**Proceedings of the 2005 IEEE
International Conference on Automation Science and Engineering
Edmonton, Canada, August 1 & 2, 2005**

Design for Product-Embedded Disassembly Pathways*

Shingo Takeuchi and Kazuhiro Saitou

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

{stakeuch, kazu}@umich.edu

Abstract **– This paper presents a computational method for designing an assembly with multiple built-in disassembly pathways, each of which can be activated to retrieve certain components. It is motivated by the global sales of consumer products whose optimal end-of-life options vary geographically due to local recycling/reuse infrastructures and regulatory requirements. Given the sets of components to be retrieved at each location, the method simultaneously determines the spatial configurations of components and locator features, such that each set of desired components is retrieved via a domino-like "self-disassembly" process triggered by the removal of a fastener. A multi-objective generic algorithm is utilized to search for Pareto-optimal designs in terms of the realization of the desired disassembly pathways, the satisfaction of distance specifications among components, the minimization of disassembly cost at each location, and the efficient use of on-component locator features. A case study demonstrates the feasibility of the method.**

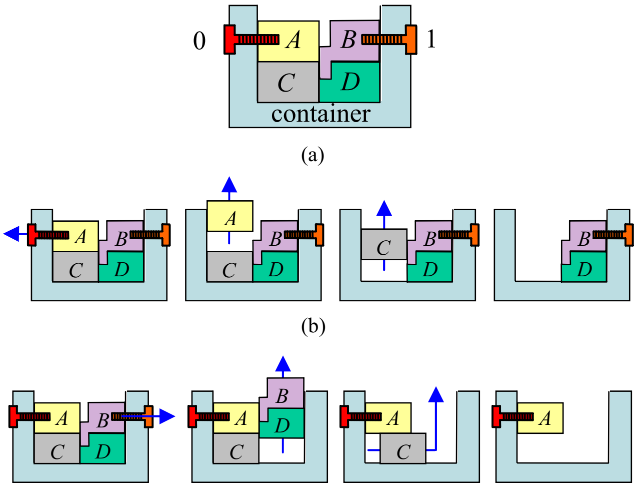
Index Terms – Design for Disassembly, Computer Aided Design, Product Design Automation, Multi-Objective Genetic Algorithm.

I. INTRODUCTION

The global increase in the abandoned products prompted the regulatory and voluntary initiatives for recycle and reuse around the world. Consequently, manufacturers are becoming more responsible for the end-of-life (EOL) treatments of their products at all locations where they are sold. Since both material recycling and component reuse typically require the disassembly of products, Design for Disassembly (DFD) has become a key design issue for realizing optimal EOL treatments in mass-produced consumer products [1].

DFD of globally-sold products poses a unique challenge, since the optimal EOL treatments vary greatly depending on the local recycling/reuse infrastructures and regulatory requirements [2]. For instance, consumer products sold in Europe are subject to significant disassembly to meet the European Union (EU) directive on Waste Electric and Electronic Equipment (WEEE) (more than 50% of product must be recycled). On the other hand, the same product sold in the United States should be disassembled only for maximum economical gain since currently no regulation exists. The optimal disassembly process in Europe, therefore, would naturally be different from the one in the United States [2].

The above thoughts motivated us to develop a concept of multiple product-embedded disassembly pathways, where different components can be retrieved from an assembly via a domino-like "self-disassembly" process triggered by the removal of a different fastener. Fig. 1 illustrates the concept. Suppose the product in Fig. 1 (a) is disassembled at two locations 0 and 1, and the retrieval of component *C* (made of a valuable material) is desired at both locations, whereas component *D* (made of a toxic material) needs to be removed at only location 1 due to the regulatory requirement. At location 0, the disassembly operator can simply remove screw 0 which activates a disassembly pathway of *A* and then *C*, as shown in Fig. 1 (b). In this case, *B* and *D* cannot be disassembled since the motion of *D* relative to *B* is constrained by a locator feature (the tab on *B* and the slot on *D*), and the motion of *B* relative to the container is constrained by screw 1. Similarly, screw 1 can be removed at location 1, to activate another disassembly pathway of *B*, *D*, and then *C*, as shown in Fig. 1 (b). Since the removal of screws 0 and 1 initiates different domino-like "self-disassembly" processes, they are referred to as *trigger screws* in the rest of the paper.



(a)

(b)

(c)

Fig. 1 Concept of multiple product-embedded disassembly pathways.

This paper presents a computational method for designing assemblies with such embedded disassembly pathways. Given the sets of components to be retrieved at location 0 and 1, the method simultaneously determines the spatial configurations of components and locator features such that each set of desired components is retrieved via a domino-like "self-disassembly" process triggered by the

---

* This work is supported by NSF 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 NSF.

0-7803-9426-7/05/$20.00 ©2005 IEEE

removal of a fastener as illustrated in Fig. 1. A multi-objective generic algorithm [3] is utilized to search for Pareto-optimal designs in terms of the realization of the two desired disassembly pathways, the satisfaction of distance specifications among components, the minimization of disassembly cost at each location, and the efficient use of on-component locator features.

The following sections discuss the related work, the proposed method, and a case study. The paper concludes with the summary and the recommendation for the future work.

II. RELATED WORK

A. *Design for Disassembly*

Design for disassembly (DFD) [4] is a class of design method and guidelines to enhance the ease of disassembly for product maintenance and/or EOL treatments. Many researchers proposed the general DFD guidelines from the viewpoint of practical disassembly processes [1, 5]. Reap *et al.* [6] reported DFD guidelines for the robotic semi-destructive disassembly, where detachable or breakable snap fits are preferred to screws due to their ease of disengagement. Matsui *et al.* [7] proposed a concept Products 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 a thermal stress to crack the glass upon the application of current. While these works suggest redesigns to improve the ease of separation for individual joints, they do not address the issues of improving entire disassembly processes involving the removal of multiple joints and components.

B. *Disassembly Sequence Planning*

Disassembly sequence planning (DSP) aims at generating the disassembly sequences that are feasible for a given assembly, where the feasibility of a disassembly sequence is checked by the existence of collision-free motions to disassemble each component in the sequence. Since the disassembly sequence generation problem is NP (Non-deterministic Polynomial time)-complete, the past researches have focused on the efficient heuristic algorithms to approximately solve the problem. Based on a number of important research results on assembly sequence planning [8-12], several automated disassembly sequence generation approaches for 2/2.5D components have been developed [13-15]. More recent works are geared towards DSP with special attention to reuse, recycling, remanufacturing and maintenance [16,17].

These works, however, only address the generation and optimization of disassembly sequences for an assembly with a pre-specified spatial configuration of components. Since the accessibility of a component is heavily dependent on the spatial configuration of its surrounding components, this would seriously limit the opportunity for optimizing an entire assembly. In addition, these works do not address the design of locator and joint configurations, which also have profound impact on the feasibility and quality of a disassembly sequence.

C. *Configuration Design Problem*

While rarely discussed in the context of disassembly, the design of spatial configuration of given shapes has been an active research area by itself [18]. Among the most popular flavours is bin packing problem (BPP), where the total volume (or area for 2D problem) the configuration occupies is to be minimized. Since this problem is also NP-complete, heuristic methods are commonly used. Fujita *et al.* [19] proposed hybrid approaches for 2D plant layout problem, where the topology and geometry of a layout are determined by simulated annealing (SA) [20] and generalized reduced gradient (GRG) method, respectively. Corcoran *et al.* [21] solved a 3D packing problem with GA using multiple crossover methods. Jain *et al.* [22] adopted discrete representation as the object expression and proposed a geometry-based crossover operation for 2D packing problem. Grignon *et al.* [23] proposed a configuration design optimization method by using multi-objective GA, where static and dynamic balance and maintainability considered in addition to configuration volume.

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

D. *Design for Product-Embedded Disassembly Sequence*

The work in this paper is most closely related to our previous work on design for product-embedded disassembly sequence [24], where the spatial configurations of components and locators are simultaneously determined to uniquely realize a given disassembly sequence. The method, however, assumes the optimal (most profitable) disassembly sequence is independent of the special configuration of the components, and can be given *a priori* as an input to the problem. While reasonable for the products assembled in predominantly z-(vertical) direction, this assumption is relaxed in the present work and instead the desired components to be retrieved are regarded as given inputs. Also, the issue of multiple disassembly pathways are not addressed in [24].

III. DESIGN FOR MULTIPLE PRODUCT-EMBEDDED DISASSEMBLIES

The proposed method can be summarized as the following optimization problem:

- **Given:** component geometries, a set of components RC_0 and RC_1 to be retrieved at locations 0 and 1, locator library LL, and distance specification DS among components.
- **Find:** special configuration of components, special configuration of locators and fasteners (including trigger screws) on each component, disassembly sequences and motions to retrieve RC_0 and RC_1.
- **Subject to:** no floating component, no over-lap among components, no unfixed component prior to disassembly, adjacency of components with interlocking locators and fasteners.
- **Minimizing:** disassembly costs to recover RC_0 and RC_1, redundant use of locators and fasteners, violation of DS

Since the problem has four objectives, Pareto optimal solutions will be obtained as outputs, using a multi-objective genetic algorithm. The following section will describe the method in detail.

A. Inputs

As in [24], the components geometries are represented by voxels, due to the efficiency in checking contacts and the simplicity in modifying geometries. CAD inputs are first voxelized using ACIS® solid modeling kernel. Two subsets of components RC_0 and RC_1 are given as the components to be retrieved at locations 0 and 1, respectively.

Locator library *LL* is a set of locator features that can be potentially added on each component to constrain its motion. Fig. 2 shows the seven locators (Fig. 2 (a)-(g)) and one fastener (screw)[2] (Fig. 2 (h)) in *LL* used in the following case study. Fasteners in *LL* are assumed to allow non-destructive detachment, and hence snaps and press fits are not included. Elements in *LL* are classified to three types: Protrusion, Void, and Fastener, according to their characteristics [24]. In Fig. 2, FaceRest, FaceSlot, FaceTab, EdgeRest, EdgeSlot and EdgeTab belong to Protrusion, Boss belongs to Void, and Screw belongs to Fastener. An important aspect of this classification is that no locator of the same type can co-exist on the same face and edge of the component due to the geometric conflict.

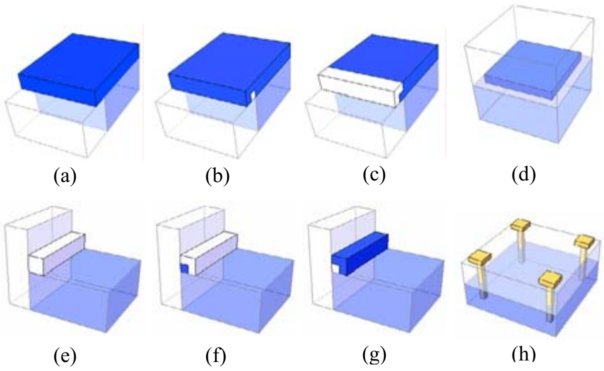

(a) (b) (c) (d)
(e) (f) (g) (h)

Fig. 2 Locator library used in the case study: (a) FaceRest, (b) Face Slot, (c) FaceTab, (d) Boss, (e) EdgeRest, (f) EdgeSlot, (g) EdgeTab, (h) Screw.

Distances or adjacency among components are often constrained by their functional relationship. For example, the cooling fan should be positioned near the CPU in the component configuration of a laptop computer. Since the distance between some pairs of components are more important than the others, the distance specification (*DS*) is defined as a set of the weights of importance for the distances between each pair of components (measured between two designated voxels) that needs to be minimized. If the weight between to two components is zero, the distance between the two components is considered as unimportant and can be arbitrary chosen. Fig. 3 shows an example of the distance specification among three components.

B. Design Variables

There are three design variables for the problem. The first variable $\boldsymbol{x}$, *configuration vector,* is a vector of the translations of components relative to the global reference frame:

$$\boldsymbol{x} = (x_0, y_0, z_0, x_1, y_1, z_1, \ldots, x_{nc\text{-}1}, y_{nc\text{-}1}, z_{nc\text{-}1}) \quad (1)$$

where *nc* is the number of components in the assembly, and x_i, y_i, and z_i, (i = 0, 1, … nc-1) are the translation of the i-th component in x-, y-, and z-directions, respectively. Note that no rotational motions are considered in the present work.

The second variable $\boldsymbol{y}$, *locator vector,* is a vector of the locator id# of each type in LL, at the positions on the components where the locators/fasteners can be added:

$$\boldsymbol{y} = (pr_0, pr_{1,\ldots,}pr_{npr\text{-}1,}\ vo_0, vo_{1,\ldots,}\ vo_{nvo\text{-}1,}\ fa_0, fa_{1,\ldots,}fa_{nfa\text{-}1}) \quad (2)$$

where pr_i (i = 0, …, npr-1), vo_i (i = 0, …, nvo-1) and fa_i (i = 0, …, nfa-1) are the locator id# of type Protrusion, Void and Fastener in *LL*, respectively, and *npr*, *nvo*, and *nfa* are the numbers of the potential positions for the locators of type Protrusion, Void, and Fasteners, respectively.

The third variable $\boldsymbol{z}$, *trigger vector*, is a vector of the component id# fixed by trigger screws:

$$\boldsymbol{z} = (t_0, t_{1,\ \ldots,}\ t_{nl}) \quad (3)$$

where t_i is the component id# fixed by the trigger screw at location i, and *nl* is the number of locations at which disassembly takes place. Since disassemblies of two different locations 0 and 1 are considered in this paper, the size of $\boldsymbol{z}$ is two ($nl = 2$).

Variables $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$ are simply concatenated to form a linear chromosome in multi-objective genetic algorithm used to solve the optimization problem.

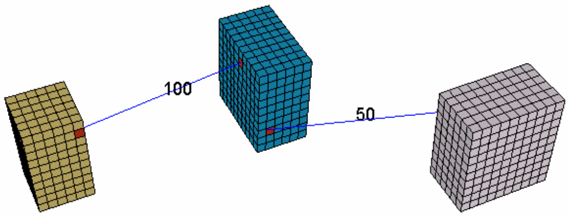


Fig. 3 An example of distance specification (DS). The labeled lines between two voxels indicate the weights of importance for the corresponding distances.

C. Constraints

The spatial configuration of components as specified by $\boldsymbol{x}$, whose geometries are altered by adding the locators as specified by $\boldsymbol{y}$, in the assembly where trigger screws are fixing the components as specified by $\boldsymbol{z}$, must satisfy the following five constraints:

- No floating components
- No overlap among components
- No unfixed component prior to disassembly
- Adjacency of components fixed by trigger screws with a fixed component (such as the container)
- Adjacency of components with interlocking locators

The last constraint is necessary since locators FaceSlot, FaceTab, Boss, EdgeRest, EdgeSlot and EdgeTab require an adjacent component with interlocking features, which is not specified by $\boldsymbol{y}$. If a component with these locators lacks an adjacent component to which the interlocking feature can be

---

[2] Fasteners are considered as a special case of locators and are included in *LL*

added, the configuration is considered as infeasible. Fig. 4 illustrates an example, where an EdgeSlot locator (base feature) cannot be added to the base component in Fig. 4 (b) since the target component is located on the opposite side of the base component.

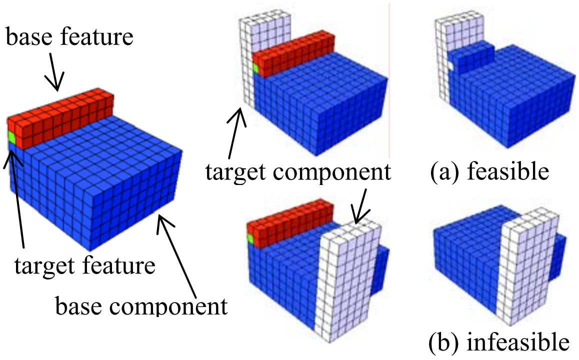


Fig. 4 An example of the feasibility of interlocking locators: (a) feasible, and (b) infeasible.

D. Objective Functions

A candidate design as specified by $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ is evaluated according to four criteria: (1) efficient disassembly at location 0, (2) efficient disassembly at location 1, (3) satisfaction of DS, (4) efficient use of locator features.

The first and second objective functions (to be minimized) are for the efficient disassemblies at location 0 and 1 defined as:

$$f_{i+1}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}) = \min_{p\in P_i}\{w\cdot \text{unretrieved}_i(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},p) + \text{disassembly\_cost}_i(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},p)\} \quad (5)$$

where i = 0, 1, w is weight, P_i is a set of all disassembly sequences that can retrieve *some* components in RC_i, $\text{unretrieved}_i(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},p)$ is the number of components in RC_i that are not retrieved by disassembly sequence p, and $\text{disassembly\_cost}_i(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},p)$ is the cost of disassembly sequence p. Based on the 2-disassemblability criterion [26, 27] (if a subassembly can be disassembled within two consecutive motions), the AND/OR graph [8] of P_i is computed as follows:

1. Set the component specified by t_j ($j \neq i$) as the fixed component, and push the assembly to stack Q and the AND/OR graph.
2. Pop a subassembly s from Q.
3. For each subassembly $ss \subset s$ that does not contain any fixed components and contains some components in RC_i, check the 2-disassemblability of ss from s. If ss is 2-disassemblable, add ss and $t = s\backslash ss$ to the AND/OR graph. If ss is composed of multiple components and contains components in RC_i, push ss to Q. Also, do the same for t.
4. If $Q = \emptyset$, return. Otherwise go to step 2.

Once P_i is obtained, $\text{disassembly\_cost}_i(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},p)$ for disassembly sequence p is calculated as:

$$\text{disassembly\_cost}_i(\boldsymbol{x},\boldsymbol{y},\boldsymbol{z},p) = \sum_{j=0}^{2} w_j \cdot dc_j \quad (6)$$

where dc_0 is the number of orientation changes, dc_1 is the sum of the moved distance of disassembled components, dc_2 is the number of removed fasteners and w_j is the weight of dc_j.

The third objective function (to be minimized) is for the satisfaction of DS, given as:

$$f_3(\boldsymbol{x},\boldsymbol{y}) = \sum_i w_i d_i \quad (7)$$

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

Finally, the fourth objective function (to be minimized) is for the efficient use of locator features, given as the total increase in manufacturing cost due to the addition of locators to components:

$$f_4(\boldsymbol{x},\boldsymbol{y}) = \sum_i mc_i \quad (8)$$

where mc_i is the manufacturing cost of the i-th locators in the assembly.

IV. CASE STUDY

The proposed method is applied to an assembly composed of 10 components with DS shown in Fig. 5, where component A is considered as fixed, and $RC_0 = \{B, I\}$ and $RC_1 = \{C\}$. LL in Fig. 2 is used.

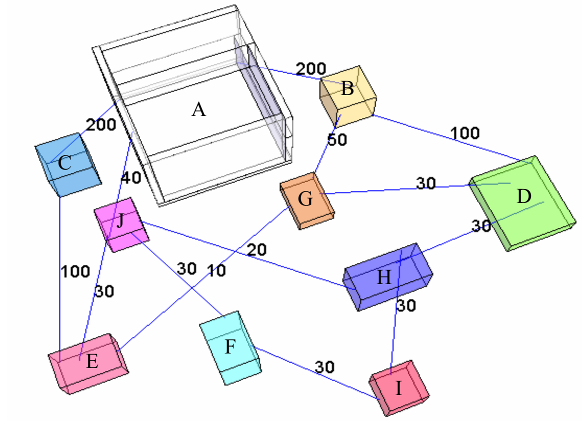


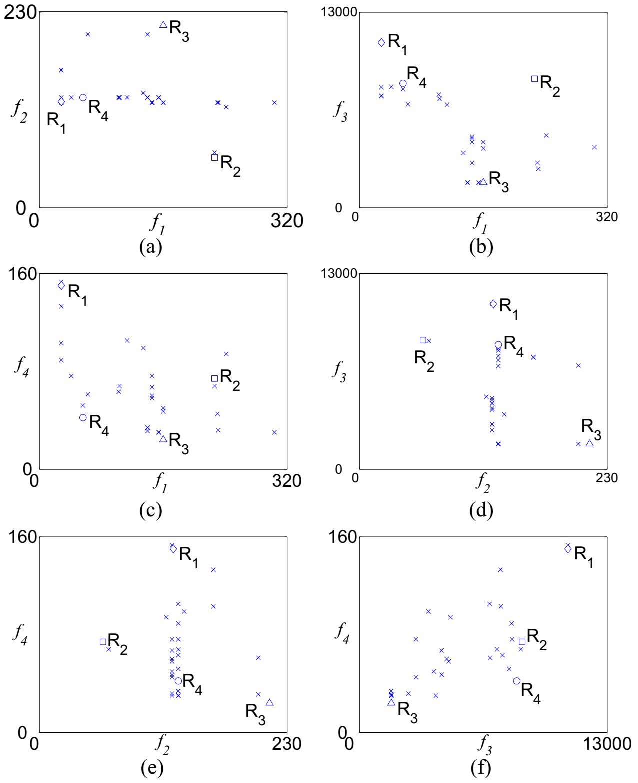
Fig. 5 Assembly used for the case study.

Among 99 Pareto optimal solutions obtained by multi-objective genetic algorithm [3] with population of 150 and at generation 400, Fig. 6 shows the 35 solutions that enable the retrieval of all components in RC_0 and RC_1. Since there are four objective functions f_1, f_2, f_3 and f_4, the resulting 4-dimensonal space is projected on to six 2-dimensional spaces in Fig. 6 (a)-(f).

Four representative Pareto optimal solutions, annotated as R_1, R_2, R_3 and R_4 in Fig. 6 are shown in Fig. 7 (a)-(d). The objective function values for R_1, R_2, R_3 and R_4 are listed in the Table I and also plotted on a spider web diagram in Fig. 8. Solutions R_1, R_2 and R_3 are the best results only considering the value of f_1, f_2, and f_3 (also f_4) respectively, whereas R_4 is a balanced result in all four objectives. Fig. 9 shows the details of solution R_3: the trigger screws (Fig. 9 (a)), the components in RC_0 (Fig. 9 (b)) and RC_1 (Fig. 9 (c)), and the disassembly sequences for location 0 (Fig. 9 (d)) and location 1 (Fig. 9 (e)). It can be seen that the desired sets of components are indeed retrieved via domino-like "self-disassembly" processes initiated by the removal of the respective trigger screws.

V. SUMMARY AND FUTURE WORK

This paper presented a computational method for designing an assembly with multiple built-in disassembly pathways, each of which can be activated to retrieve certain components. It is motivated by the global sales of consumer products whose optimal end-of-life options vary geographically due to local recycling/reuse infrastructures and regulatory requirements. Given the sets of components to be retrieved at each location, the method simultaneously determines the spatial configurations of components and locator features, such that each set of desired components is retrieved via a domino-like "self-disassembly" process triggered by the removal of a fastener. A multi-objective generic algorithm is utilized to search for Pareto-optimal designs in terms of the realization of the desired disassembly pathways, the satisfaction of distance specifications among components, the minimization of disassembly cost at each location, and the efficient use of on-component locator features. A case study demonstrates the feasibility of the method. Although the results obtained by the proposed method cannot be used as the final design due to a number of other design factors, they are expected to provide early insights on designers during conceptual design stages.



(a) (b)
(c) (d)
(e) (f)

Fig. 6 Distribution of Pareto optimal solutions in objective function space.

The future work includes the incorporation of rotations in the allowable disassembly motions and an application to more realistic examples with larger number of components and LCA data. The computational time of the proposed method grows quadratically with the number of allowable disassemble motions (due to additional motions to examine in the 2-disassembleability check), whereas it grows exponentially with the number of components. While the voxel representation of component geometries greatly enhances the run-time efficiency, further developments are necessary to address these problems with a larger scale.

TABLE I
OBJECTIVE FUNCTION VALUES FOR R_1, R_2, R_3 AND R_4

	f_1	f_2	f_3	f_4
R_1	29	124.6	10960.8	150
R_2	226.6	59.2	8559.13	74
R_3	160.4	214	1685.49	24
R_4	57	129.2	8267.06	42

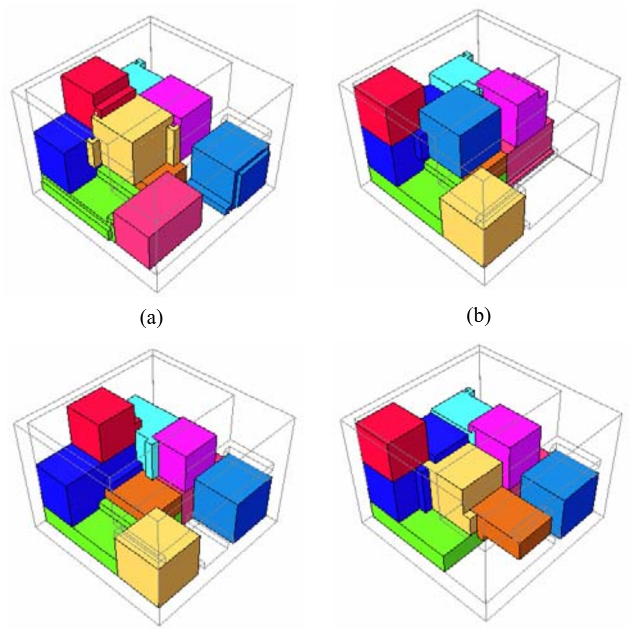

(a) (b)
(c) (d)

Fig. 7 Representative Pareto optimal solutions labeled in Fig. 6: (a) R_1, (b) R_2, (c) R_3, and (d) R_4.

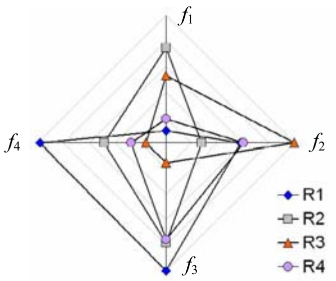


Fig. 8 Spider web diagram for the objective function values of the representative Pareto optimal solutions R_1 -R_4.

REFERENCES

[1] Shetty, D., Rawolle, K., and Campana, C. "A new methodology for ease-of-disassembly in production design," *Recent Advances in Design for Manufacture (DFM),* ASME 2000.

[2] A. Hulla, K. Jalali, K. Hamza, S. Skerlos, K. Saitou, "Multi-criteria decision making for optimization of product disassembly under multiple situations," *Environmental Science and Technology, Special issue on Principles of Green Engineering*, v. 37:23, 2003, pp. 5303 - 5313.

[3] C. A. Coello, D. A. Veldhuizen, and G. B. Lamont, "Evolutionary algorithms for solving multi-objective optimization problems", *volume 5 of Book Series on Genetic Algorithms and Evolutionary Computation*, Kluwer Academic Publishers, ISBN-0306467623.

[4] Mahendra S. Hundal, "Mechanical Life Cycle Handbook," Marcel Dekker, Inc. , 2002, ISBN: 0-8247-0572-6.

[5] H. P. Wiendahl, G. Seliger, H. Perlewitz, and S. Burkner, "A general approach to disassembly planning and control," *Journal of Production Planning and Control,* 1999, vol. 10, no. 8, pp. 718-726.

[6] J. Reap and B. Bras, "Design for disassembly and the value of robotic semi-destructive disassembly," Proceedings of *DETC'02, ASME 2002 Design Engineering Technical Conferences, And Computers and Information in Engineering Conferences*, September 29 – October 2, 2002, Montreal, Canada

[7] K. Matsui, K. Mizuhara, K. Ishii, and R. M. Catherine, "Development of products embedded disassembly process based on end-of-life strategies," *EcoDesign 1999, First International Symposium on Environmentally Conscious Design and Inverse Manufacturing*, February 1-3, 1999, pp. 570-575.

[8] L. S. Homem dé Mello and A. C. Sanderson "AND/OR graph representation of assembly plans," *IEEE Transactions on Robotics and Automation*, vol. 6, pp. 188-99, 1990.

[9] T. L. De Fazio and D. E. Whitney, "Simplified generation of all mechanical assembly," *IEEE Journal of Robotics and Automation*, December, 1987, vol. 3, No. 6.

[10] S. Lee and Y. G. Shin, "Assembly planning based on geometric reasoning," *Computer & Graphics,* vol. 14, No. 2, pp. 237-250, 1990.

[11] L. S. Homem dé Mello and A. C. Sanderson "A correct and complete algorithm for the generation of mechanical assembly sequences," *IEEE Transactions on Robotics and Automation*, vol. 7, pp. 228-40, 1991.

[12] D. F. Baldwin, T. E. Abell, 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*, vol. 7, No. 1, pp. 78-94, 1991.

[13] D. Dutta and T.C. Woo, "Algorithm for multiple disassembly and parallel assemblies," *Transactions of the ASME, Journal of Engineering for Industry*, vol. 117, pp. 102-9, 1995.

[14] S.-F. Chen, J. H. Oliver, S.-Y. Chou and L.-L. Chen, "Parallel disassembly by onion peeling," *Transactions of the ASME*, *Journal of Mechanical Design,* vol. 119, pp. 267-274, 1997.

[15] S. G. Kaufman, R. H. Wilson, R. E. Jones, T. L. Calton and A. L. Ames, , "The Archimedes 2 mechanical assembly planning system," *Proceedings of 1996 IEE, International Conference on Robotics and Automation,* April, 1996, Minneapolis, MN.

[16] K. Lee and R. Gadh, "Destructive disassembly to support virtual prototyping," IIE Transactions, vol. 30, no. 10, pp. 959-972(14), October, 1998.

[17] A. J. D. Lambert, "Optimal disassembly of complex products," *Journal of Technovation*, 1997, vol. 35, No. 9, pp. 2509-2523.

[18] Mittal, S and Frayman, F, "Towards a generic model of configuration tasks," *In proceedings of the 11th IJCAI*, vol. 2 no. 43, 1989, pp. 1395-1401.

[19] K. Fujita, S. Akagi and S. Shimazaki, "Optimal space partitioning method based on rectangular duals of planar graphs," *JSME International Journal*, vol. 60, pp. 3662-3669, 1996.

[20] A. Kolli, J. Cagan and R. Rutenbar, "Packing of generic, three-dimensional components based on multi-resolution modeling," Proceedings of *DETC96, 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, August 18-22, 1996, Irvine, CA.

[21] A. L. Corcoran III and R. L. Wainwright, "A genetic algorithm for packing in three dimensions," Proceedings of the *1992 ACM/SIGAPP Symposium on Applied Computing*, 1992, Kansas City, MO.

[22] S. Jain and H. C. Gea, "Two-dimensional packing problems using genetic algorithm," *Journal of Engineering with Computers*, 1998, vol. 14, pp. 206-213.

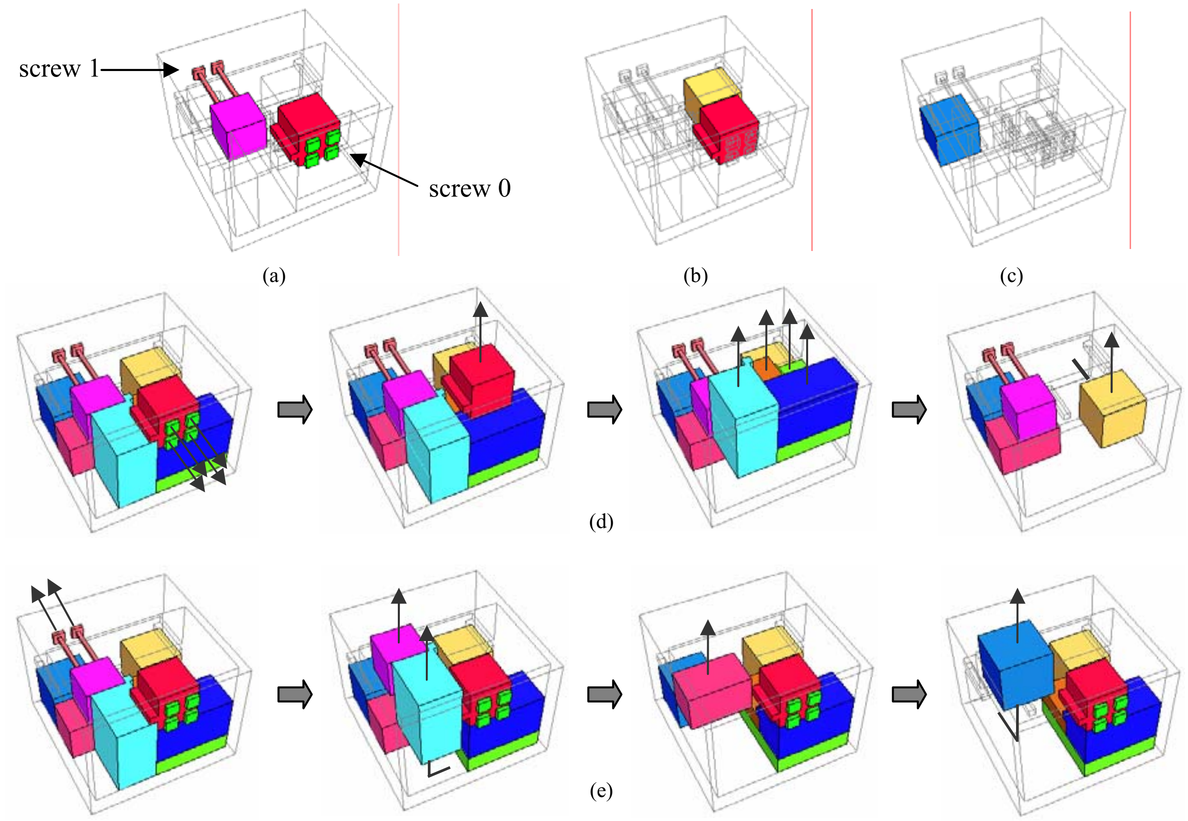
[23] P. M. Grignon and G. M. Fadel, "Configuration design optimization method," Proceedings of *DETC99, 1999 ASME Design Engineering Technical Conferences*, September 12-15, 1999, Las Vegas, NV.

[24] S. Takeuchi and K. Saitou, "Design for product embedded disassembly sequence," in press.

[25] Paul R. Bonenberger, *The First Snap-Fit Handbook: Creating Attachments for Plastic Parts*, Hanser Gardner Publications, ISBN-1-56990-279-8.

[26] T. C. Woo and D. Dutta, "Automatic Disassembly and Total Ordering in Three Dimensions," *Transactions of ASME, Journal of Engineering for Industry*, 1991, vol. 113, pp. 207-213.

[27] D. Beasley and R. R. Martin, "Disassembly Sequences for Objects Built from Unit Cubes," *Journal of Compute-Aided Design*, December, 1993, vol. 25, No. 12.



(a) (b) (c)

(d)

(e)

Fig. 9 Details of design R_3: (a) trigger screws, (b) RC_0, (c) RC_1, (d) disassembly sequence for location 0, and (d) disassembly sequence for location 1