Chapter 2

DESIGN FOR PRODUCT EMBEDDED DISASSEMBLY

Shingo Takeuchi[1] and Kazuhiro Saitou[1]

[1]*Department of Mechanical Engineering, University of Michigan, Ann Arbor, Michigan 48109-2125, USA*

Abstract This chapter discusses an application of multi-objective genetic algorithm for designing products with a built-in disassembly means that can be triggered by the removal of one or a few fasteners at the end of the product lives. Given component geometries, the method simultaneously determines the spatial configuration of components, locators and fasteners, and the end-of-life (EOL) treatments of components and subassemblies, such that the product can be disassembled for the maxim profit and minimum environmental impact through recycling and reuse via domino-like "self-disassembly" process. A multi-objective genetic algorithm is utilized to search for Pareto optimal designs in terms of 1) satisfaction of the distance specification among components, 2) efficient use of locators on components, 3) profit of EOL scenario, and 4) environmental impact of EOL scenario. The method is applied to a simplified model of the Power Mac G4 cube® for demonstration.

Keywords: Design for disassembly, environmentally-conscious design, design optimization, multi-objective genetic algorithm

1. INTRODUCTION

Increased regulatory pressures (e.g., EU's WEEE directive) and voluntary initiatives have placed manufacturers more responsible for end-of-life (EOL) treatments such as material recycling and component reuse. Since both recycling and reuse typically require disassembly, design for disassembly (DFD) has become a key design issue in almost any mass-produced product. DFD is particularly critical in consumer electronic products due to the large amount of production and short cycle time for technological obsolescence. Also, components

S. Takeuchi and K. Saitou: *Design for Product Embedded Disassembly*, Studies in Computational Intelligence (SCI) **88**, 9–39 (2008)
www.springerlink.com

in these products are typically required to fit into a tight enclosing space, which makes disassembly even more challenging.

Economic feasibility of an end-of-life (EOL) scenario of a product is determined by the interaction among disassembly cost, revenue from the EOL treatments of the disassembled components, and the regulatory requirements on products, components and materials. While meeting regulatory requirements is obligatory regardless of economic feasibility, EOL decision making is often governed by economical considerations (Chen et al., 1993). Even if a component has high recycling/reuse value or high environmental impact, for instance, it may not be economically justifiable to retrieve it if doing so requires excessive disassembly cost. Since the cost of manual disassembly depends largely on the number of fasteners to be removed and of components to be reached, grabbed, and handled during disassembly, it is highly desirable to locate such high-valued or high-impact components within a product enclosure, such that they can be retrieved by removing less fasteners and components.

The above thoughts motivated us to develop a concept of product-embedded disassembly, where the relative motions of components are constrained by locator features (such as catches and lugs) integral to components, in such a way that the optimal disassembly sequence is realized via a domino-like "self-disassembly" process triggered by the removal of one or a few fasteners.

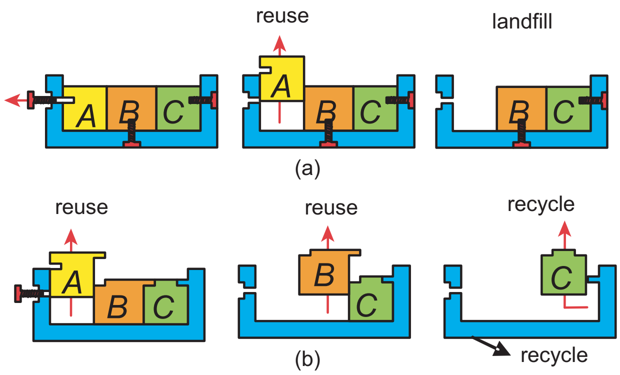
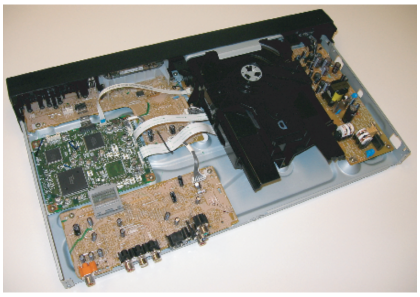


Figure 2-1. (a) Conventional assembly (b) assembly with embedded disassembly.

Fig. 2-1 illustrates the concept of product-embedded disassembly as compare to the conventional disassembly. In the conventional assembly (Fig. 2-1a), components A, B and C are fixed with three fasteners. With high labor cost for removing fasteners (as often the case in developed countries), only

A may be economically disassembled and reused, with the remainder sent to a landfill. This end-of-life (EOL) scenario (*i.e.*, disassemble and reuse A, and landfill the remainder) is obviously not ideal from either economical or environmental viewpoints. In the assembly with embedded disassembly (Fig. 2-1b), on the other hand, the motions of B and C are constrained by the locators on components. As such, the removal of the fastener to A (called a trigger fastener) activates the domino-like self-disassembly pathway $\mathrm{A} \rightarrow \mathrm{B} \rightarrow \mathrm{C}$. Since no additional fasteners need to be removed, B and C can also be disassembled, allowing the recycle/reuse of all components and the case. This EOL scenario (*i.e.*, disassemble all components, reuse A and B, and recycle C and the case) is economically and environmentally far better than the one for the conventional assembly.

(a)



(b)

Figure 2-2. Example products suited for product-embedded disassembly, (a) desktop computer (b) DVD player.

The concept of product-embedded disassembly can be applied to a wide variety of products, since it requires no special tools, materials, or actuators to implement. It is particularly well suited for electrical products assembled of functionally modularized components, whose spatial configurations within the enclosure have some flexibility. Fig. 2-2 shows examples of such products. A desktop computer in Fig. 2-2a is assembled of functionally distinct components such as a motherboard, a hard drive, and a power unit, arranged to fit within a tight enclosure. The components are, however, not completely packed due to the need of the air passage for cooling and the accessibility for upgrade and repair. Thanks to this extra space and electrical connections among components, the spatial configurations of the components have a certain degree of flexibility. A DVD player in Fig. 2-2b shows even roomier component arrangements, due to the consumers' tendency to prefer large sizes in home theater appliances. Since designing products with a single "disassembly button" may cause safety concerns, the method will, in practice, be best utilized as an inspiration to the designer during the early stage configuration design and critical components can be independently fastened with a secure, conventional means.

The concept, however, may be unsuitable to the products that allow very little freedom in component arrangements. Examples include mobile IT products such as cell phones, laptop computers, and MP3 players, due to their extremely tight packaging requirements and mostly layer-by-layer assembly.

This chapter discusses an application of a multi-objective genetic algorithm for designing products that optimally embody the above concept of product-embedded disassembly (Takeuchi and Saitou, 2005; Takeuchi and Saitou, 2006). Given component geometries, the method simultaneously determines the spatial configuration of components, locators and fasteners, and the end-of-life (EOL) treatments of components and subassemblies, such that the product can be disassembled for the maximum profit and minimum environmental impact through recycling and reuse via a domino-like "self-disassembly" process. A multi-objective genetic algorithm (Fonseca and Fleming, 1993; Deb et al., 2002) is utilized to search for Pareto optimal designs in terms of 1) satisfaction of the distance specification among components, 2) efficient use of locators on components, 3) profit of EOL scenario, and 4) environmental impact of EOL scenario. The method is applied to a simplified model of Power Mac G4 cube® for demonstration.

2. RELATED WORK

2.1 Design for Disassembly

Design for disassembly (DFD) is a class of design methods and guidelines to enhance the ease of disassembly for product maintenance and/or EOL treatments (Boothroyd and Alting, 1992). Kroll et al. (1996) utilized disassembly evaluation charts to facilitate the improvements of product design. Das et al. (2000) introduced the Disassembly Effort Index (DEI) score to evaluate the ease of disassembly. Reap and Bras (2002) reported DFD guidelines for robotic semi-destructive disassembly, where detachable or breakable snap fits are preferred to screws due to their ease of disengagement. O'Shea et al. (1999) focused on tool selection during disassembly where the optimal tool selection path, in terms of the ease of disassembly, is produced via dynamic programming. Recently, Desai and Mital (2003) developed a scoring system, where factors associated with disassembly time such as disassembly force, tool requirements, and accessibility of fasteners are considered. Sodhi et al. (2004) focused on the impact of unfastening actions on disassembly cost and constructed U-effort model that helps designers to select fasteners for easy disassembly. Matsui et al. (1999) proposed the concept of Product Embedded Disassembly Process, where a means of part separation that can be activated upon disassembly is embedded within a product. As an example, they developed cathode-ray tube (CRT) with a Nichrome wire embedded along the desired separation line, which can induce thermal stress to crack the glass of the CRT tube upon the application of current.

While these works suggest locally redesigning an existing assembly for improving the ease of its disassembly, they do not address the simultaneous decisions of the spatial configuration of components and joints for improving the entire disassembly processes.

2.2 Disassembly Sequence Planning

Disassembly Sequence Planning (DSP) aims at generating feasible disassembly sequences for a given assembly, where the feasibility of a disassembly sequence is checked by the existence of collision-free motions to disassemble each component or subassembly in the sequence. Since the disassembly sequence generation problem is NP-complete, the past research has focused on efficient heuristic algorithms to approximately solve the problem. Based on a number of important research results on assembly sequence planning (Homem dé Mello and Sanderson, 1990; De Fazio and Whitney, 1987; Lee and Shin, 1990; Homem dé Mello and Sanderson,. 1991; Baldwin, et al., 1992), several automated disassembly sequence generation approaches for 2/2.5D components have been developed (Woo and Dutta, 1991; Dutta and Woo, 1995; Chen et al., 1997; Srinivasan and Gadh, 2000; Kaufman et al., 1996). More recent work is geared towards DSP with special attention to reuse, recycling, remanufacturing, and maintenance. Lambert (1999) built a linear programming model to obtain the optimal EOL disassembly. Li et al. (2002) used Genetic Algorithm (GA) combined with Tabu search (Glover, 1974; Glover 1986) to find the optimal disassembly sequence for maintenance.

This previous work, however, only addresses the generation and optimization of disassembly sequences for an assembly with a pre-specified spatial configuration of components. Since the feasibility of disassembly sequences largely depends on the spatial configuration of components, this would seriously limit the opportunities for optimizing an entire assembly. In addition, these works do not address the design of joint configurations, which also have a profound impact on the feasibility and quality of a disassembly sequence.

2.3 Configuration Design Problem

While rarely discussed in the context of disassembly, the design of the spatial configuration of given shapes have been an active research area by itself. Among the most popular flavors is the bin packing problem (BPP), where the total volume (or area for 2D problems) a configuration occupies is to be minimized. Since this problem is also NP-complete, heuristic methods are commonly used. Fujita et al. (1996) proposed hybrid approaches for a 2D plant layout problem, where the topological neighboring relationships of a layout are determined by Simulated Annealing (SA), whereas the generalized reduced gradient (GRG) method determines the geometry. Kolli et al. (1996) used SA for packing 3D components with arbitrary geometry. GA is also widely used for the configuration design problem. Corcoran and Wainwright (1992) solved a 3D packing problem with GA using multiple crossover methods. Jain and Gea (1998) adopted discrete representation as an object expression and proposed a geometry-based crossover operation for a 2D packing problem. Grignon and Fadel (1999) proposed a configuration design optimization method by using multi-objective GA, where static and dynamic balances and maintainability are considered in addition to configuration volume.

These works, however, do not address the integration with DSP.

2.4 Life Cycle Assessment

Life Cycle Assessment (LCA) has been widely used as a tool to estimate the environmental impact of an EOL scenario of various products (Caudill et al., 2002; Rose and Stevels, 2001) including computers (Williams and Sasaki, 2003; Aanstoos et al., 1998; Kuehr and Williams, 2003). Since the optimal EOL scenario should be economically feasible as well as environmentally sound, LCA is often integrated with cost analysis. Goggin and Browne (2000) constructed a model for determining the recovery of a product, components and materials, where EOL scenarios are evaluated from economical and environmental perspectives. Kuo and Hsin-Hung (2005) integrated LCA into Quality Function Development (QFD) to achieve the best balance between customer satisfaction and environmental impact. In our previous work (Hula et al., 2003), we compared the optimal EOL scenarios of a coffee maker in Aachen, Germany and in Ann Arbor, MI, and concluded the optimal EOL scenario varied greatly depending on the local recycling/reuse infrastructures and regulatory requirements.

This work, however, merely addressed the evaluation and optimization of the environmental impact of a given product, and did not address the design of component, locator, and fastener configuration as addressed in this paper.

3. METHOD

The method can be summarized as the following optimization problem:

- **Given**: geometries, weights, materials, and recycle and reuse values of each component, contact and distance specifications among components, locator library, and possible EOL treatments and associated scenarios.
- **Find**: spatial configuration of components and locators, EOL treatments of disassembled components and subassemblies.
- **Subject to**: no overlap among components, no unfixed components prior to disassembly, satisfaction of contact specifications, assemble-ability of components.
- **Minimizing**: violation of distance specification, redundant use of locators, and environmental impact of EOL scenario.
- **Maximizing**: profit of EOL scenario.

Since the optimization problem has four objectives, a multi-objective genetic algorithm (MOGA) (Fonseca and Fleming, 1993; Deb et al., 2002) is utilized to obtain Pareto optimal solutions.

3.1 Inputs

There are four (4) categories of inputs for the problem as listed below:

- **Component information**: This includes the geometries, weights, materials and reuse values of components. Due to the efficiency in checking contacts and the simplicity in modifying geometries (Beasley and Martin, 1993; Minami et al., 1995; Sung et al., 2001), the component geometries are represented by voxels. CAD inputs are first voxelized using ACIS® solid modeling kernel.
- **Contact and distance specifications**: The adjacencies and distances among components are often constrained by their functional relationships. For example, a heat sink and CPU in a computer should be in contact, and a cooling fan and CPU should be nearby. The contact specification specifies the required adjacencies among the component, such as CPU and a heat sink in a computer. Since the distances between some pairs of components are more important than the others, the distance specification is defined as a set of the weights of importance for the distances between pairs of components (measured between two designated voxels) that need to be minimized. If the weight between two components is not defined, their distance is considered unimportant and can be arbitrarily chosen. Fig. 2-3 shows an example.

- **Locator library**: Since the types of feasible locators depend on manufacturing and assembly processes, they are pre-specified by a designer as a locator library. It is a set of locators for a specific application domain, which can be potentially added on each component to constrain its motion. Fig. 2-4 shows schematics of locators commonly found on sheet metal or injection-molded components in computer assemblies (Bonenberger, 2000), which are also used in the following case study. Note that screws are regarded as a special type of locators, and a slot can only be used with two circuit boards. Locator constraints (LC) shown in the third column of Fig. 2-4 illustrates the set of directions locators are constrained to when they are oriented as shown in the second column, formally represented as a subset of $\{-x, +x, -y, +y, -z, +z\}$.
- **Possible EOL treatments and scenarios**: An EOL scenario is a sequence of events, such as disassembly, cleaning, and refurbishing, before a component receives an EOL treatment such as recycle and reuse. The EOL treatments available to each component and the associated scenarios leading to each treatment must be given as input. Fig. 2-5 shows an example of EOL treatments (reuse, recycle, or landfill) and the associated EOL scenarios represented as a flow chart.

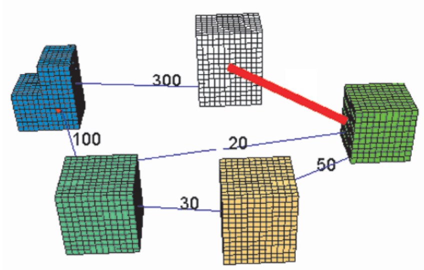


Figure 2-3. Example of contact specification (thick line) and distance specification (thin lines). Labels on thin lines indicate relative importance of minimizing distances.

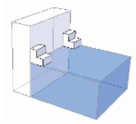
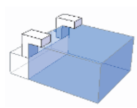
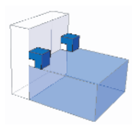
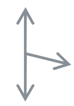
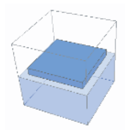
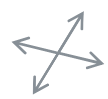
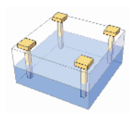
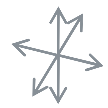
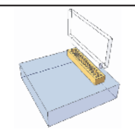
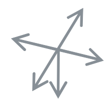
Type	Schematic	*LC*
(a) Catch		
(b) Lug		
(c) Track		
(d) Boss		
(e) Screw		
(f) Slot		

Figure 2-4. Graphical representation of typical locators for sheet metal or injection-molded components (Bonenberger, 2000).

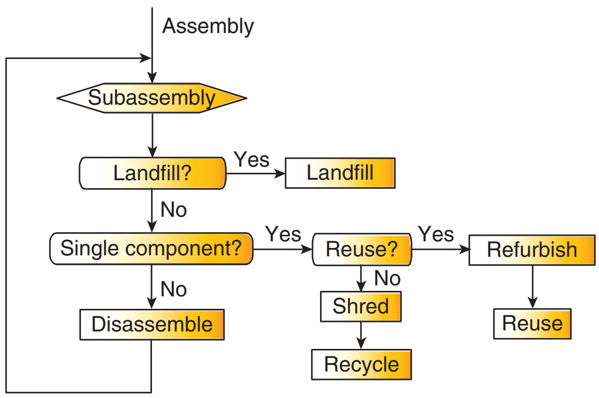


Figure 2-5. Flow chart of example EOL scenario.

3.2 Design Variables

There are three (3) design variables for the problem. The first design variable, *configuration vector*, represents the spatial configuration and dimensional change of each component:

$$\boldsymbol{x} = (x_0, x_1, \ldots., x_{n-1}) \tag{1}$$

$$x_i = (\boldsymbol{t}_i, \boldsymbol{r}_i, \boldsymbol{d}_i,);\ i = 0, 1, \ldots n-1 \tag{2}$$

$$\boldsymbol{t}_i \in \{0, \pm c, \pm 2c, \pm 3c, \ldots\}^3 \tag{3}$$

$$\boldsymbol{r}_i \in \{-90°, 0°, 90°, 180°\}^3 \tag{4}$$

$$\boldsymbol{d}_i \in \{0, \pm c, \pm 2c, \pm 3c, \ldots\}^f \tag{5}$$

where n is the number of components in the assembly, $\boldsymbol{t}_i$ and $\boldsymbol{r}_i$ are the vectors of the translational and rotational motions of component i with respect to the global reference frame, and $\boldsymbol{d}_i$ is a vector of the offset values of the f faces of component i in their normal directions, and c is the length of the sides of a voxel. Note that $\boldsymbol{d}_i$ is considered only for the components whose dimensions can be adjusted to allow the addition of certain locator features. For example, the components designed and manufactured in-house can have some flexibility in their dimensions, whereas off-the-shelf components cannot.

The second design variable, *locator vector*, represents the spatial configuration of the locators on each component:

$$\boldsymbol{y} = (\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_{m-1}) \tag{6}$$

$$\boldsymbol{y}_i = (CD_i, p_i);\ i = 0, \ldots, m-1 \tag{7}$$

where $m = n(n-1)/2$ is the number of pairs of components in the assembly, and $CD_i \subseteq \{-x, +x, -y, +y, -z, +z\}$ is a set of directions in which the motion of component c_0 in the i-th pair $(c_0,\ c_1)$ is to be constrained, and p_i is a sequence of locators indicating their priority during the construction of the locator configuration.

The choice of locator for the i-th component pair is indirectly represented by CD_i and $p_{\rm i}$, since the direct representation of the locator id in the library would result in a large number of infeasible choices. The construction of locator configurations from a given $\boldsymbol{y}_i$ is not trivial since 1) multiple locator types can constrain the motion of c_0 as specified by CD_i, and 2) among such locator types, geometrically feasible locators depend on the relative locations of components c_0 and c_1. Fig. 2-6 shows an example. In order to constrain the motion of c_0 in $+z$ direction, a catch can be added to c_1 if c_0 is "below" c_1 as shown in Fig. 2-6a. However, a catch cannot be used if c_0 is "above" c_1 as shown in Figs. 2-6b and 6c, in which case boss (Fig. 2-6b) or track (Fig. 2-6c) needs to be used. Thus, the locator configuration of a component is dynamically constructed by testing locator types in the sequence of p_i.

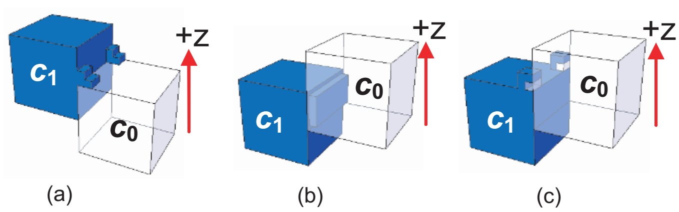


Figure 2-6. Construction of locator configuration.

Given $\boldsymbol{y}_i = (CD_i,\ p_i)$, the locator configuration of the i-th pair of components c_0 and c_1 is constructed by testing locator types, in sequence p_i, for constraining each direction in CD_i as follows:

1. For each $d \in CD_i$, remove d from CD_i if the motion of c_0 in $d \in CD$ is constrained by other components or locators. This step is necessary to reduce the redundant use of locator features.
2. Remove locator type t at the beginning of p_i. If p_i is empty, return FALSE.
3. Select direction $d \in CD$.
4. Find an orientation of o of locator type t whose locator constraint *LC* (after re-orientation) contains d. If several orientations are found, select an orientation with maximum $|LC \cap CD_i|$. If none is found, go to step 2.
5. Add t to c_0 or c_1 in o.
6. $CD_i \leftarrow CD_i \backslash LC$. If $CD = \emptyset$, return TRUE. Otherwise, go to step 3.

The above procedure returns TRUE if a locator configuration constraining all directions in CD_i is found by using the locator types in p_i, and FALSE otherwise. During optimization, the value of $\boldsymbol{y}_i$ returning FALSE is considered as infeasible.

Fig. 2-7 shows an example construction of locator configuration of components c_0 and c_1 according to the above procedure with $CD = \{+z\}$ and $p =$ <Catch, Screw, Lug, Track, Boss>:

- Step 1: Since component c_1 does not constrain the motion of c_0 in $+z$ (Fig. 2-7a), $+z$ remains in *CD*
- Step 2: Remove Catch from p. Since $p =$ <Screw, Lug, Track, Boss> is non-empty, proceed.
- Step 3: Select $+z$ from *CD*.
- Step 4: Systematically examine the possible orientations of Catch on c_0 and c_1 to find the orientations that constraint $+z$ (o_0 through o_7 in Fig. 2-7b and 7c). Note, however, that the orientations other than o_0 and o_5 in Fig. 2-7d are infeasible due to the lack of an adjacent component. Since both o_0 and o_5 has $|LC \cap CD_i| = |\{+z\} \cap \{+z\}| = |\{+z\}| = 1,\ o_0$ is chosen.
- Step 5: Catch in orientation o_0 is added to c_1 (Fig. 2-7e).
- Step 6: Since $CD_i \backslash LC = \{+z\} \backslash \{+z\} = \emptyset,\ CD_i = \emptyset$. Return TRUE.

Fig. 2-8 illustrates how two different values of priority sequence p with the same *CD* can result in the different locator configurations. For the two components in Fig. 2-8a with $CD = \{-x, +x, +\mathrm{z}\}$, sequence $p =$ <Track, Boss, Screw, Catch, Lug> results in the locators in Fig. 2-8b, whereas sequence $p =$ <Catch, Lug, Screw, Track, Boss> results in the locators in Fig. 2-8c. In Fig. 2-8c, two locator types, Catch and Lugs are used since Catch (top priority) cannot be oriented to constrain c_0 in +z direction while Lug (second priority) can.

While indirect, constraint direction *CD* and priority sequence p realizes a compact representation of a locator configuration of a pair of components. Compared to the direct representation in (Takeuchi and Saitou, 2005) that specifies the existence of a locator type in an orientation at a potential location on a component, it can generate far fewer infeasible locator configurations during the "generate and test" process of genetic algorithms. As a result, the computational efficiency is dramatically improved. Instead of treating the priority sequence as a design variable, one might imagine checking for locator types always in the (fixed) ascending sequence of their manufacturing costs is sufficient. However, such costs are difficult to determine *a priori*, since the actual geometry (and hence the cost) of a locator heavily dependents on the configuration of the surrounding components.

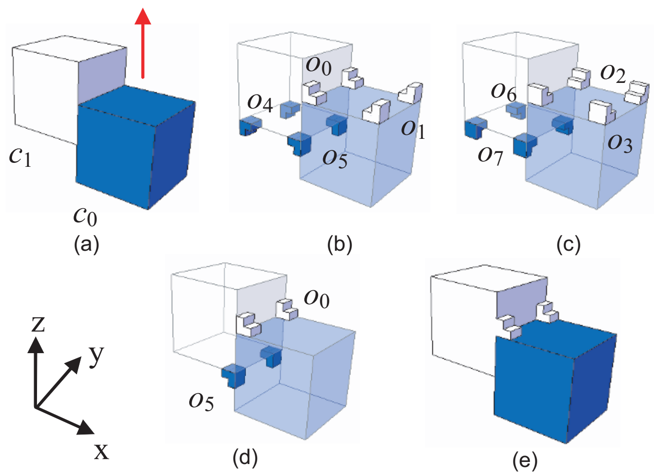


Figure 2-7. An example construction of locator configuration: (a) two components, (b) and (c) possible orientations of Catch, (d) two feasible orientations, and (e) final locator configuration.

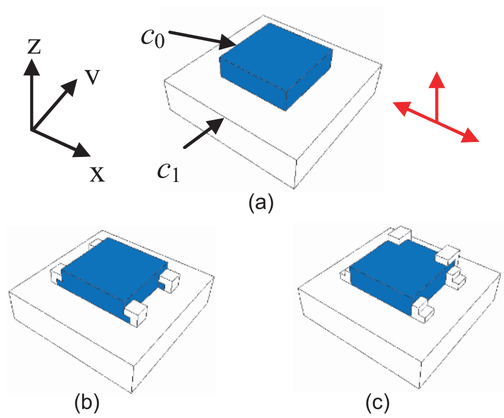


Figure 2-8. Influence of priority sequence p in locator configurations: (a) two components with $CD = \{-x, +x, +z\}$, (b) locators constructed with $p = $ <Track, Boss, Screw, Catch, Lug>, and (c) locators constructed with $p = $ <Catch, Lug, Screw, Track, Boss>.

The third design variable, *EOL vector*, represents the EOL treatments of components:

$$z = (z_0, z_1, \ldots, z_{n-1}); z_i \in E_i \tag{8}$$

where E_i is a set of feasible EOL treatments of component i. In the following case study, $E_i = \{\text{recycle, reuse, landfill}\}$ for all components.

3.3 Constraints

There are four (4) constraints for the problem:

1. No overlap among components.
2. Satisfaction of contact specification.
3. No unfixed components prior to disassembly.
4. Assemble-ability of components.

Since the constraints are all geometric in nature, the voxel representation of component geometry facilitates their efficient evaluation. Constraints 1-3 are checked solely based on the information in $\boldsymbol{x}$, since the locator configurations constructed from $\boldsymbol{y}$ generate no overlaps. For constraint 3, immobility of all possible subassemblies is examined. Constraint 4 is necessary to ensure all components, whether or not to be disassembled, can be assembled when the product is first put together. It requires the information from both $\boldsymbol{x}$ and $\boldsymbol{y}$. Since checking this constraint requires simulation of assembly motions (assumed as the reverse of disassembly motions), it is done as a part of the evaluation of disassembly cost needed for one of the objective functions.

3.4 Objective Functions

There are four (4) objective functions for the problem. The first objective function (to be minimized) is for the satisfaction of the distance specification, given as:

$$f_1(\boldsymbol{x}, \boldsymbol{y}) = \sum_i w_i d_i \tag{9}$$

where w_i is the weight of the importance of distance d_i between two designated voxels.

The second objective function (to be minimized) is for the efficient use of locators, given as:

$$f_2(\boldsymbol{x}, \boldsymbol{y}) = \sum_i mc_i \tag{10}$$

where mc_i is the manufacturing difficulty of the i-th locator in the assembly, which represents the increased difficulty in manufacturing components due to the addition of the i-th locator.

The third objective function (to be maximized) is the profit of the EOL scenario of the assembly specified by $\boldsymbol{x}$ and $\boldsymbol{y}$, given as:

$$f_3(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = \sum_{i=0}^{n-1} p_i(z_i) - c^*(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) \tag{11}$$

In Eq. 11, $p_i(z_i)$ is the profit of the i-th component from EOL treatment z_i, calculated by the EOL model described in the next section. Also in Eq. 11, $c^*(\boldsymbol{x},\ \boldsymbol{y},\ \boldsymbol{z})$ is the minimum disassembly cost the assembly under the EOL scenario required by $\boldsymbol{z}$:

$$c^*(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = \min_{s \in S_{xyz}} c(s) \tag{12}$$

where S_{xyz} is the set of the *partial and total* disassembly sequences of the assembly specified by $\boldsymbol{x}$ and $\boldsymbol{y}$, for retrieving the components with $z_i =$ reuse or recycle and the components with regulatory requirement, and $c(s)$ is the cost of disassembly sequence s. Since an assembly specified by $\boldsymbol{x}$ and $\boldsymbol{y}$ can be disassembled in multiple sequences, Eq. 12 computes the minimum cost over S_{xyz}, which contains all disassembly sequences feasible to $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$, and their subsequences. Set S_{xyz} is represented as AND/OR graph (Homem dé Mello and Sanderson, 1990) computed based on the 2-disassemblability criterion (Woo and Dutta, 1991; Beasley and Martin, 1993) (i.e. the component can be removed by up to two successive motions) as follows:

1. Push the assembly to stack Q and the AND/OR graph.
2. Pop a subassembly *sa* from Q.
3. If *sa* does not contains component with $z_i =$ reuse or recycle and components with regulatory retrieval requirements, go to step 5.
4. For each subassembly $sb \subset sa$ that does not contain any fixed components, check the 2-disassemblability of *sb* from *sa*. If *sb* is 2-disassemblable, add *sb* and $sc = sa \backslash sb$ to the AND/OR graph. If *sb* and/or *sc* are composed of multiple components, push them to Q.
5. If $Q = \emptyset$, return. Otherwise go to step 2.

where the 2-disassembleability of two subassemblies *sb* from *sa* is checked as follows. For efficiency, only translational motions are considered:

1. For each mating surfaces between *sb* and $sc = sa \backslash sb$ (including the ones of the locators), obtain a set of constrained directions as a subset of six possible translational directions $D = \{-x, +x, -y, +y, -z, +z\}$.
2. Compute constrained directions CD_{bc} between *sb* and *sb* as a union of all constrained directions obtained in step 1.
3. If $D \backslash CD_{bc} = \emptyset$, return FALSE.
4. If there exist a direction in $D \backslash CD_{bc}$ along which *sb* can be moved infinitely without a collision, return *TRUE* (*sb* is 1-disassembleable).
5. Select a direction d in $D \backslash CD_{bc}$. If all have been selected, return infeasible. Otherwise, go to the next step.
6. Move *sb* by unit length along d. If *sb* collides with other components, go to step 5.
7. If *sb* is 1-disassembleable at the current location, return TRUE (*sb* is 2-disassembleable). Otherwise, go to step 6.

Assuming manual handling, insertion and fastening as timed in (Boothroyd et al., 1994), $c(s)$ is estimated based on the motions of the components and the numbers and accessibilities of the removed screws at each disassembly step. The cost of the i-th disassembly step is given by:

$$c_i = \omega_0 \cdot dc_0 + \omega_1 \cdot dc_1 + \omega_2 \cdot dc_2 \tag{13}$$

where dc_0 and dc_1 are the number of orientation changes and the sum of the moved distances, respectively, of the disassembled component at the i-th disassembly step, dc_2 is the sum of accessibilities a_s of the removed screws, and ω_j is the weights. The accessibility a_s of a removed screw is given as (Takeuchi and Saitou, 2005):

$$a_s = 1.0 + \omega_a/(aa + 0.01) \tag{14}$$

where ω_a is weight and *aa* is the area of the mounting face of the screw accessible from the outside of the product in its normal direction.

The forth objective function (to be minimized) is the environmental impact of the EOL scenario:

$$f_4(\boldsymbol{z}) = \sum_i e_i(z_i) \tag{15}$$

where $e_i(z_i)$ is the environmental impact of i-th component according to the EOL scenario for treatment z_i. The value of $e_i(z_i)$ is estimated by the EOL model described in the next section.

3.5 EOL Model

The EOL model adopted in the following case study assumes the EOL scenarios in Fig. 2-5 for all components (reuse only for some components), and uses energy consumption as the indicator for environmental impact (Hula et al., 003). Accordingly, profit $p_i(z_i)$ in Eq. 11 is defined as:

$$p_i(z_i) = \begin{cases} r_i^{reuse} - c_i^{trans} - c_i^{refurb} & \text{if } z_i = \text{reuse} \\ r_i^{recycle} - c_i^{trans} - c_i^{shred} & \text{if } z_i = \text{recycle} \\ -c_i^{trans} - c_i^{landfill} & \text{if } z_i = \text{landfill} \end{cases} \tag{16}$$

where r_i^{reuse} and $r_i^{recycle}$ are the revenues from reuse and recycle, respectively, and c_i^{trans}, c_i^{refurb}, c_i^{shred} and $c_i^{landfill}$ are the cost for transportation, refurbishment, shredding, and landfill, respectively. Similarly, energy consumption $e_i(z_i)$ in Eq. 15 is defined as:

$$e_i(z_i) = \begin{cases} e_i^{reuse} + e_i^{trans} + e_i^{refurb} & \text{if } z_i = \text{reuse} \\ e_i^{recycle} + e_i^{trans} + e_i^{shred} & \text{if } z_i = \text{recycle} \\ e_i^{landfill} + e_i^{trans} & \text{if } z_i = \text{landfill} \end{cases} \tag{17}$$

where e_i^{reuse}, e_i^{trans}, $e_i^{recycle}$, e_i^{refurb}, e_i^{shred} and $e_i^{landfill}$ are the energy consumptions of reuse, transportation, recycle, refurbishment, shredding, and landfill, respectively.

Revenue from reuse r_i^{reuse} is the current market value of component i, if such markets exist. Energy consumption of reuse e_i^{reuse} is the negative of the energy recovered from reusing component i:

$$e_i^{reuse} = -\sum_j me_j^{intens} \cdot m_{ij} \tag{18}$$

where me_j^{intens} is the energy intensity of material j and m_{ij} is the weight of material j in component i. Reuse, if available, is usually the best EOL treatment for a component because of its high revenue and high energy recovery. The availability of the reuse option for a component, however, is infrastructure dependent, and even if available, the revenue from reuse can greatly fluctuate in the market and hence difficult to estimate *a priori*.

Revenue from recycle $r_i^{recycle}$ and energy consumption of recycle $e_i^{recycle}$ are also calculated based on the material composition of a component:

$$r_i^{recycle} = \sum_j mr_j^{recycle} \cdot m_{ij} \tag{19}$$

$$e_i^{recycle} = -\sum_j me_j^{recover} \cdot m_{ij} \tag{20}$$

where $mr_j^{recycle}$ and $me_j^{recover}$ are the material value and recovered energy of material j, respectively.

Since little data is available for the refurbishment of components, the cost for refurbishment is simply assumed as:

$$c_i^{refurb} = 0.5 \cdot r_i^{reuse} \tag{21}$$

Based on the data on desktop computers (Aanstoos et al., 1998), energy consumption for refurbishment e_i^{refurb} is estimated as:

$$e_i^{refurb} = 1.106 \cdot m_i \tag{22}$$

where m_i is the weight of the i-th component.

Cost and energy consumption of transportation c_i^{trans} and e_i^{trans} are estimated as (Hula et al., 2003):

$$c_i^{trans} = \Delta c_i^{trans} \cdot D_i \cdot m_i \tag{22}$$

$$e_i^{trans} = \Delta e_i^{trans} \cdot D_i \cdot m_i \tag{23}$$

where $\Delta c_i^{trans} = 2.07\text{e}-4$ $[\$/\text{kg}\cdot\text{km}]$, $\Delta e_i^{trans} = 1.17\text{e}-3$ $[\text{MJ/kg}\cdot\text{km}]$, and D_i is the travel distance. Similarly, costs and energy consumptions for shredding and landfill c_i^{shred}, $c_i^{landfill}$, e_i^{shred} and $e_i^{landfill}$ are calculated as (Hula et al., 2003):

$$c_i^{shred} = \Delta c_i^{shred} \cdot m_i \tag{24}$$

$$e_i^{shred} = \Delta e_i^{shred} \cdot m_i \tag{25}$$

$$c_i^{landfill} = \Delta c_i^{landfill} \cdot m_i \tag{26}$$

$$e_i^{landfill} = \Delta e_i^{landfill} \cdot m_i \tag{27}$$

where $\Delta c_i^{shred} = 0.12\,[\$/\mathrm{kg}\cdot\mathrm{km}]$, $\Delta e_i^{shred} = 1.0\,[\mathrm{MJ/kg}\cdot\mathrm{km}]$, $\Delta c_i^{landfill} = 0.02\,[\$/\mathrm{kg}\cdot\mathrm{km}]$ and $\Delta e_i^{landfill} = 20000\,[\mathrm{MJ/kg}\cdot\mathrm{km}]$.

3.6 Optimization Algorithm

Since the problem is essentially a "double loop" of two NP-complete problems (i.e. disassembly sequence planning within a 3D layout problem), it should be solved by a heuristic algorithm. Since design variables $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ are discrete ($\boldsymbol{x}$ is a discrete variable since geometry is represented as voxels) and there are four objectives, a multi-objective genetic algorithm (Fonseca and Flemming, 1993; Deb et al., 2002) is utilized to obtain Pareto optimal design alternatives. A multi-objective genetic algorithm is an extension of the conventional (single-objective) genetic algorithms that do not require multiple objectives to be aggregated to one value, for example, as a weighted sum. Instead of static aggregates such as a weighted sum, it dynamically determines an aggregate of multiple objective values of a solution based on its relative quality in the current population, typically as the degree to which the solution dominates others in the current population.

A chromosome, a representation of design variables in genetic algorithms, is a simple list of the 3 design variables:

$$c = (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) \tag{28}$$

Since the information in $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ are linked to the geometry of a candidate design, the conventional one point or multiple point crossover for linear chromosomes are ineffective in preserving high-quality building blocks. Accordingly, a geometry-based crossover operation based on (Jain and Gea, 1998) is adopted:

1. Randomly select a point in the bounding box of the assembly.
2. Cut two parent designs p_1 and p_2 with the three planes parallel to $x,\ y,\ z$ axes, and passing through the point selected in step 1, into eight pieces each (Fig. 2-9a).
3. Assemble two child designs c_1 and c_2 by alternately swapping the pieces of p_1 and p_2 (Fig. 2-9b).
4. Repair c_1 and c_2 by moving each component C to the child containing the larger volume (of the sliced piece) of C. If c_1 and c_2 contain the same volume, C is placed in the same way as the parent with the higher rank.
5. Add locators to c_1 and c_2 by checking which parent each pair of component is inherited from. If a child contains both components of a pair, the corresponding locator is added to the child. Otherwise, a locator is randomly added to either child.

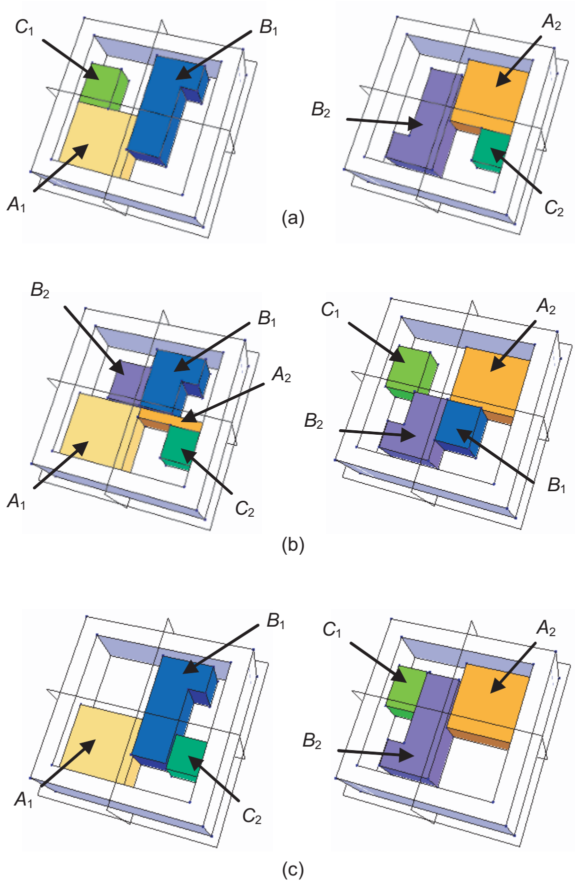


Figure 2-9. Geometry-based crossover operator. (a) two parents p_1 (left) and p_2 (right), (b) two children c_1 (left) and c_2 (right) after crossover, and (c) two children c_1 (left) and c_2 (right) after repair.

4. CASE STUDY

4.1 Problem

The method is applied to a model of Power Mac G4 Cube® manufactured by Apple Computer, Inc. (Fig. 2-10). Ten (10) major components are chosen based on the expected contribution to profit and environmental impact. Fig. 2-11a shows the ten components and their primary liaisons, and Fig. 2-11b shows the voxel representation of their simplified geometry and the contact (thick lines) and distance (thin lines with weights) specifications. The contacts between component B (heat sink) and C (CPU), and C (circuit board) and G (memory) are required due to their importance to the product function. Component A (case) is considered as fixed in the global reference frame. Component J (battery) needs to be retrieved due to regulatory requirements. The locator library in Fig. 2-4 is assumed for all components. The relative manufacturing difficulty of locators in the library is listed in Table 2-1.

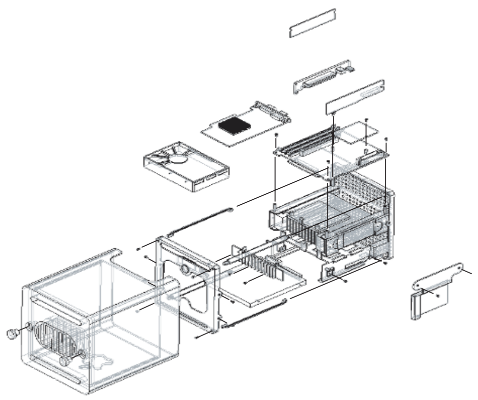
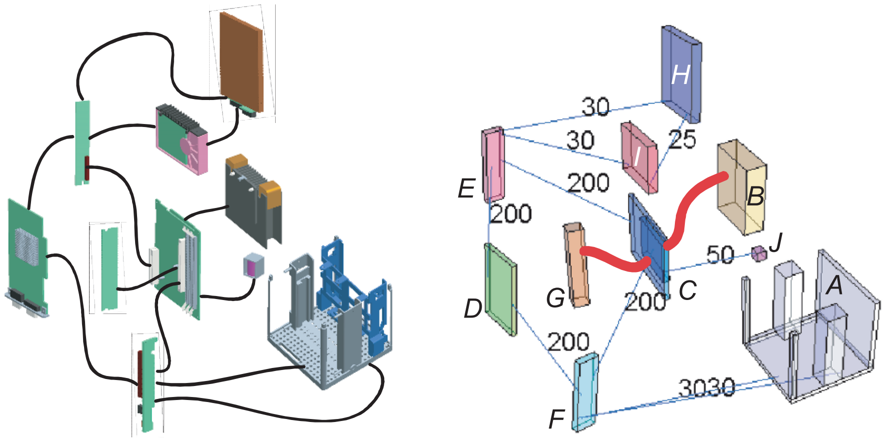


Figure 2-10. Assembly of Power Mac G4 Cube®.

Table 2-2 shows the material composition m_{ij} of components A–J in Fig. 2-11b. For components C–F, the material composition data in (Goosey and Kellner, 2003) is utilized. Table 2-3 shows energy intensity me_j^{intens}, recovered energy me_j^{recove}, and material values $mr_j^{recycle}$ (Kuehr and Williams, 2003; Hula, 2003). Considering Apple Computer's Electronic Recycling Program in United States and Canada (www.apple.com/environment), the EOL Power Mac G4 Cubes® are assumed to be transported to one of two facilities in United States (Worcester, MA and Gilroy, CA) for reuse, recycle, and landfill. The average distance between the collection point and the facility is estimated as $D_i = 1000\,\mathrm{km}$ for all components. It is assumed that 40 ton tracks are used for transportation. Based on this assumption, Table 2-4 shows the revenues, costs and energy consumptions of components A–J calculated using Eqs. 18–27. Revenue from reuse r_i^{reuse} reflects current values in the PC reuse markets in the United States (www.dvwarehouse.com and store.yahoo.com/hardcoremac/hardware.htm). Note that reuse option is not available to components A (frame) and B (heat sink).



(a) (b)

Figure 2-11. (a) Ten major components and their primary liaisons, and (b) contact and distance specifications.

Table 2-1. Relative manufacturing difficulty of the locators in the locator library in Fig. 2-4

Locator	**Lug**	**Track**	**Catch**	**Boss**	**Screw**	**Slot**
Mfg. difficulty	20	30	10	70	20	20

Table 2-2. Material composition [kg] of components A–J in Fig. 2-11.

Component	**Aluminum**	**Steel**	**Cupper**	**Gold**	**Silver**
***A* (frame)**	1.2	0	0	0	0
***B* (heat sink)**	0.6	0	0	0	0
***C* (circuit board)**	1.5e-2	0	4.8e-2	7.5e-5	3.0e-4
***D* (circuit board)**	1.0e-2	0	3.2e-2	5.0e-5	2.0e-4
***E* (circuit board)**	4.0e-3	0	1.3e-2	2.0e-5	8.0e-5
***F* (circuit board)**	5.0e-3	0	1.6e-2	2.5e-5	1.0e-4
***G* (memory)**	2.0e-3	0	6.4e-3	2.0e-5	4.0e-5
***H* (CD drive)**	0.25	0.25	0	0	0
***I* (HD drive)**	0.10	0.36	6.4e-3	1.0e-5	4.0e-5
***J* (battery)**	8.0e-5	0	1.4e-3	0	0

Component	**Tin**	**Lead**	**Cobalt**	**Lithium**	**Total**
A (frame)	0	0	0	0	1.2
B (heat sink)	0	0	0	0	0.60
C (circuit board)	9.0e-3	6.0e-3	0	0	0.30
D (circuit board)	6.0e-3	4.0e-3	0	0	0.20

E (circuit board)	2.4e-3	1.6e-3	0	0	8.0e-2
F (circuit board)	3.0e-3	2.0e-3	0	0	0.10
G (memory)	1.2e-3	8.0e-4	0	0	4.0e-2
H (CD drive)	0	0	0	0	0.50
I (HD drive)	1.2e-3	8.0e-4	0	0	0.50
J (battery)	0	0	3.3e-3	4.0e-3	2.0e-3

Table 2-3. Material information (Kuehr and Williams, 2003; Hula et al., 2003). Underlined values are estimations due to the lack of published data.

Material	me_j^{intens} **[MJ/kg]**	me_j^{recove} **[MJ/kg]**	$mr_j^{recycle}$ **[$/kg]**
Aluminum	2.1e2	1.4e2	0.98
Steel	59	19	0.22
Cupper	94	85	1.2
Gold	8.4e4	$\underline{7.5e4}$	1.7e4
Silver	1.6e3	$\underline{1.4e3}$	2.7e2
Tin	2.3e2	2.0e2	6.2
Lead	54	48	1.0
Cobalt	8.0e4	$\underline{6.0e4}$	38
Lithium	$\underline{1.5e3}$	$\underline{1.0e3}$	7.5

Table 2-4. revenue (r [$]), cost ($c$ [$]) and energy consumption (e [MJ]) of the major components A–J.

	$\boldsymbol{A}$	$\boldsymbol{B}$	$\boldsymbol{C}$	$\boldsymbol{D}$	$\boldsymbol{E}$
$\boldsymbol{r}_i^{reuse}$	N/A	N/A	3.5e2	80	1.3e2
$\boldsymbol{r}_i^{recycle}$	1.2	0.60	1.5	1.0	0.39
$\boldsymbol{c}_i^{trans}$	0.25	0.12	6.2e-2	4.1e-2	1.7e-2
$\boldsymbol{c}_i^{refurb}$	N/A	N/A	1.8e2	40	65
$\boldsymbol{c}_i^{shred}$	0.14	7.2e-2	3.6e-2	2.4e-2	9.6e-3
$\boldsymbol{c}_i^{landfill}$	2.4e-2	1.2e-2	6.0e-3	4.0e-3	1.6e-3
$\boldsymbol{e}_i^{reuse}$	$-2.6e2$	$-1.3e2$	-17	-12	-4.5
$\boldsymbol{e}_i^{trans}$	1.4	0.70	0.35	0.23	9.4e-2
$\boldsymbol{e}_i^{refurb}$	2.7	1.3	0.66	0.44	0.18
$\boldsymbol{e}_i^{rcycle}$	-170	-84	-14	-9.5	-3.8
$\boldsymbol{e}_i^{shred}$	1.2	0.60	0.30	0.20	8.0e-2
$\boldsymbol{e}_i^{landfill}$	2.4e4	1.2e4	6.0e3	4.0e3	1.6e3

	$\boldsymbol{F}$	$\boldsymbol{G}$	$\boldsymbol{H}$	$\boldsymbol{I}$	$\boldsymbol{J}$
$\boldsymbol{r}_i^{reuse}$	39	57	40	60	5.0

$\boldsymbol{r}_i^{recycle}$	0.49	0.36	0.30	0.37	0.12
$\boldsymbol{c}_i^{trans}$	2.1e-2	8.3e-3	0.10	0.10	4.1e-3
$\boldsymbol{c}_i^{refurb}$	20	29	20	30	2.5
$\boldsymbol{c}_i^{shred}$	1.2e-2	4.8e-3	6.0e-2	6.0e-2	2.4e-3
$\boldsymbol{c}_i^{landfill}$	2.0e-3	8.0e-4	1.0e-2	1.0e-2	4.0e-4
$\boldsymbol{e}_i^{reuse}$	-5.6	-3.1	-68	-45	-2.6e2
$\boldsymbol{e}_i^{trans}$	0.12	4.7e-2	0.59	0.59	2.3e-2
$\boldsymbol{e}_i^{refurb}$	0.22	8.8e-2	1.1	1.1	4.4e-2
$\boldsymbol{e}_i^{recycle}$	-4.8	-2.7	-40	-23	-2.0e2
$\boldsymbol{e}_i^{shred}$	0.10	4.0e-2	0.50	0.50	2.0e-2
$\boldsymbol{e}_i^{landfill}$	2.0e3	8.0e2	1.0e4	1.0e4	4.0e2

4.2 Results

After running the multi-objective genetic algorithm for approximately 240 hours (10 days) on a desktop PC with a 3.2 GHz CUP and a 2 GB RAM (number of population and generation are 100 and 300), thirty seven (37) Pareto optimal designs were obtained as design alternatives. Since the number of objective functions is four, the resulting 4-dimensional space is projected on to six 2-dimensional spaces in Fig. 2-12a–f. Fig. 2-13 shows five representative designs $R_1,\ R_2,\ R_3,\ R_4$ and R_5. Their objective function values are listed in Table 2-5 and also plotted on a bar chart in Fig. 2-14. As seen in Fig. 2-12, designs $R_1,\ R_2,\ R_3$, and R_4 are the best results only considering an objective function $f_1,\ f_2,\ f_3$ and f_4 regardless of the other objective function values, whereas R_5 is a balanced result in all four objectives.

The spatial configurations of R_3 and R_5 are quite similar, with noticeable differences in the EOL treatments. Figs. 2-15 and 2-16 show one of the optimal disassembly sequences of R_3 and R_5 with the EOL treatments of components, respectively. Design R_3 (design biased for profit) uses three screws, one of which is used between components A and B. Since components A and B have no reuse options, and recycling them is less economical than land-filling due to high labor cost for removing screws; they are not disassembled and simply discarded altogether for higher profit. On the other hand, components A and B are disassembled and recycled in R_5 (balanced design for all objectives) to reduce environmental impact at the expense of higher disassembly cost (lower profit).

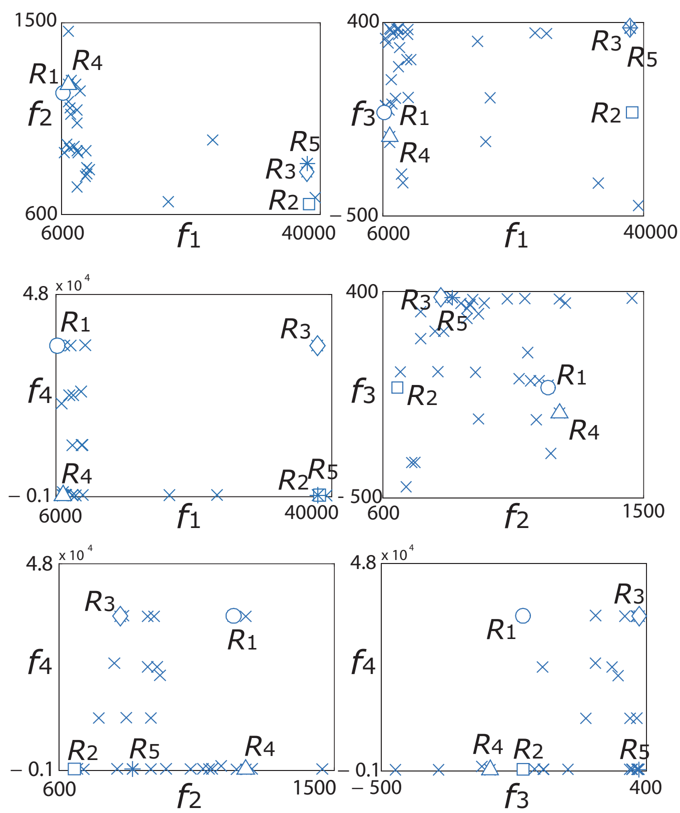


Figure 2-12. Distribution of Pareto optimal designs in six 2-dimensional spaces (a)–(f).

Table 2-5. Objective function values of R_1, R_2, R_3, R_4 and R_5

	F_1 **(dist. spec.)**	f_2 **(mfg. diff.)**	f_3 **(profit)**	f_4 **(env. impact)**
R_1	6175	1170	−19.30	35627
R_2	38496	650	−19.34	−642
R_3	38227	800	374.72	35593
R_4	6884	1210	−130.79	−741
R_5	38299	840	373.24	−647

As stated in the previous section, reuse, if available, is usually the best EOL treatment for a component because of its high revenue and high energy recovery. For the components without reuse option, the choice between recycle and landfill depends on the ease of disassembly, as seen in these results. If the disassembly cost is low enough that recycling the component is more profitable than land-filling it, recycle becomes the most profitable EOL treatment. Otherwise, there is a trade-off between the profit and the environmental impact, which is found in the Pareto optimal designs.

Oftentimes such trade-off among alternative designs can hint at opportunities for further design improvements. For example, the examination of the differences between R_3 and R_5 suggests the possibility of replacing the screws between A and B by slot-like locators (which are not available for A and B in the locator library) for higher profit and lower environmental impact.

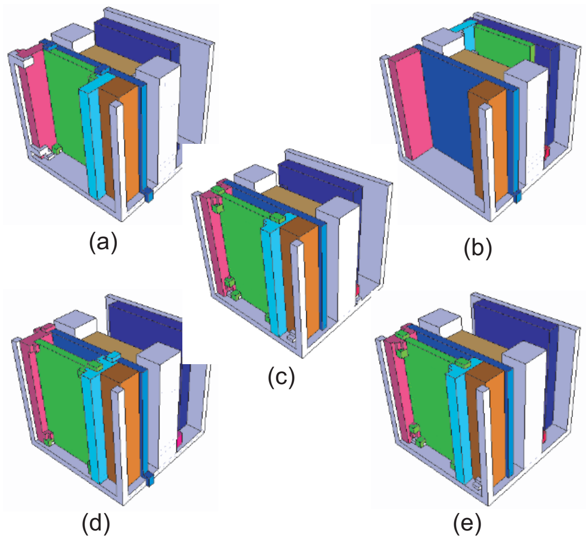


Figure 2-13. Representative Pareto designs: (a) R_1, (b) R_2, (c) R_3, (d) R_4 and (e) R_5.

5. SUMMARY AND FUTURE WORKS

This paper presented an extension of our previous work on a computational method for product-embedded disassembly, which newly incorporates EOL treatments of disassembled components and subassemblies as additional decision variables, and LCA focusing on EOL treatments as a means to evaluate environmental impacts. The method was successfully applied to a realistic example of a desktop computer assembly, and a set of Pareto optimal solutions is obtained as design alternatives.

Future work includes the adoption of more detailed LCA covering entire product life including the production and use phases, the development of more efficient optimization algorithm, the study on the effect of embedded disassembly on assembly, and the derivation of the generalizable design rules through the comparison of the optimization results with the existing designs of other product types.

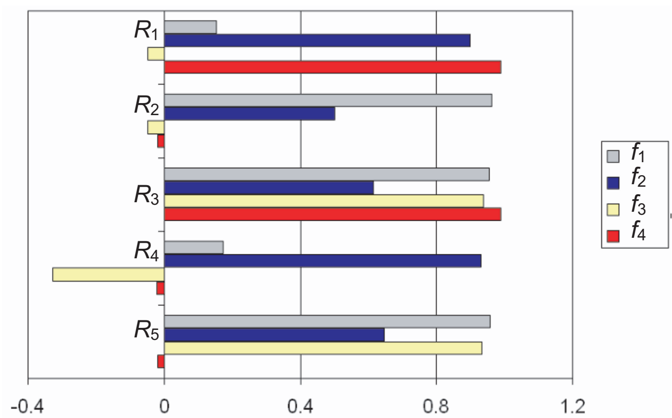


Figure 2-14. Objective function values of R1, R2, R3, R4 and R5 (scaled as f1 : 1/40000, f2 : 1/1300, f3 : 1/400, f4 : 1/36000).

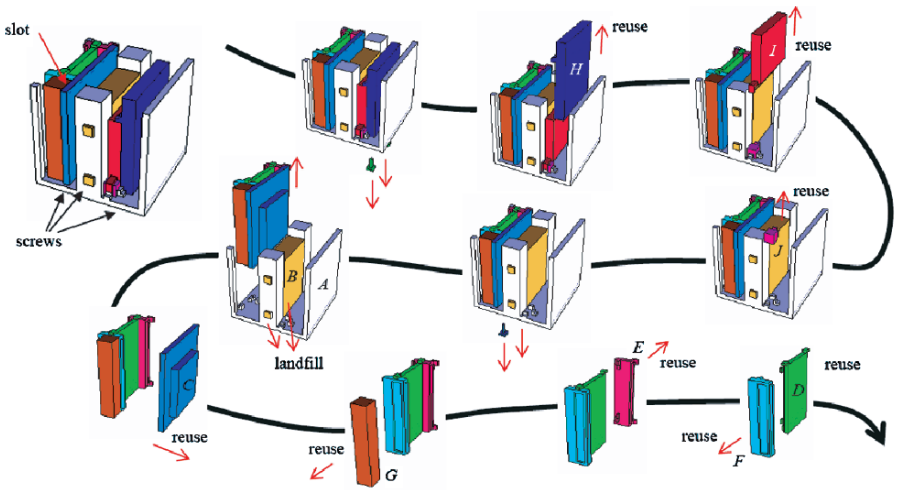


Figure 2-15. Optimal disassembly sequence of R_3 with the optimal EOL treatments of components.

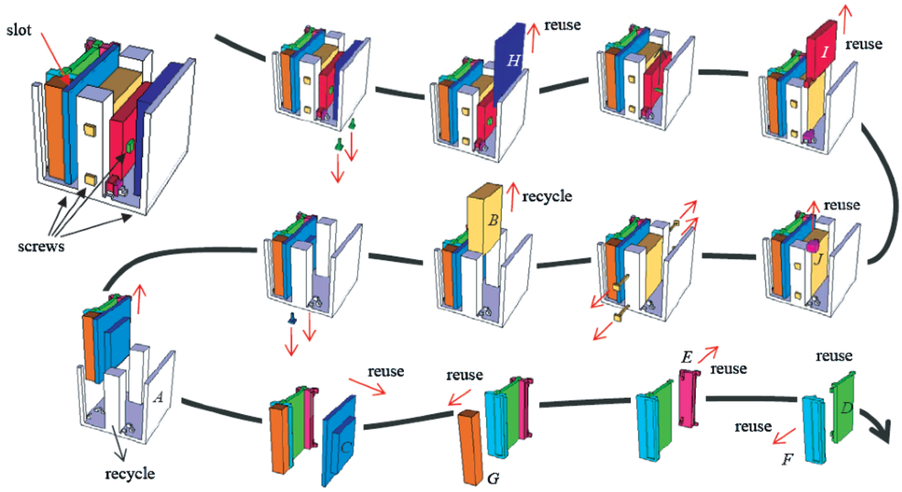


Figure 2-16. Optimal disassembly sequence of R_5 with the optimal EOL treatments of components.

Acknowledgments

The funding for this research was provided by the National Science Foundation of the United States through grant # BES-0124415. Any options, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

Aanstoos, T.A., Torres, V.M., and Nichols, S.P. (1998) Energy model for end-of-life computer disposition, *IEEE Transactions on components, packaging, and manufacturing technology*, 21(4): 295–301.

Baldwin, D.F., Abell, TE., Lui, M.-C., De Fazio, T.L., and Whitney, D.E. (1992) An integrated computer aid for generating and evaluating assembly sequences for mechanical products, *IEEE Transactions on Robotics and Automation*, 7(1): 78–94.

Beasley, D. and Martin, R.R. (1993) Disassembly sequences for objects built from unit cubes, *Journal of Compute-Aided Design*, 25(12): 751–761.

Bonenberger, P.R. (2000) *The First Snap-Fit Handbook: Creating Attachments for Plastic Parts*, Hanser Gardner Publications, München, Germany.

Boothroyd, G. and Alting, L. (1992) Design for assembly and disassembly, *Annals of CIRP*, 41(22): 625–636.

Boothroyd, G., Dewhurst, P., and Knight, W. (1994) *Product Design for Manufacture and Assembly*, Marcel Dekker, Inc., New York, NY.

Caudill, J.R., Zhou, M., Yan, P., and Jim, J. (2002) Multi-life cycle assessment: an extension of traditional life cycle assessment, In: M.S. Hundal (ed.), *Mechanical Life Cycle Handbook*, Marcel Dekker. New York, NY, pages 43–80.

Chen, R.W., Navinchandra, D., and Prinz, F. (1993) Product design for recyclability: a cost benefit analysis model and its application, *IEEE Transactions on Components, Packaging, and Manufacturing Technology*, 17(4): 502–507.

Chen, S.-F., Oliver, J.H., Chou, S.-Y., and Chen, L.-L. (1997) Parallel disassembly by onion peeling, *Transactions of ASME, Journal of Mechanical Design*, 119(22): 267–274.

Corcoran III, A.L. and Wainwright, R.L. (1992) A genetic algorithm for packing in three dimensions, Proceedings of *the ACM/SIGAPP Symposium on Applied Computing*, Kansas City, Missouri, pages 1021–1030.

Das, S.K., Yedlarajiah, P., and Narendra, R. (2000) An approach for estimating the end-of-life product disassembly effort and cost, *International Journal of Production Research*, 38(3): 657–673.

De Fazio, T.L. and Whitney, D.E. (1987) Simplified generation of all mechanical assembly, *IEEE Transactions of Robotics and Automation*, 3(6): 640–658.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, 6(2): 182–197.

Desai, A. and Mital, A. (2003) Evaluation of disassemblability to enable design for disassembly in mass production, *International Journal of Industrial Ergonomics*, 32(4): 265–281.

Dutta, D. and Woo, T.C. (1995) Algorithm for multiple disassembly and parallel assemblies, *Transactions of ASME, Journal of Engineering for Industry*, 117: 102–109.

Fonseca, C.M. and Fleming, P.J. (1993) Genetic algorithms for multiobjective optimization: formulation, discussion and generalization, Proceedings of *the 5th International Conference on Genetic Algorithms*, July 17–22, Urbana-Champaign, IL, pages 416–423.

Fujita, K., Akagi, S., and Shimazaki, S. (1996) Optimal space partitioning method based on rectangular duals of planar graphs, *JSME International Journal*, 60: 3662–3669.

Glover, F. (1974) *Heuristics for Integer Programming using Surrogate Constraints*, Business Research Division, University of Colorado.

Glover, F. (1986) Further paths for integer programming and links to artificial intelligence, *Journal of Computer and Operations Research*, 13(5): 533–549.

Goggin, K. and Browne, J. (2000) The resource recovery level decision for end-of-life products, *Production Planning and Control*, 11(7): 628–640.

Goosey, M. and Kellner, R. (2003) Recycling technologies for the treatment of end of life printed circuit boards (PCBs), *Circuit World*, 29(3): 33–37.

Grignon, P.M. and Fadel, G.M. (1999) Configuration design optimization method, Proceedings of *the ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, September 12–15, Las Vegas, Nevada, DETC99/DAC-8575.

Homem dé Mello, L.S. and Sanderson, A.C. (1990) AND/OR graph representation of assembly plans, *IEEE Transactions on Robotics and Automation*, 6(2): 188–199.

Homem dé Mello, L.S. and Sanderson, A.C. (1991) A correct and complete algorithm for generation of mechanical assembly sequences, *IEEE Transactions on Robotics and Automation*, 7(2): 228–240.

Hula, A., Jalali, K., Hamza, K., Skerlos, S., and Saitou, K. (2003) Multi-criteria decision making for optimization of product disassembly under multiple situations, *Environmental Science and Technology*, 37(23): 5303–5313.

Jain, S. and Gea, H.C. (1998) Two-dimensional packing problems using genetic algorithm, *Journal of Engineering with Computers*, 14: 206–213.

Kaufman, S.G., Wilson, R.H., Jones, R.E., Calton, T.L., and Ames, A.L. (1996) The Archimedes 2 mechanical assembly planning system, Proceedings of *the IEEE International Conference on Robotics and Automation*, April, 1996, Minneapolis, Minnesota, pages 3361–3368.

Kroll, E., Beardsley, B., and Parulian, A. (1996) A methodology to evaluate ease of disassembly for product recycling, *IIE Transactions*, 28(10): 837–845.

Kolli, A., Cagan, J., and Rutenbar, R. (1996) Packing of generic, three-dimensional components based on multi-resolution modeling, Proceedings of *the ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, August 18–22, Irvine, California, DETC/DAC-1479.

Kuehr, R. and Williams, E. (Eds.) (2003) *Computers and the Environment*, Kluwer Academic Publishers, Dordrecht, The Netherlands.

Kuo, T. and Hsin-Hung, W. (2005) Fuzzy eco-design product development by using quality function development, Proceedings of the *EcoDesign: Fourth International Symposium on Environmentally Conscious Design and Inverse Manufacturing*, December 12–14, Tokyo, Japan, 2B-3-3F.

Lambert, A.J.D. (1999) Optimal disassembly sequence generation for combined material recycling and part reuse, Proceedings of *the IEEE International Symposium on Assembly and Task Planning*, Portugal, pages 146–151.

Lee, S. and Shin, Y.G. (1990) Assembly planning based on geometric reasoning, *Computer and Graphics*, 14(2): 237–250.

Li, J.R., Tor, S.B., and Khoo, L.P. (2002) A hybrid disassembly sequence planning approach for maintenance, *Transactions of ASME, Journal of Computing and Information Science in Engineering*, 2(1): 28–37.

Matsui, K., Mizuhara, K., Ishii, K., and Catherine, R.M. (1999) Development of products embedded disassembly process based on end-of-life strategies, Proceedings of *EcoDesign: First International Symposium on Environmentally Conscious Design and Inverse Manufacturing*, February 1–3, Tokyo, Japan, pages 570–575.

Minami, S., Pahng, K.F., Jakiela, M. J., and Srivastave, A. (1995) A cellular automata representation for assembly simulation and sequence generation, Proceedings of the *IEEE International Symposium on Assembly and Task Planning*, August 10–11, Pittsburgh, Pennsylvania, pages 56–65.

O'Shea, B., Kaebernick, H., Grewal, S.S., Perlewitz, H., Müller, K., and Seliger, G. (1999) Method for automatic tool selection for disassembly planning, *Assembly Automation*, 19(1): 47–54.

Reap, J. and Bras, B. (2002) Design for disassembly and the value of robotic semi-destructive disassembly, Proceedings of *the ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, September 29 – October 2, Montreal, Canada, DETC2002/DFM-34181.

Rose, C.M. and Stevels, A.M. (2001) Metrics for end-of-life strategies (ELSEIM), *Proceedings of the IEEE International Symposium on Electronics and the Environment*, May 7–9, Denver, Colorado, pages 100–105.

Sodhi, R., Sonnenberg, M. and Das, S. (2004) Evaluating the unfastening effort in design for disassembly and serviceability, *Journal of Engineering Design*, 15(1): 69–90.

Srinivasan, H. and Gadh, R. (2000) Efficient geometric disassembly of multiple components from an assembly using wave propagation, *Transactions of ASME, Journal of Mechanical Design*, 122(2): 179–184.

Sung, R.C.W., Corney, J.R., and Clark, D.E.R. (2001) Automatic assembly feature recognition and disassembly sequence generation, *Transactions of ASME, Journal of Computing and Information Science in Engineering*, 1(4): 291–299.

Takeuchi, S. and Saitou, K. (2005) Design for product-embedded disassembly, Proceedings of *the ASME Design Engineering Technical Conferences*, Long Beach, California, September 24–28, DETC2005-85260.

Takeuchi, S. and Saitou, K. (2006) Design for optimal end-of-life scenario via product-embedded disassembly, Proceedings of *the ASME Design Engineering Technical Conferences*, Philadelphia, Pennsylvania, September 10–13, DETC2006-99475.

Williams, E.D. and Sasaki, Y. (2003) Energy analysis of end-of-life options for personal computers: resell, upgrade, recycle, Proceedings of the *IEEE International Symposium on Electronics and the Environment*, May 19–22, Boston, MA, pages 187–192.

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