Proceedings of the 1999 IEEE
International Symposium on Assembly and Task Planning
Porto, Portugal - July 1999

Selective Disassembly: Representation and Comparative Analysis of Wave Propagation Abstractions in Sequence Planning

Hari Srinivasan[1] and **Rajit Gadh**[2]

[1] Doctoral Candidate, hari@smartcad.me.wisc.edu
[2] Associate Professor, gadh@engr.wisc.edu
Department of Mechanical Engineering,
1513 University Avenue, #347
University of Wisconsin-Madison,
Madison, WI - 53706, USA.

ABSTRACT

This paper analyses the problem of removing one or more components of an assembly, defined as selective disassembly, and presents geometric abstractions and representations for automated selective disassembly analysis of geometric models.

Based on the number of target components to be disassembled from an assembly (**A**) of **n** components, the selective disassembly (**SD**) problem is categorized into three classes. (I) Single component disassembly (**1-SD**): the motivation is that the disassembly analysis can be localized with respect to the target component and analysis of all components for disassembly may not be required. (II) Disassembling **s** ($1 < s << n$) components from **A** (defined as **s-SD**): the motivation is that a better solution may be obtained if two or more components are disassembled along a common sequence. (III) Disassembling **S** (**S→n**) components from **A** (defined as **S-SD**): the motivation is to reduce the computational complexity, at the same time determining a locally optimal solution.

Some preliminary results of a prototypical system, *Assembly Disassembly in Three Dimensions* (**A3D**), implementing the **SD** abstractions, and analysis of the abstractions for different classes of **SD** problems are presented.

INTRODUCTION

Selective Disassembly (**SD**) involves disassembling a subset of components (**C**) from an assembly (**A**) to obtain a **SD** sequence (**S**). For example, to disassemble **C** = {C_4, C_5} from **A** in Figure 1a, **S** = {C_6, C_4, C_5}, as shown in Figure 1b.

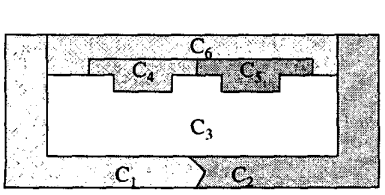

Figure 1a. Test Assembly to illustrate **SD**

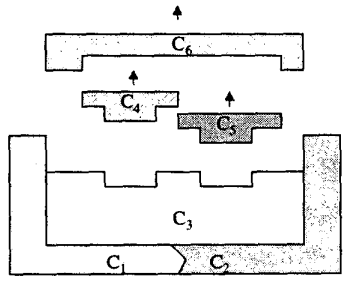

Figure 1b. S = {C_6, C_4, C_5} for C = {C_4, C_5}

Applications such as maintenance, recycling, and reuse usually require removal of a subset of components of **A**, and not the entire assembly, hence providing a need for **SD** analysis [11]. For example, aircraft engine maintenance requires the **SD** of the engine and not the disassembly of the entire aircraft. Another application of **SD** is in reuse application requiring removal of some high-valued components such as **SD** of an instrument panel (for its reuse) from a car assembly.

Selective Disassembly Planning

This research focuses on automatic determination of **S** for **C** from **A**. The **SD** *problem* is formulated as follows:

- *Given:* An assembly **A** and a selected set of components **C** to be disassembled.
- *Requirement:* Determine automatically **S** for **C** with minimal component removals.

The objective of minimal component removals is appropriate, since for 1-disassemblable components (a 1-disassemblable component requires a single linear motion to be removed from **A** [15]) the objective becomes minimizing the disassembly motions (operations), which is a measure of difficulty of disassembling [6]. Moreover, the product design for manufacture suggests simple motions and easier separation of components [1, 3, 4] for maintenance/recycling. Also, the above objective is consistent with earlier research [15] in **SD** analysis. Therefore, **S** with minimal component removals is defined as an Optimal Sequence (**OS**).

EFFICIENT SELECTIVE DISASSEMBLY

One potential approach to determining **OS** is an exhaustive enumeration of all the possible sequences and the selection of **OS** with minimal removals. However, this analysis is computationally expensive (typically exponential with respect to the number of components in **A**), and is not recommended.

Several researchers (e.g. [7, 8, 13, 14]) have proposed automated Complete Disassembly **CD** algorithms, which involves disassembling all the components in **A**. An application for **CD** is assembling, since reversing a **CD** sequence can potentially yield an assembly sequence (e.g. [6, 13, 15]). An extensive amount of research in **CD** sequencing and assembly planning exists. Although **S** can be obtained from a **CD**, it may not give an optimal solution [10].

Another approach is the construction of a Disassembly Tree [15] for **CD** and a single component disassembly. The tree is designed to model the 'Onion Peeling' algorithm—recursively disassembling removable components, starting from the boundary of **A** and proceeding inwards. The Disassembly Tree approach [15] is proposed for 2.5 dimension objects and the analysis is based on the contact geometry. However, the algorithm is only applicable for assemblies in which every component is disassembled by removing none or one of its mating adjacent components. Therefore, the above approach is restrictive for our use.

In the **SD** problem, the requirement is to identify **S** to disassemble **C**. However, apart from the objective that the **SD** analysis should be automatic and analyze 3D geometric models, there are two other important issues: (i) Computationally efficient algorithms and (ii) Optimum **SD** solution. Efficiency and optimality are related, and one is usually achieved at the cost of the other. For example, if efficiency is the only issue, then any of the **CD** solution can be extended for **SD**. However, this results in a non-optimum solution. On the other hand, if optimality is the only issue, then exhaustive enumeration will give an optimum solution; however, this approach is computationally inefficient.

Therefore, the current research attempts to provide algorithms that balance the requirements of computationally efficient and optimal solution; i.e., determining a **SD** solution with fewer component removals that can be computed in a feasible computation time.

Based on the number of components to be disassembled, the **SD** problem of disassembling **C** from **A** of **n** components is categorized in to three classes:

1. **SD** of one component--defined as Single **SD** (**1-SD**),
2. **SD** of **s** << **n** components, defined as Multiple **SD** (**s-SD**), and
3. **SD** of **S→n** components, defined as Large **SD** (**S-SD**).

Each class of **SD** problem is studied, and analysis of some methods for automated **SD** is presented in subsequent sections. Prior to the analysis, some terms related to **SD** algorithms are defined.

DEFINITIONS

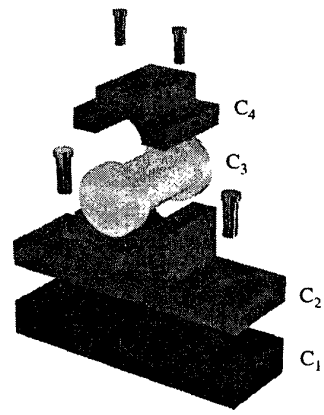

Figure 2. Test assembly

- *Disassemblability*: Δ_i is a binary value that indicates if $\mathbf{C_i} \in \mathbf{A}$ is removable [9]. For example, in Figure 2, Δ_4 = TRUE for $\mathbf{C_4}$ and Δ_3 = FALSE for $\mathbf{C_3}$.
- *Removal Influence*: $\mathbf{RI_i^j}$ is a binary value that indicates if $\mathbf{C_i} \in \mathbf{A}$ is removable after the removal of $\mathbf{C_j} \in \mathbf{A}$ [9]. For example, in Figure 2, $\mathbf{RI_3^4}$ = TRUE, since Δ_3 is TRUE with removal of $\mathbf{C_4}$ in **A**. Similarly, $\mathbf{RI_2^4}$ = FALSE.
- *1-dependent component:* A $\mathbf{C_i}$ requiring disassembly of one component or none of its adjacents for infinitesimal translation in **A** is defined as *1-dependent component* [9]. For example, in Figure 2, $\mathbf{C_3}$ is 1-dependent; i.e., infinitesimal translation of $\mathbf{C_3}$ is possible by disassembling $\mathbf{C_4}$.

Assumptions of the Current Research

The assumptions for the current research are:

1. The relative motions of the components are determined without considering the tools, fixtures or robots required to achieve these motions.
2. Assemblies are assumed to be rigid, frictionless and defined by nominal geometry.
3. Components are 1-disassemblable (single linear motion to be removed from **A**) and 1-dependent.
4. Disassembly sequences are sequential, monotonic, and non-destructive (no component is destroyed).

Assumptions 1-4 are standard assumptions followed by different researchers (e.g. [7, 13, 15]) in automated assembly/disassembly analysis. The 1-disassemblable assumption is utilized by several researchers (see e.g. [6, 8, 15]), since automatic generation of disassembly sequences allowing general disassembly motion is computationally expensive. Moreover, the 1-disassemblable assumption is realistic for some real world examples, as indicated by some existing assembly planning systems (e.g. [8]).

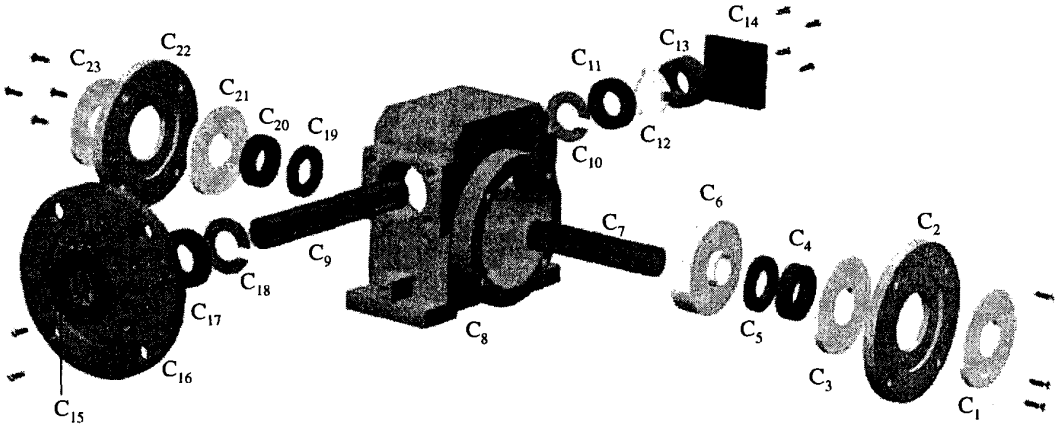

Figure 3. Gear Reducer Subassembly

SINGLE SELECTIVE DISASSEMBLY

The motivation for **1-SD** problem is that the disassembly analysis can be a localized with respect to the target component $\mathbf{C_x}$, and analysis of all components in **A** for disassembly may not be required.

In Srinivasan and Gadh [9], an algorithm called *Single Wave Propagation* (**SWP**) has been introduced for single component disassembly. The **SWP** algorithm defines a wave to topologically arrange components in **A** to denote the disassembly order such that a component in one wavefront, τ_{a-1} ($(\mathbf{a-1})^{th}$ wavefront), can be disassembled by removing its adjacent component(s) in τ_a ($\mathbf{a}^{th}$ wavefront).

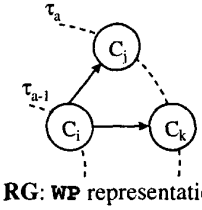

Figure 4. **RG**: **WP** representation

A disassembly wave is represented by a *Removal influence Graph* (**RG**) [9], whose nodes correspond to components in the disassembly wave and arcs correspond to the removal influence between the components. An edge $\mathbf{C_i \rightarrow C_j}$, indicating that $\mathbf{C_i}$ is disassemblable after removing $\mathbf{C_j}$, represents a **WP** from $\mathbf{C_i}$ to $\mathbf{C_j}$. For example, Figure 4 illustrates a **WP** from τ_{a-1} to τ_a, where $\mathbf{C_i} \in \tau_{a-1}$, $\mathbf{C_j} \in \tau_a$, Δ_i = FALSE and $\mathbf{RI_i^j}$ = TRUE. The **WP** from $\mathbf{C_i}$ to $\mathbf{C_j}$ implies that $\mathbf{C_i}$ is disassemblable after removing $\mathbf{C_j}$ (similarly for $\mathbf{C_i \rightarrow C_k}$).

SWP algorithm defines $\mathbf{C_x}$ as a wave source, with the waves propagating outwards until a removable component (defined as the boundary component $\mathbf{C_b}$) is reached. The wavefronts provide a hierarchy for component removal. $\mathbf{OS} = \{\mathbf{C_b} \overset{P}{\sim} \mathbf{C_x}\}$ is derivable from **RG**: where $\mathbf{C_b} \overset{P}{\sim} \mathbf{C_x}$ denotes the shortest path from $\mathbf{C_b}$ to $\mathbf{C_x}$ in the graph **RG**. To illustrate this, consider **A** in Figure 3 with $\mathbf{C} = \{\mathbf{C_x}\} = \{\mathbf{C_3}\}$, Figure 5 shows the **WP** from $\mathbf{C_3}$. $\mathbf{C_3}$ is disassemblable by removing $\mathbf{C_2}$ or $\mathbf{C_4}$ in τ_1 ($\mathbf{RI_3^2}$ = TRUE, $\mathbf{RI_3^4}$ = TRUE). $\mathbf{C_4}$ is disassemblable after removing $\mathbf{C_5}$ in τ_2 ($\mathbf{RI_4^5}$ = TRUE) and $\mathbf{C_2}$ is disassemblable after removing $\mathbf{C_1}$ in τ_2 ($\mathbf{RI_2^1}$ = TRUE). Since Δ_1 = TRUE for $\mathbf{C_1}$ (a boundary component), $\mathbf{OS} = \{\mathbf{C_1}, \mathbf{C_2}, \mathbf{C_3}\}$ for $\mathbf{C_3}$.

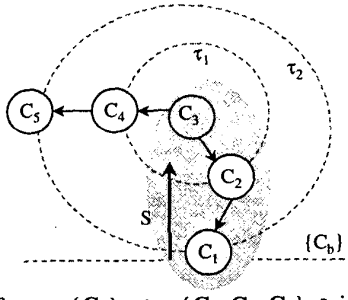

Figure 5. **RG** for $\mathbf{C} = \{\mathbf{C_3}\}$, $\mathbf{S} = \{\mathbf{C_1}, \mathbf{C_2}, \mathbf{C_3}\}$: **A** in Fig. 3

MULTIPLE SELECTIVE DISASSEMBLY

One potential algorithm to perform **s-SD** analysis is by applying **SWP** for every component in **C**. Although this approach may determine **OS** for individual target components, the resultant **S** (aggregation of all disjoint **OS**) is not necessarily **OS**. To illustrate the **s-SD** problem, consider **A** in Figure 3 with the requirement to disassemble $\mathbf{C} = \{\mathbf{C_5}, \mathbf{C_{19}}\}$. Let $\mathbf{n_S}$ = number of components in **S**. Applying **1-SD** algorithm results in $\mathbf{S} = \{\mathbf{C_1}, \mathbf{C_2}, \mathbf{C_3}, \mathbf{C_4}, \mathbf{C_5}, \mathbf{C_{23}}, \mathbf{C_{22}}, \mathbf{C_{21}}, \mathbf{C_{20}}, \mathbf{C_{19}}\}$ with $\mathbf{n_S}$ = 10. But better sequences $\{\mathbf{C_1}, \mathbf{C_2}, \mathbf{C_3}, \mathbf{C_4}, \mathbf{C_5}, \mathbf{C_6}, \mathbf{C_{19}}\}$ and $\{\mathbf{C_{23}}, \mathbf{C_{22}}, \mathbf{C_{21}}, \mathbf{C_{20}}, \mathbf{C_{19}}, \mathbf{C_6}, \mathbf{C_5}\}$ exist with $\mathbf{n_S}$ = 7. Therefore, a better solution may be obtained if two or more components are disassembled along a common sequence. This motivates the need for **s-SD** analysis.

In Srinivasan and Gadh [10], an algorithm called *Multiple Wave Propagation* (**MWP**) has been introduced for multiple components disassembly. The **MWP** algorithm, defines two types of disassembly waves: τ and β waves that determine the disassembly ordering with

respect to **C** and the β, boundary of **A**, respectively. For example, Figure 6a illustrates a τ wave from τ_{a-1} to τ_a, where $C_i \in \tau_{a-1}$, $C_j \in \tau_a$, Δ_i = FALSE and RI_i^j = TRUE. The τ wave from C_i to C_j implies that C_i is disassemblable after removing C_j. Similarly, the β wave from $C_j \in \beta_{a-1}$ to $C_i \in \beta_a$ is represented as $C_i \rightarrow C_j$ denoting that C_i is disassemblable after disassembling C_j. Figure 6b shows a β wave from $C_j \in \beta_{a-1}$ to $C_i \in \beta_a$, implying the minimum number of components to be removed to disassemble C_j from **A** is (**a**-1) and C_i is **a**.

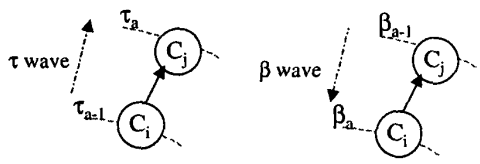

Figure 6. (a) τ Wave (b) β Wave

Based on the intersection event (**IE**) between τ and β waves, a sequence to disassemble **C** is determined. The importance of the intersection between waves lies in the determination of the component at which the waves intersect (therefore the shape of the wave in the geometry space is irrelevant).

Every occurrence of an **IE** for **m** (> 0) τ wave(s) (τ^{y1}, τ^{y2},...,τ^{ym}) determines **S** for **C1** = {C_{y1}, C_{y2}, ..., C_{ym}}; where **C1** $\subseteq$ **C**. For example, Figure 7 shows the **RG** for **C** = {C_5, C_{19}} for the Gear Reducer assembly in Figure 3. An **IE** occurs at C_6: τ_1 of $C_5 \cap \tau_1$ of $C_{19} \cap \beta_6$ at C_6 which determines S_1 = {C_1, C_2, C_3, C_4, C_5, C_6, C_{19}}, S_2 = {C_{23}, C_{22}, C_{21}, C_{20}, C_{19}, C_6, C_5}. Similarly, all other **IE**'s and the corresponding sequences are determined to evaluate an **OS** for **C**.

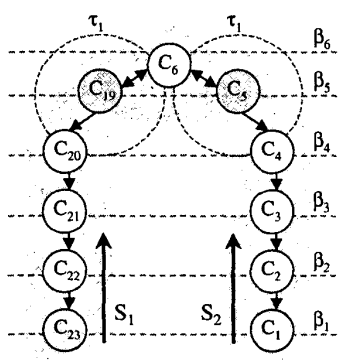

Figure 7. **RG** for C = {C_5, C_{19}} S_1 = {C_1, C_2, C_3, C_4, C_5, C_6, C_{19}}, S_2 = {C_{23}, C_{22}, C_{21}, C_{20}, C_{19}, C_6, C_5}

LARGE SELECTIVE DISASSEMBLY

A **MWP** algorithm is computationally feasible for **s** (<<**n**). However, for **S** (→ **n**) components, it is computationally prohibitive, which provides motivation for a new algorithm for **S-SD**.

In Srinivasan and Gadh [12], an algorithm called *Priority Intersection Event* (**PIE**) algorithm has been introduced for multiple components disassembly. The **PIE** algorithm defines prioritized intersection events between τ and β waves to determine locally minimum component removal sequences. The τ and β wave propagation's are as illustrated in Figure 6, however, for every time step **T** the τ wave advances by one wavefront and β wave by **T** wavefronts. For **S-SD**, let $\tau^x = \tau$ wave of C_x; $\tau^x_a = a^{th}$ wavefront of τ^x wave; $\beta_a = a^{th}$ wavefront of β wave; $\beta^m = \beta_1 \cap \beta_2 \ldots \cap \beta_{(m * T)}$; where 1<=**m**<=**S**. For example, a priority event is as follows: Let **m** = number of τ waves intersecting $C_w \in$ **A**. An **IE** between the **m** τ waves and β^m at C_w with the number of components in the sequence being less than the corresponding disjoint sequence is a priority event. Figure 8 shows the **RG** for the Gear Reducer assembly in Figure 3 with **C** = {C_2, C_3, C_4} at **T** = 2. At C_2, there are five **IE**'s ($\tau^2 \cap \tau^3 \cap \tau^4 \cap \beta^3$; $\tau^4 \cap \beta^1$; $\tau^2 \cap \tau^3 \cap \beta^2$; $\tau^2 \cap \tau^4 \cap \beta^2$; $\tau^3 \cap \tau^4 \cap \beta^2$). However, a priority event at C_2 is $\tau^2 \cap \tau^3 \cap \tau^4 \cap \beta^3$ determines a common sequence with locally minimum component removals.

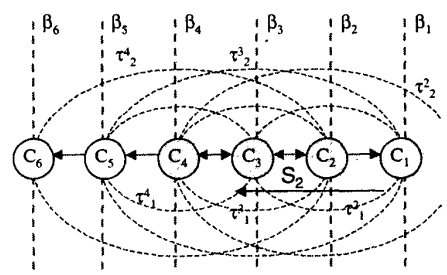

Figure 8. **RG** at **T** = 2 for **C** = {C_2, C_3, C_4}; **IE** at C_2, S_2 = {C_1, C_2, C_3, C_4}

Different classes of priority events, their characteristics, and evaluation of an **OS** based on the order of priority events are defined [12]. For the Gear Reducer assembly in Figure 3 with **C** = {C_2, C_3, C_4, C_{12}, C_{13}, C_{20}, C_{21}, C_{22}} an **OS** from **PIE** algorithm is {C_1, C_2, C_3, C_4, C_{23}, C_{22}, C_{21}, C_{20}, C_{14}, C_{13}, C_{12}}.

DISCUSSION

This section presents some of the attributes and preliminary implementation results of **SD** algorithms, and discusses the applicability of these algorithms for **SD** planning.

Attributes of WP algorithms in SD Planning

Some of the attributes of the **SWP** algorithm are:

- The wave denotes the disassembly order and minimal waves denote a minimal component removal sequence.
- The average computational complexity of determining an **OS** is O(n^2) for a **1**-dependent solution [9].

- The number of components analyzed ranges from 1 to **n**, depending on the geometry of C_x in **A**. If C_x is closer to the boundary, then the number of components analyzed is significantly less than **n**.

Some of the primary attributes of **MWP** algorithm are:

- An **IE** for **m**=1 determines a disjoint sequence for $C_{y1} \in$ **C**. However, for **m**>1, **IE** determines a common sequence to disassemble all the **m** components $C_{y1}, C_{y2}, \ldots, C_{ym} \in$ **C**.
- **IE** occurrence for **C1** depends on the geometric configuration of **A**, i.e., if in **A**, there is no **IE** of **m** (> 0) τ wave(s) $(\tau^{y1}, \tau^{y2}, \ldots, \tau^{ym})$ with a β wave at $C_w \in$ **A**, then there exist no common sequence **S** for $\{C_{y1}, C_{y2}, \ldots, C_{ym}\}$ of type $\{C_b \overset{P}{\sim} C_w, C_w \overset{P}{\sim} C_{y1}, \ldots, C_w \overset{P}{\sim} C_{ym}\}$.
- The computational complexity of the **MWP** approach is of order $O(n^2 2^s)$ [10]. For **s** << **n**, this algorithm is computationally feasible, and is efficient compared to the enumeration approach of computational complexity $O(2^n)$.

Some of the primary attributes of **PIE** algorithm are:

- The priority events are necessary candidate events in determining an **OS** and number of such events is polynomial in number [12].
- The order of event occurrence in **A** depends on the geometric configuration of components in **A** and every **IE** for **m** (> 0) τ wave(s) $(\tau^{y1}, \tau^{y2}, \ldots, \tau^{ym})$, with a β^m wave, determines **S** for **C** = $\{C_{y1}, C_{y2}, \ldots, C_{ym}\}$ with locally minimum component removals; where **C1** $\subseteq$ **C**.
- The computational complexity of **PIE** algorithm in determining an **OS** is $O(Sn^2)$ [12].

The **SD** algorithms presented for **1-SD**, **s-SD** and **S-SD** problems have some attributes in common:

- A wave provides topological disassembly ordering based on the geometric attributes, such as accessibility of components.
- A wave embeds the disassemblability of components in previous wavefronts; i.e., once a component in one wavefront is disassembled, its adjacent components in the previous wavefront become disassemblable.
- A **SD** solution with minimal wavefronts gives a locally optimum solution.

Preliminary Implementation Results

Some preliminary implementation results of **SD** algorithms in a prototypical system **A3D** (Assembly Disassembly in Three Dimensions) are presented below.

Figures 9b and 9c show the result for single component disassembly. For the Engine sub-assembly shown in Figure 9a with **C** = $\{C_3\}$ there are two optimal solutions: Figure 9b shows the **SD** with **OS** = $\{C_1, C_2, C_3\}$ and Figure 9c shows the **SD** with **OS** = $\{C_6, C_5, C_4\}$.

Figure 10b shows the result of disjoint **SD** sequences obtained from **SWP** algorithm. For the Cab sub-assembly shown in Figure 10a with **C** = $\{C_2, C_3\}$; **S** = $\{C_1, C_2, C_4, C_3\}$. The resultant **S** is obtained from two disjoint sequences: **OS** = $\{C_1, C_2\}$ for **C1** = $\{C_2\}$ and **OS** = $\{C_4, C_3\}$ for **C2** = $\{C_3\}$.

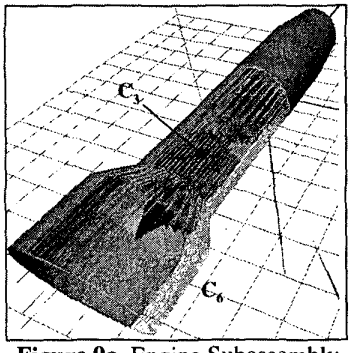

Figure 9a. Engine Subassembly

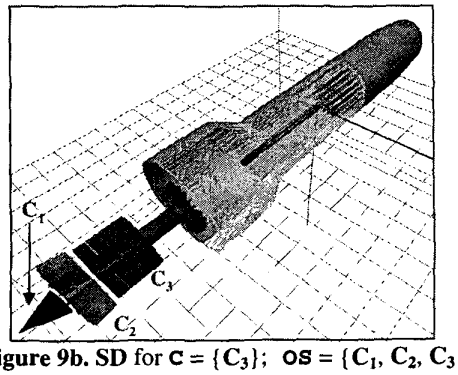

Figure 9b. **SD** for **C** = $\{C_3\}$; **OS** = $\{C_1, C_2, C_3\}$

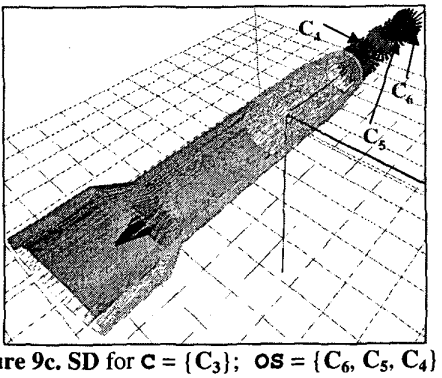

Figure 9c. **SD** for **C** = $\{C_3\}$; **OS** = $\{C_6, C_5, C_4\}$

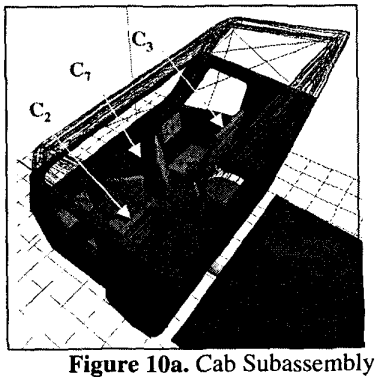

Figure 10a. Cab Subassembly

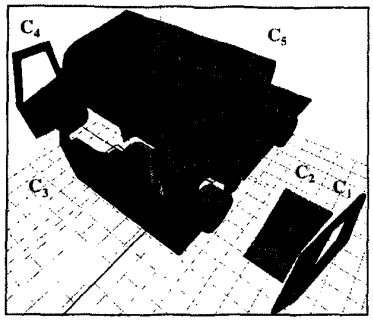

Figure 10b. SD for $\mathbf{C} = \{C_2, C_3\}$; $\mathbf{S} = \{C_1, C_2, C_4, C_3\}$

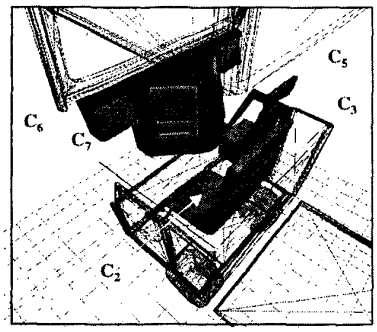

Figure 10c. $\mathbf{C} = \{C_2, C_3, C_7\}$; $\mathbf{OS} = \{C_5, C_6, C_2, C_3, C_7\}$

Figure 10c shows the result generated from a common **SD** sequence. For $\mathbf{C} = \{C_2, C_3, C_7\}$ in Figure 10a, the resultant $\mathbf{OS} = \{C_5, C_6, C_2, C_3, C_7\}$. Similarly, if $\mathbf{C} = \{C_2, C_3\}$ in Figure 10a, the resultant $\mathbf{S}$ by applying a multiple **SD** algorithm would be $\mathbf{OS} = \{C_5, C_2, C_3\}$, which is better than the disjoint sequence shown in Figure 10b.

WP Algorithms: A Comparison

	Number of Removals			*Computation Time*		
	SWP	MWP	PIE	SWP	MWP	PIE
1-SD	® ✓	✓	✓	® ✓	↑	↑
s-SD	↑	® ✓	↑	↓	® ✓	↓
S-SD	↑	↓	® ✓	↓	↑	® ✓

Legend ✓ = Feasible (lower order complexity and a solution with minimal removals)

↓ = Decreases ↑ = Increases ® = Reference

Figure 11. Comparison of algorithms for **SD**

In specific, the **SWP** algorithm analyzes a subset of components; **MWP** algorithm determines all common and disjoint sequences between the target components, and **PIE** algorithm determines necessary candidate sequences for a locally optimum solution. Both **SWP** and **PIE** algorithm have polynomial complexity, and **MWP** algorithm has a lower-order exponential complexity. A comparison of **SD** algorithms for different **SD** problems is shown in Figure 11. For **1-SD**, **s-SD** and **S-SD** problems, **SWP**, **MWP** and **PIE** algorithms are taken as references for comparison with respect to other algorithms.

REFERENCES

[1] D.M. Anderson. *Design for manufacturability – optimizing cost, quality and time-to-market.* CIM Press, 1990.

[2] D.F. Baldwin, T.E. Abel, M.C.M. Lui, T.L. De-Fazio, and D.E. Whitney. An integrated computer aid for generating and evaluating assembly sequences for mechanical products. *IEEE Transactions on Robotics and Automation*, 7(1):78-94, 1991.

[3] G. Boothroyd, P. Dewhurst, and W. Knight. *Product design for manufacturing and assembly.* Marcel Dekker Inc., 1994.

[4] T.E. Graedel and B.R. Allenby. *Design for Environment.* Prentice Hall, 1996.

[5] D. Halperin, J.C. Latombe, and R.H. Wilson. A General framework for assembly planning-the motion space approach. In *Proc. of the ACM Computational Geometry*, pages 9-15, 1998.

[6] L.S. Homem de Mello and S. Lee. *Computer aided mechanical assembly planning.* Kluwer Academic Publishers, 1991.

[7] R. Mattikalli and P. Khosla. Motion constrains from contact geometry: representation and analysis. In *Proc. IEEE Int. Conference on Robotics and Automation*, pages 2178-2185, 1992.

[8] B. Romney, C. Godard, M. Goldwasser, and G. Ramkumar. An efficient system for geometric assembly sequence generation and evaluation. *ASME Design Engineering Technical Conference,* pages 699-712, 1995.

[9] H. Srinivasan and R. Gadh. A geometric algorithm for single selective disassembly using the wave propagation abstraction. *Journal of Computer Aided Design*, 30(8): 603-613, 1998.

[10] H. Srinivasan and R. Gadh. Efficient disassembly of multiple components from an assembly using wave propagation. Submitted for publication in the *ASME Journal of Mechanical Design*, March 1998.

[11] H. Srinivasan, R. Figueroa, and R. Gadh. Selective disassembly for virtual prototyping as applied to de-manufacturing. *Journal of Robotics and Computer Integrated Manufacturing*, Accepted July 1998.

[12] H. Srinivasan and R. Gadh. Complexity reduction in geometric selective disassembly using the wave propagation abstraction. In *Proc. IEEE Int. Conference on Robotics and Automation*, pages 1478-1483, 1998.

[13] R.H. Wilson, L. Kavraki, T. Lozano-Perez, and J.C. Latombe. Two-handed assembly sequencing. *Int. Journal of Robotics research*, 14(4): 335-350, 1995.

[14] J.D. Wolter. A combinatorial analysis of enumerative data structures for assembly planning. *Journal of Design and Manufacturing*, 2(2): 93-104, 1992.

[15] T.C. Woo and D. Dutta. Automatic disassembly and total ordering in three dimensions. *ASME Journal of Engineering for Industry*, 113(1): 207-213, 1991.