrate the efficiency of the planner, we present an example of disassembly sequence planning for Wave-hand (Fig. 9), which consists of 12 parts. According to the hierarchy of BOM and the eventual disposal of the components, Wave-hand is composed of three parts (P1, P2 and P4) and two subassemblies (Sub2.1 and Sub2.2) on the first layer. The subassemblies Sub2.1 and Sub2.2 are composed of five and four parts respectively. Therefore, the WHALG of Wave-hand has two layers, as shown in Fig. 10. In order to simplify the graph, the label of weight with each edge is omitted.

To build the plans for disassembly, the WHALG and the geometric information of the solid model are input into the system. Table 2 shows how the value of each W(e_i) is calculated. It is noted that the assignment of weights relies upon various heuristic functions having a number of parameters associated with them. For each layer, the recursive application of the decomposition process and the subassembly selection based on the *SSI* values result in a hierarchical disassembly plan. In the experiment, equal weights are assigned to all selection indices, i.e. k_1=k_2=k_3=k_4=0.25.

After computing the weights of WHALG, the threshold is set to 1, which is the highest weight in each layer of WHALG; there is no merged node. Therefore, the disassemblability of each node is checked and disassemblable nodes are selected as tentative subassemblies. In this case, for layer1, P1, P2, P4 and Sub2.2 are selected as tentative subassemblies. The

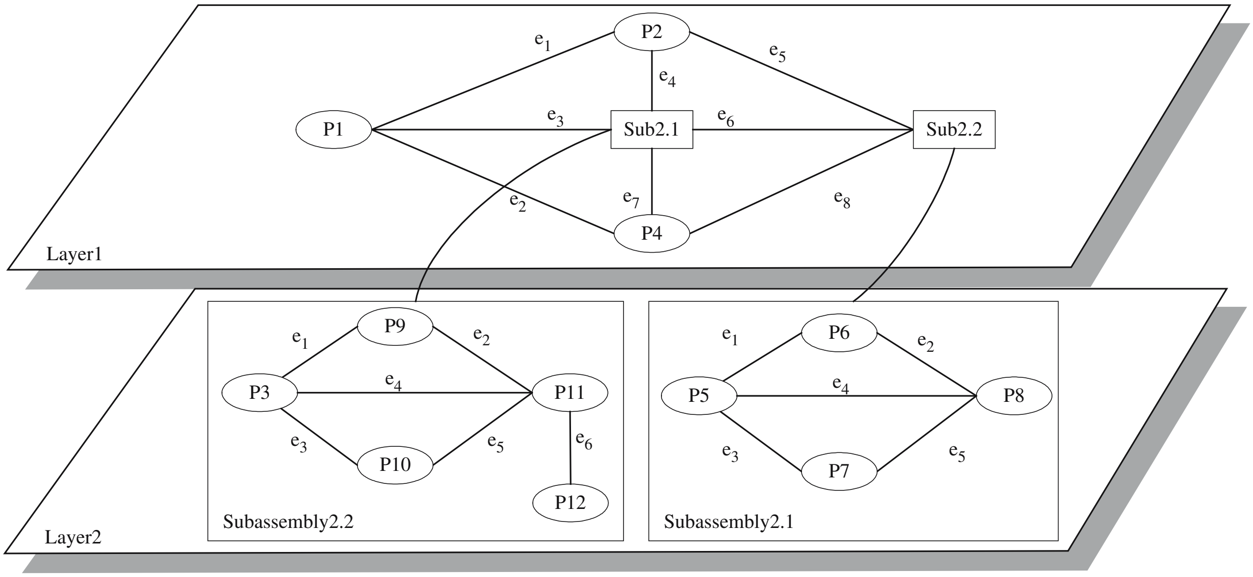


Fig. 10 HALG of Wave-hand

Table 2 Calculation of edge weights in HALG for Wavehand

Layer	edge	$X_s(e_i)$	DFS(e_i)	$X_d(e_i)$	$S_N(e_i)$	$S_N'(e_i)$	W(e_i)
Layer1	e_1	0.2	1	0.8333	0.5167	0.7062	0.8579
	e_2	0.4	5	0.1667	0.2834	0.4729	0.5745
	e_3	0.2	1	0.8333	0.5167	0.8101	0.9841
	e_4	0.4	4	0.3333	0.3667	0.8232	1
	e_5	0.4	4	0.3333	0.3667	0.5562	0.6757
	e_6	0.2	1	0.8333	0.5167	0.7856	0.9543
	e_7	0.4	4	0.3333	0.3667	0.7026	0.8535
	e_8	0.4	4	0.3333	0.3667	0.5562	0.6757
Layer2.1	e_1	0.6	4	0.3333	0.4667	0.6329	0.7998
	e_2	0.6	5	0.1667	0.3834	0.5857	0.7402
	e_3	0.6	4	0.3333	0.4667	0.6329	0.7998
	e_4	0.2	2	0.6667	0.4334	0.7913	1
	e_5	0.6	5	0.1667	0.3834	0.5857	0.7402
	e_6	0.4	5	0.1667	0.2834	0.2834	0.3581
Layer2.2	e_1	0.6	4	0.3333	0.4667	0.6329	0.7998
	e_2	0.6	5	0.1667	0.3834	0.5857	0.7402
	e_3	0.2	2	0.6667	0.4334	0.6329	0.7998
	e_4	0.6	4	0.3333	0.4667	0.7913	1
	e_5	0.6	5	0.1667	0.3834	0.5857	0.7402

Table 3 A set of tentative subassemblies of layer1 for Wave-hand

Layer	Threshold	New subassembly	Disassemblable subassembly
Layer1	1	{P1},{P2},{P4},{Sub2.1},{Sub2.2}	{P1},{P2},{P4},{Sub2.2}
	0.9841	S1={P2, Sub2.1}	{P1},{P2},{P4},{Sub2.2}
	0.9543	S2={P1, S1}	{P1},{P2},{P4},{Sub2.2}
	0.8579	S3={Sub2.2, S2}	{P1},{P2},{P4},{Sub2.2},{S3}
	0.8535		{P1},{P2},{P4},{Sub2.2},{S3}
	0.6757		{P1},{P2},{P4},{Sub2.2},{S3}
	0.5745		{P1},{P2},{P4},{Sub2.2},{S3}

next lower threshold value of 0.9841 makes {*P2*, *Sub2.1*} merge into a super-node S1, and {P2, Sub2.1} is disassemblable, so S1 is selected as a tentative subassembly. This decomposition process is continued all the way down to the lowest threshold value in this layer of WHALG. Table 3 shows all corresponding tentative subassemblies with several different threshold settings for layer1 of WHALG.

Because there are five tentative subassemblies, which share some common nodes, they can not be removed simultaneously. A preferred subassembly needs to be selected out of the multiple tentative subassemblies. In this case, the preferred subassemblies are selected according to the subassembly selection indices. For layer1 of WHALG, selection indices are shown in Table 4. The subassembly P4 has the highest value of SSI, so P4 is removed first, and the remainder of the product is S3. The subassembly Sub2.2 and P1 have the same value of 0.8745, thus any one can be selected to be removed. In this case, we disassemble the subassembly Sub2.2.

Table 4 Subassembly selection indices of layer1

Subassembly	*MI*	*SPI*	*CNI*	*PI*	*SSI*
P1	0	0.0636	0	0.6000	0.8745
P2	0	0	0	0.6000	0.8607
P4	0	0.5860	0	0.6000	0.9965
Sub2.2	0	0.0636	0	0.6000	0.8745
S3	0.2464	0.5860	0	0.6000	0.9369

After the disassembly of Sub2.2, the subassembly Sub2.1 is disassemblable, so *SSI* of Sub2.1 is computed as shown in Table 5. Now the *SSI* of Sub2.1 is 0.9348 higher than those of P1 and P2, so Sub2.1 is removed before P1 and P2. The hierarchical partial order graph is shown Fig. 11.

For layer 2.1, the tentative subassemblies are selected in the same way as those of layer 1. Table 6 shows all corresponding tentative subassemblies with several different threshold settings for layer 2.1 of WHALG, and Table 7 illustrates their selection indices.

Table 5 Subassembly selection indices of layer1 after disassembly of Sub2.2

Subassembly	*MI*	*SPI*	*CNI*	*PI*	*SSI*
Sub2.1	0	0.0636	0	0.3333	0.9348

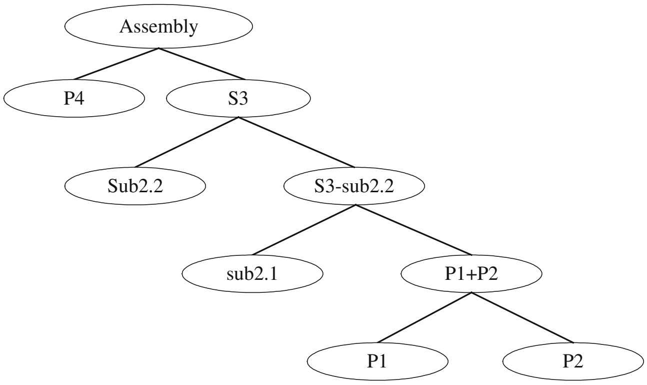


Fig. 11 The hierarchical partial order graph of layer 1

There are only two disassemblable subassemblies. The subassembly P12 having the highest value of SSI is removed first, and the remainder of the product is S2. After disassembly of P12, the subassemblies P9 and P10 are disassemblable, their selection indices are shown in Table 8. Table 9 shows the *SSI* of P3 and P11 after disassembling P9 and P10. The local hierarchical partial order graph of Sub2.1 is shown in Fig. 12.

For layer 2.2, the tentative subassemblies can be selected in the same way as layer 2.1. Fig. 13 shows the local hierarchical partial order graph of Sub2.2.

The planner has been implemented partially with programming in C++ based on the DENEB ENVISION system on a Pentium IV compatible-PC. In order to analyze the efficiency of the hierarchical approach proposed in this paper, a set of experiments was contrived using several industrial examples such as *toy motorgrab* (Fig. 14) and *vacuum cleaner* (Fig. 15). *Toy motorgrab* is composed of twenty-two parts and *vacuum cleaner* is composed of forty-eight parts. In some ways, these products which consist of so many parts are bad cases for any purely geometric reasoning that generates a large amount of the AND/OR graph. In our system, the hierarchy of the product and the eventual disposal of the product are considered in disassembly sequence planning. In the analysis, the module of model simplification in the planner was disabled and separate runs were done. In this section, we study the behavior of the planner on some industrial products, and demonstrate that the use of the simplified models can drastically improve performance in the case. In this experiment, the value of *t* is 0.92 and the *XOY* plane is divided into 100×100 grids, that means *n*=100.

Table 6 A set of tentative subassemblies of layer 2.1 for Wave-hand

Layer	Threshold	New subassembly	Disassemblable subassembly
Layer2.1	1	{P3},{P9},{P10},{P11}, {P12}	{P12}
	0.7998	S1={P3,P11}	{P12}
	0.7402	S2={S1,P9,P10}	{P12},{S2}
	0.3581		{P12},{S2}

Table 7 Subassembly selection indices of layer 2.1

Subassembly	*MI*	*SPI*	*CNI*	*PI*	*SSI*
P12	0	3.2095	0	0.6000	1.9201
S2	2.0020	3.2095	0	0.6000	1.1640

Table 8 Subassembly selection indices of layer1 after disassembly of P12

Subassembly	*MI*	*SPI*	*CNI*	*PI*	*SSI*
P9	0	3.9990	1	0.5000	1.8678
P10	0	3.9990	1	0.5000	1.8678

Table 9 Subassembly selection indices of layer1 after disassembly of P9 and P10

Subassembly	*MI*	*SPI*	*CNI*	*PI*	*SSI*
P3	0	2	1	0	1.2840
P11	0	2	1	0	1.2840

The graph in Fig. 16 compares the run-times with and without the use of simplified models. In the graph, we see exponential growth in the run-times without the use of simplified models. When the module of model simplifica-

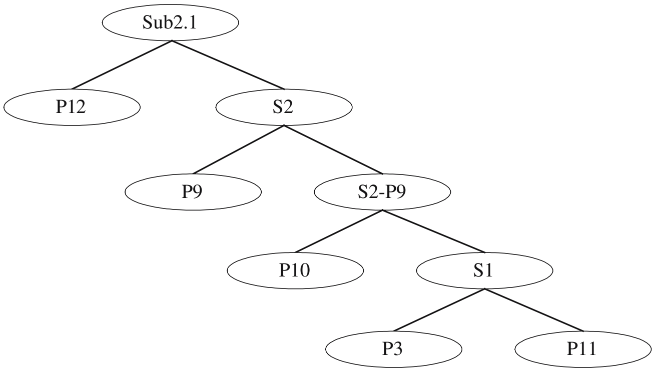


Fig. 12 The hierarchical partial order graph of layer 2.1

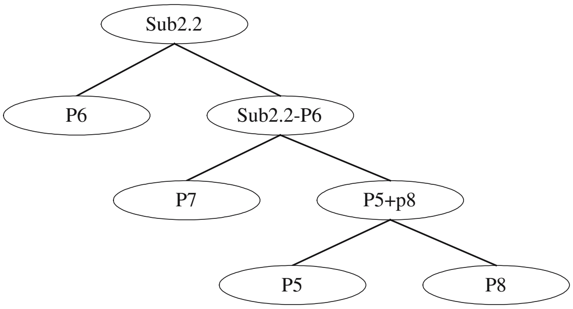


Fig. 13 The hierarchical partial order graph of layer 2.2

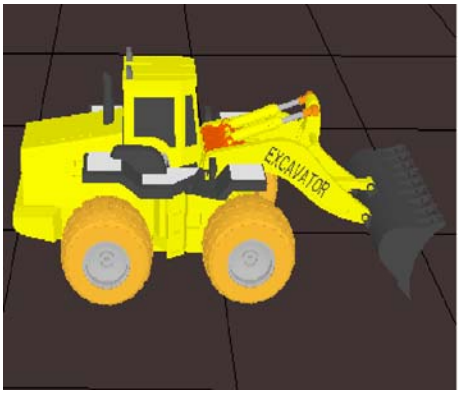


Fig. 14 Toy Motor Grab

tion is turned on, the performance of the planner improves significantly. In this case, these simplified models can be considered as new components in other groups or assemblies. This type of representation, in addition to being intuitive, affords us a connection among operations or tasks to be done for the real disassembly by automatically generating the sequence of operations to achieve the disassembly of a component or of a subassembly of the product. As shown in Fig. 16, the hierarchical approach to generate disassembly sequences for simple assembly, such as the product Wave-hand, is not very desirable. However, for the complicated assemblies, the approach is quite effective. It should be noted that the time spent in the simplification of models must be considered and the time saved by the use of simplified models may be less dramatic than in these cases. But the simplified models can be used not only in disassembly sequence planning but also in the collision detection. Thus, the influence of model simplification for complicated assemblies is great.

The most commonly used method of gaining the local feasible direction reported in the literature is to calculate the volume centroid of a surface or polysurface that corresponds to the local disassembly direction set [31]. In order to present the comparison between our method of computing the local feasible direction with the traditional method, we implement these two methods in our system. The graph in Fig. 17 compares the run-times of our method and the traditional method. In the graph, we see exponential growth in the run-times of the traditional method. It should be noted that the more grids are divided, the more time is spent in disassembly planning.

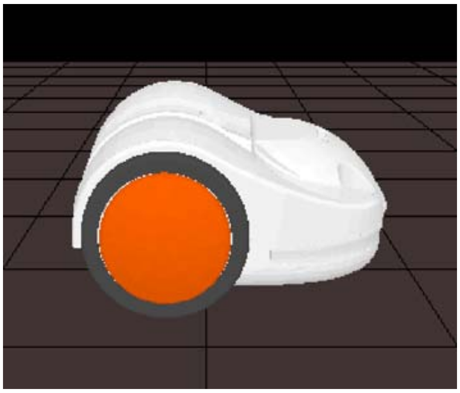


Fig. 15 Vacuum cleaner

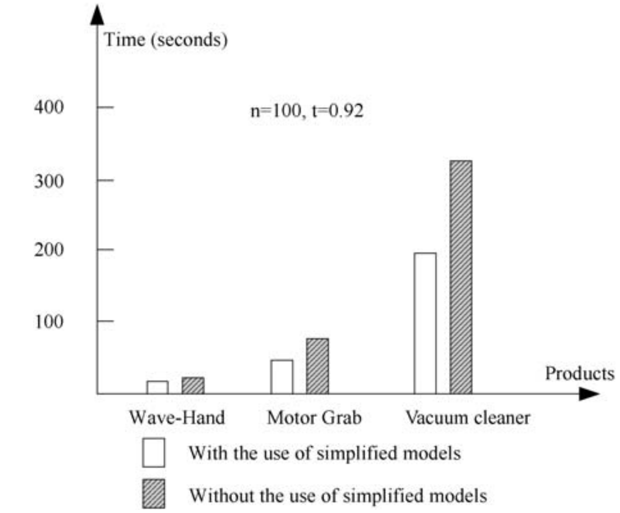


Fig. 16 The run-times of planning for the industrial assemblies

5 Conclusion

This paper proposes a hierarchical approach for generating disassembly sequences based on HALG. The knowledge in engineering, design and demanufacturing domains is employed to build the HALG, and the simplified B-Rep models are used in disassembly sequence planning and collision detection. This implementation assists in significantly reducing the complexity and amount of planning to determine the more feasible and practical sequences for the disassembly. In this paper, disassembly sequence planning examples are provided to illustrate the practicality and effectiveness of the

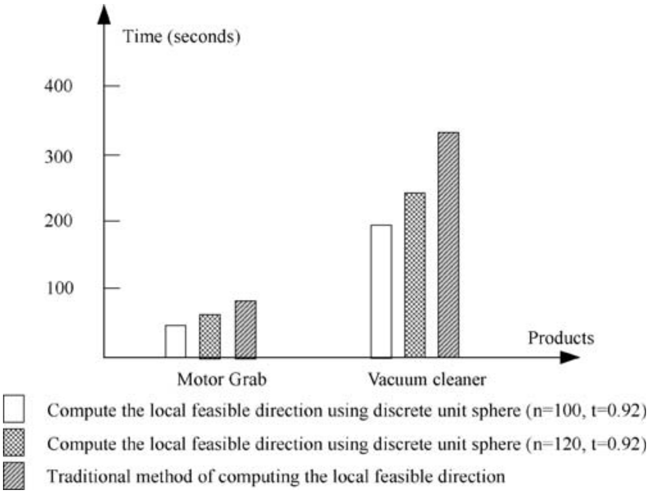


Fig. 17 The comparison between our method to compute the local feasible direction with the traditional method

hierarchical approach. The applications show that this approach can reduce the computational complexity drastically and obtain the more feasible and practical disassembly plans. Although the approach integrates general geometric reasoning with the knowledge about how to generate feasible and practical disassembly sequences, there remains something to do. In the future, we will pay more attention to non-geometric information in the entire life of the product that can be utilized in disassembly planning. Furthermore, we will promote the selection criteria by applying the approach to more various disassembly environments and thus find better subassemblies.

Acknowledgements We gratefully acknowledge the support of the Special Funds for Major State Basic Research Program (973) of China under grant 2002CB312106 and Zhejiang Natural Science Foundation of China under grant 602092. We would also like to thank the editors and the anonymous referees for their insightful comments.

References

1. Siddique Z, Rosen DW (1997) A virtual prototyping approach to product disassembly reasoning. Comput Aided Des 29:847–860
2. Kuo TC, Huang S, Zhang H (2001) Design for manufacture and design for ‘X’: concepts, applications and perspectives. Comput Ind Eng 41:241–260
3. Eng T–H, Ling Z–K, Olson W, et al (1997) Feature-based assembly modeling and sequence generation. Comput Ind Eng 36:18–33
4. Yuan B, Zhou Y, Hu SM, Sun JG (2000) The assembly model of hierarchical components. J Comput Aided Des Comput Graph 12:450–454 (in Chinese)
5. Eastman CM (1981) The design of assembly. SAE Technical Paper Series 0148-7191/81/0223–0197
6. Homem De Mello LS, Sanderson AC (1991) A correct and complete algorithm for the generation of mechanical assembly sequences. IEEE T Robotic Autom 7:228–240
7. Ko H, Lee K (1987) Automatic assembling procedure generation from mating conditions. Comput Aided Des 19:3–10
8. DeFazio TL, Whitney DE (1987) Simplified generation of all mechanical assembly sequences. IEEE T Robotic Autom 3:640–658
9. Wilson RH (1995) Minimizing user queries in interactive assembly planning. IEEE T Robotic Autom 11:308–312
10. Homem de Mello LS, Sanderson AC (1991) A correct and complete algorithm for the generation mechanical assembly sequences. IEEE T Robotic Autom 7:228–240
11. Baldwin DF, Abell TE, Lui M–CM, De Fazio TL, Whitney DE (1991) An integrated computer aid for generating and evaluating assembly planning. IEEE T Robotic Autom 7:78–94
12. Chakrabarty S, Wolter J (1997) A structure-oriented approach to assembly sequence planning. IEEE T Robotic Autom 13:14–29
13. Swaminathan A, Barber KS (1996) An experience-based assembly sequence planner for mechanical assemblies. IEEE T Robotic Autom 12:252–266
14. Yin ZP, Ding H, Li HX, Xiong YL (2003) A connector-based hierarchical approach to assembly sequence planning for mechanical assemblies. Comput Aided Des 35:38–56
15. Zha XF, Du H (2002) A PDES/STEP-based model and system for concurrent integrated design and assembly planning. Comput Aided Des 34:1087–1110
16. Srinivasan H, Figueroa R, Gadh R (1999) Selective disassembly for virtual prototyping as applied to de-manufacturing. Robot Com-Int Manuf 15:231–245
17. Hu D, Hu Y, Li C (2002) Mechanical product disassembly sequence and path planning based on knowledge and geometric reasoning. Int J Adv Manuf Technol 19:688–696
18. Desai A, Mital A (2003) Evaluation of disassemblability to enable design for disassembly in mass production. Int J Ind Ergon 32:265–281
19. Torres F, Puente ST, Aracil R (2003) Disassembly planning based on precedence relations among assemblies. Int J Adv Manuf Technol 21:317–327
20. Kuo Tsai C (2000) Disassembly sequence and cost analysis for electromechanical products. Robot Com-Int Manuf 16:43–54
21. Moore KE, Gungor A, Gupta SM (1998) A petri net approach to disassembly process planning. Comput Ind Eng 35:165–168
22. Moore KE, Gungor A, Gupta SM (2001) Petri net approach to disassembly process planning for products with complex AND/OR precedence relationships. Eur J Oper Res 135:428–449
23. Gungor A, Gupta SM(1998) Disassembly sequence planning for products with defective parts in product recovery. Comput Ind Eng 35:161–164
24. Ong NS, Wong YC (1999) Automatic subassembly detection from a product. Int J Adv Manuf Technol 15: 425–431
25. Lee S (1994) Subassembly identification and evaluation for assembly planning. IEEE T Syst Man Cyb 24:493–503
26. Murayama T, Kagawa K, Oba F (1999) Computer-aided redesign for improving recyclability. Proceedings of EcoDesign’99: 1st international symposium on environmentally conscious design and inverse manufacturing, Tokyo, Japan, pp 746–751
27. Hoppe H (1996) Progressive meshes. Proceedings of SIGGRAPH1 ’96, New York, ACM Press, pp 99–108
28. Hussain M, Okada Y, Niijima K (2001) Efficient simplification of polygonal surface models. Proceedings of the 5th international conference on information visualisation, London, England, pp 464–469
29. Zhu H, Menq CH (2002) B-rep model simplification by automatic fillet/round suppressing for efficient automatic feature recognition. Comput Aided Des 34:109–123
30. Koo S, Lee K (2002) Wrap-around operation to make multi-resolution model of part and assembly. Comput Graph 26:687–700
31. Tian LZ, Fu YL, Ma YL, et al (2001) Finding disassembly direction in assembly sequence planning. J Comput Aided Des Comput Graph 13:1110–1113 (in Chinese)