INTERNATIONAL DESIGN CONFERENCE - DESIGN 2012
Dubrovnik - Croatia, May 21 - 24, 2012.

OPTIMAL DESIGN OF MODULAR CONFIGURATION CONSIDERING HIERARCHICY OF A PRODUCT FUNCTION STRUCTURE

M. Kobayashi, Y. Matsumoto and M. Higashi

Keywords: modular design, design for environment, design optimization, function structure, hierarchical genetic algorithm

1. Introduction

Due to rise of environmental awareness in recent years, engineering designers are required to consider not only product model but also whole product lifecycle in order to reduce environmental burdens of their products. Modular design is originally a technique intended to shorten development period by parallel design for each module, improve productivity and reduce cost by standardization of modules. In recent years, modular design is also receiving attention in research field of design for environment (DfE). This is because adequate modularization can improve product's recyclability, reusability, maintainability etc., which lead to improvement of product's environmental characteristics.

Modular design for DfE has been studied for many researchers. Gu and Sosale proposed a method in which interaction among components comprising a module is analyzed by using an interaction matrix and a module configuration of a whole product is decided in order to minimize such interaction [Gu and Sosale 1999]. Ishii et al. proposed method for evaluating modular configuration by using 4 evaluation charts named modularity design chart, manufacturing evaluation chart, serviceability evaluation chart and recyclability evaluation chart [Ishii et al. 1995]. 4 evaluation charts aid in enhancing life-cycle modularity of product families and generations. Shimabukuro et al. proposed a method for determining modular configuration by using SOM (Self-Organizing Map) [Shimabukuro et al. 2005]. In their method, components are classified based on various attributes by using SOM and modules are configured based on those classifications. Koga et al. proposed a method for designing product family architecture considering variety of lineups, profit of platform modules and lifecycle scenario [Koga et al. 2010]. However, since existing methods configure modules based on similarities of components' lifecycle characteristics without consideration of hierarchy of a product function structure, there is a possibility of obtaining functionality and geometrically infeasible modules.

To overcome such problem, this paper proposes a new optimal design method of a modular configuration considering hierarchy of a product function structure. In the proposed method, modules are configured based on a product function structure unlike existing method. Using such representation method, modular design becomes a hierarchical problem. To optimize such hierarchical problem, hierarchical genetic algorithm (HGA) [Yoshimura and Izui 2002] is used. As for the objective function, contribution of each module to improvement of recyclability, reusability and maintainability is evaluated. Using the proposed method, optimal and feasible modular configuration considering whole product lifecycle can be obtained.

2. Function/means tree and hierarchical genetic algorithm

The proposed method adopts Function/Means tree (F/M tree) [Hansen and Andreasen 2005] to represent a product function structure and hierarchical genetic algorithm to optimize modular configuration. This section introduces these methods in advance.

2.1 Function/means tree

F/M tree is a type of a method for representing a product function structure based on the Function/Means law proposed by Hubka [Hubka 1967]. The feature of F/M tree is that a function structure consists of "function" and "means," which is a solution principle to achieve a specified function, and is embodied by alternately defining "function" and "means." A means corresponds one-to-one with a component or assembly of components in a shape design or real-world. Figure 1 shows examples of F/M tree. In the figure, rectangles show functions whereas ovals show means. In the case of Figure 1, "Electric Motor" and "Fuel Engine" are means to achieve a function named "Generate Torque," for example. When a function structure is represented by F/M tree style, more than one means can be configured as alternatives of means for each function and a lower functional structure varies completely according to the upper selections. In the case of Figure 1, there are two alternatives of means for achieving "Generate Torque." Figure 1(a) is the result where "Electric Motor" is selected, whereas Figure 1(b) is the result where "Fuel Engine" is selected. In the figures, black ovals show the selected means. As shown in the figures, lower structures of Figure 1(a) and (b) are completely different.

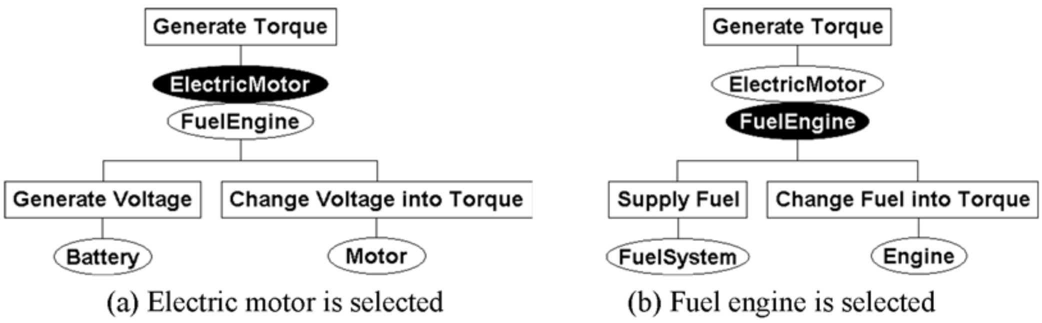
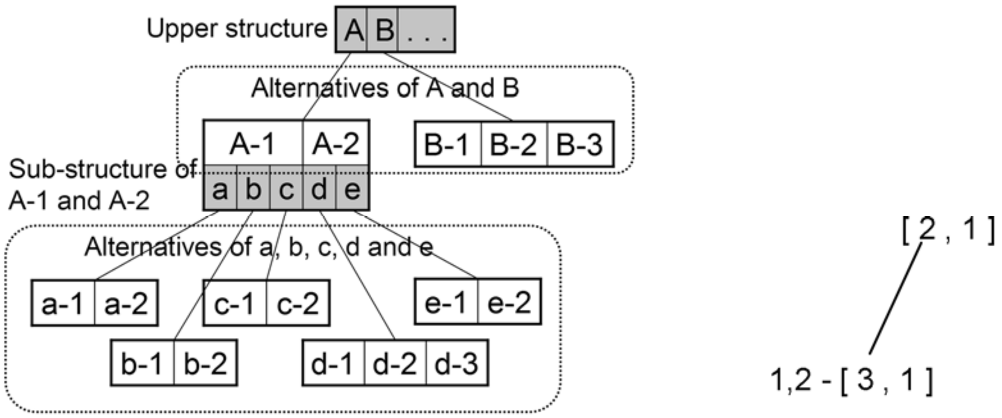


Figure 1. Examples of function / means tree

The advantages of F/M tree are:
(1) Means for achieving a function is explicitly considered, which enables to create more practical function structures.
(2) Various means can be considered for achieving one function and represented in a same tree, which enables broad exploration and consideration of product design concepts.

2.2 Hierarchical genetic algorithm

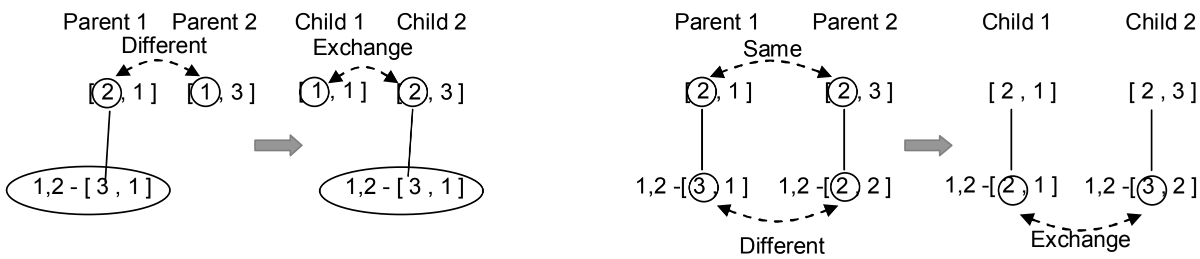
Hierarchical genetic algorithm (HGA) is a type of genetic algorithm. HGA features hierarchical genotype representations to exactly describe hierarchical structures of mechanical system designs. Figure 2 (a) is an example of a simple hierarchical design problem. In this case, the whole system consists of substructures A, B and the others. A-1 and A-2 are alternatives of substructure A, whereas B-1, B-2, and B-3 are as alternatives of substructure B. A-1 incorporates yet lower substructures a, b and c, whereas A-2 incorporates d and e. Substructure a has two alternatives, namely a-1 and a-2, while substructures b, c, d and e also have several alternatives as shown in the figure. If an alternative A-1 is selected, a lower substructure of A consists of a, b and c, whereas, if an alternative A-2 is selected, a lower substructure of A consists of d and e. Figure 2(b) shows the gene example of the hierarchical structure shown in Figure 2. Here, [2, 1] shows that alternatives A-2 and B-1 are selected at the upper structure, whereas [3, 1] shows that alternatives d-3 and e-1 are selected at the substructure of A-2. The prefix notation symbol "1, 2-" added to [3, 1] indicates that [3, 1] are the alternatives belonging to the substructure A-2 that is the second alternative of A.



(a) Simple hierarchical design problem (b) Gene representation of HGA

Figure 2. Basic concept of HGA

The flow of HGA is basically the same as a conventional genetic algorithm. The main differences from a conventional one are hierarchical genotype representations described above and new crossover and mutation operators for treating the hierarchical genotype representations. Figure 3 shows the basic concept of the new crossover operator. See the reference [Yoshimura and Izui 2002] for details of HGA.



(a) Genes of two parents are different (b) Genes of two parents are same

Figure 3. Crossover operator

3. Optimal design of modular configuration considering hierarchy of product function structure

3.1 Preconditions of the proposed method

In the proposed method, the following preconditions are assumed.

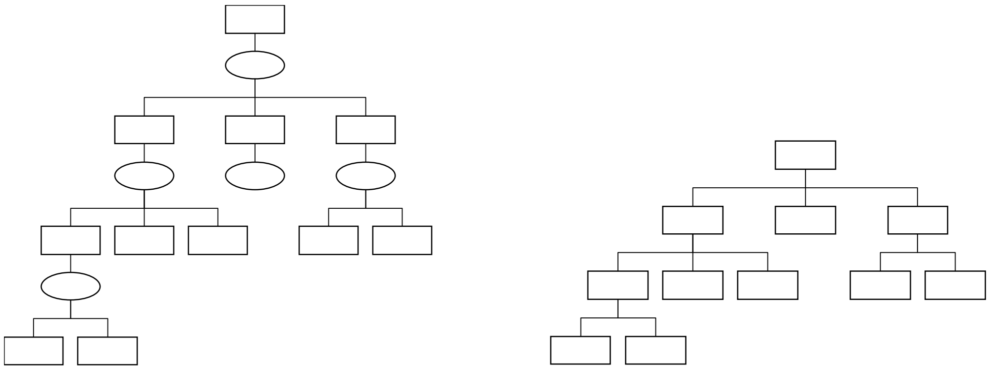
(a) A F/M tree of a design object is completed in advance of modular optimization.

(b) Although F/M tree representation method allows single function to have multiple alternatives of means, as shown in Figure 1, the proposed method assumes that functions and means have only one-to-one correspondence relationships. Thus, a pair of function and means is represented as a single node in this paper for simplicity, as shown in Figure 4(b).

(c) All components have attributes of longevity and material type. In addition, components requiring regular maintenance have attribute of maintenance frequency.

(d) All modules are independent of each other. Thus, a new module can not be configured in the existing modules.

(e) A module must consists of more than one component.



(a) Original representation (b) Simplified representation

Figure 4. F/M tree representation

3.2 Problem definition

In the proposed method, modules are configured based on hierarchy of a product function structure unlike existing modular design methods. In particular, a module is represented as a subtree of a function structure. If a node is selected as a root node of a module, the subtree rooted at the selected node are considered as a module, as shown in Figure 5. In this figure, each group of gray nodes surrounded by dashed line is a module. According to this representation method, a modular is configured by only selecting its root node from a function structure. Thus, modular configuration can be optimized by handling those selections as design variables and using HGA.

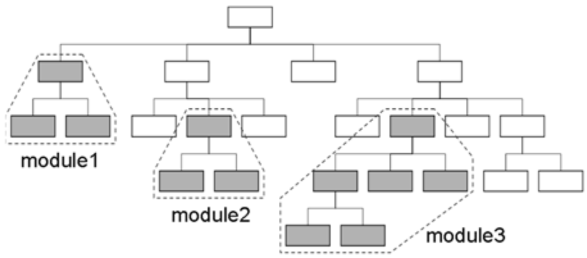


Figure 5. Definition of a module

However, if a function structure shown in Figure 5 is directly used as a search space of modular optimization, it is difficult to maintain consistency of modular configuration during optimization processes. Thus, a search space is constructed and manipulated during optimization processes according to the below two operation.

(a) Since a module must consist of more than one component as mentioned in section 3.1(e), a leaf node of a function structure will not be selected as a root node of a module. Thus, leaf nodes are deleted from search space before starting optimization, as shown in Figure 6.

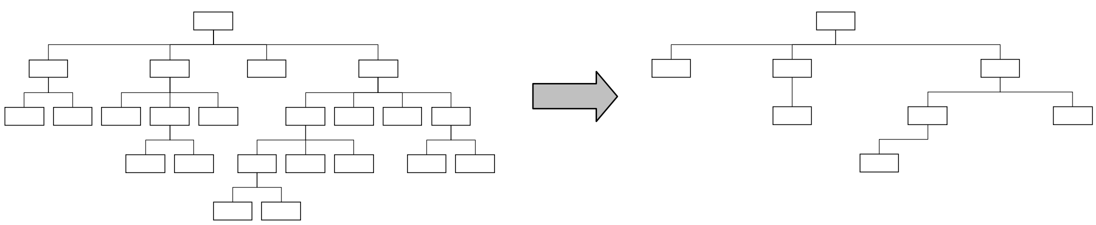


Figure 6. Configuration of search space

(b) As mentioned in section 3.1(d), new modules can not be configured in the existing modules. Thus, if a node is selected as a root of a module, the node and its subtree are replaced by a single node that represents the module, as shown in Figure 7. This operation can be easily done by utilizing function of HGA.

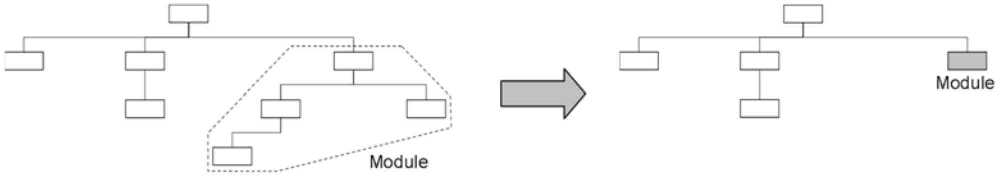


Figure 7. Manipulation of search space

Fitness function of HGA is based on product's recyclability, reusability and maintainability. Its calculation is explained in the next section.

3.3 Evaluation of modular configuration

During the process of HGA, generated modular configurations are evaluated from the viewpoints of recyclability, reusability and maintainability and fitness function is defined as the below equation.

$$Fitness = w_{recycle}EV_{recycle} + w_{reuse}EV_{reuse} + w_{maintenane}EV_{maintenane} \tag{1}$$

Where, $w_{recycle}$, w_{reuse} and $w_{maintenance}$ are weighting coefficients, whereas $EV_{recycle}$, EV_{reuse} and $EV_{maintenance}$ are evaluation values of recyclability, reusability and maintainability respectively. Details and calculation procedure of each EV are explained in the following sections.

(1) Evaluation of recyclability

When products are recycled at the EOL stage, since products are constructed of a variety of materials, products need to be disassembled and components need to be sorted according to material type. However, if all components comprising the module are constructed of same material or materials having material compatibility, the module can be directly sent to recycling processes without further disassembly. Which means disassembly cost and effort can be reduced. Therefore, the proposed method evaluates recyclability based on comparisons of materials used in the components comprising the module. In particular, materials of components comprising the module are compared in pairs and the correlation matrix shown in Table 1 is generated based on those comparisons. The value of each element of the matrix $CM_{recycle\ i,j}$ is set according to the following rules.

Table 1. Corretion matrix

	Compornent 1	Compornent 2	Compornent 3	•••	Compornent n
Compornent 1	-	1	0.5	•••	0
Compornent 2		-	0	•••	0
Compornent 3			-	•••	0.5
⋮				⋱	⋮
Compornent n					-

(i) Compatibility between metal materials

In the case paired components are constructed of metal materials, if their materials are same, the value of the element is set to 1. Otherwise, the value is set to 0.

$$CM_{recycle,ij} = \begin{cases} 1 & Metal_i = Metal_j \\ 0 & Other \end{cases} \tag{2}$$

(ii) Compatibility between plastic materials

In the case paired components are constructed of plastic materials, the proposed method adopts the material compatibility chart proposed by Marks et al. [Marks et al. 1993] shown in Table 2. According to this chart, the value is set to 0, 0.5 or 1.

Table 2. Material compatibility chart

	PC	ABS	PPO/PS	PPO/PS/Nylon	PBT	PBT/PC	ASA	PEI	ABS/PC	Polystyrene	Crystaline-Nylon
PC	-										
ABS	1	-									
PPO/PS	0.5	0	-								
PPO/PS/Nylon	0	0	1	-							
PBT	1	1	0	0	-						
PBT/PC	1	1	0	0	1	-					
ASA	1	0	0	0	0.5	0.5	-				
PEI	0.5	0	0	0	0	0	0	-			
ABS/PC	1	1	0	0	1	1	0.5	0	-		
Polystyrene	0	0	1	0.5	0	0	0	0	0	-	
Crystaline Nylon	0	0.5	0.5	0.5	0	0	0	0	0	0	-

$$CM_{recycle,i,j} = \begin{cases} 1 & Compatible \\ 0.5 & Compatible\ to\ a\ certain\ level \\ 0 & Not\ compatible \end{cases} \tag{3}$$

(iii) Compatibility between metal and plastic materials

There is no compatibility between a metal component and a plastic component. In this case, the value is set to 0.

$$CM_{recycle,i,j} = 0 \tag{4}$$

After calculating the correlation matrix, average value of all element of the matrix is calculated and the value is defined as recyclability of the module. Recyclability of the whole product EVrecycle is defined as the average value of recyclability of all modules comprising the whole product. Equation 5 shows definition of product's recyclability.

$$EV_{recycle} = \sum_{n=1}^{N} \left(\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} CM_{recycle,i,j} / {}_nC_2 \right) / N \tag{5}$$

(2) Evaluation of reusability

Each component has its own longevity and longevity of a modular is equal to the shortest longevity among components comprising the module. From the viewpoint of module reuse, it is ideal that longevity of all components comprising the module is uniform. In this case, since all components comprising the module reach their longevity at the same time, that module can be simply discarded or recycled. If longevity of components comprising the module is not uniform, long-life components are wastefully discarded or the module needs to be repaired by changing broken components. Therefore, the proposed method evaluates reusability based on comparisons of longevity of components comprising the module. In particular, each emelent of correlation matrix is calculated by the below equation. Reusability is calculated by the same calculation procedure as equation 5.

$$CM_{reuse,i,j} = \frac{Longevity\ of\ shorter\ life\ component}{Longevity\ of\ longer\ life\ component} \tag{6}$$

(3) Evaluation of maintainability

Some components may require regular maintenance during use. From the viewpoint of maintenance of modular products, it is desirable and efficient that maintenance frequency of components comprising the module is uniform because the module is removed only once from the product and maintenance of all components can be done at the same time. Therefore, the proposed method evaluates maintainability based on comparisons of maintenance frequency of components comprising the module. In particular, each element of correlation matrix is calculated by the below equation. Maintenability is calculated by the same calculation procedure as equation 5.

$$CM_{maintenance,i,j} = \frac{Maintenance \quad frequency \quad of \quad component \quad requiring \quad more \quad frequent}{Maintenance \quad frequency \quad of \quad component \quad requiring \quad less \quad frequent} \tag{7}$$

4. Case study

To confirm effectiveness of the proposed method, it is applied to a modular design of an inkjet printer. Figure 8 shows a F/M tree of a inkjet printer used here. Based on this F/M tree, search space of modular optimization is configured as shown in Figure 12. All components have attributes of longevity and material type and some components have attribute of frequency of maintenance, as described in the section 3.1(c). Table 3 shows examples of their specifications.

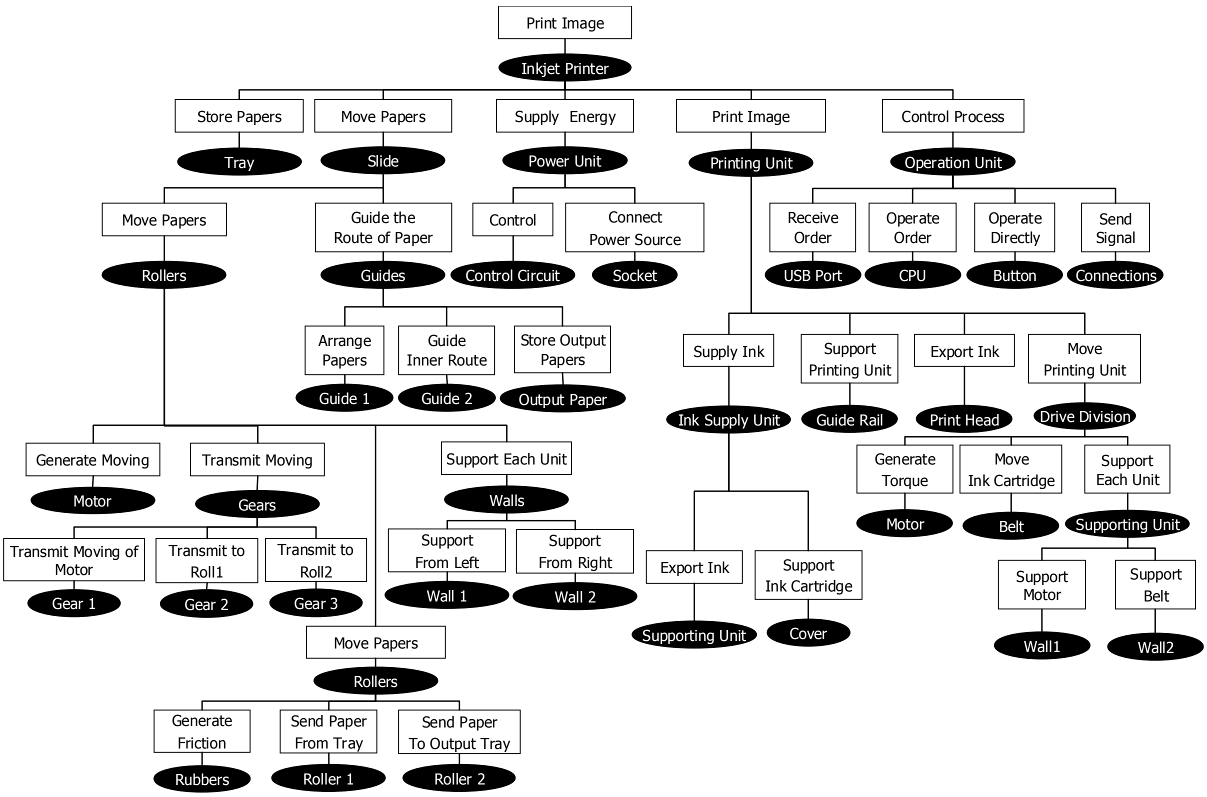


Figure 8. F/M tree of an inkjet printer

Modular optimization is executed 4 times by changing weighting coefficients of fitness function, $w_{recycle}$, w_{reuse} and $w_{maintenance}$. Their values are (0.4, 0.3, 0.3), (0.8, 0.1, 0.1), (0.1, 0.8, 0.1) and (0.1, 0.1, 0.8) respectively. The first set of weighting coefficients places equal importance on recyclability, reusability and maintainability, the other three sets place importance on recyclability, reusability and maintainability respectively.

Results of optimization are shown in Table 4 and Figure 10. Table 4 shows values of $EV_{recycle}$, EV_{reuse}, $EV_{maintenance}$ and fitness function. Table 4 also shows the values of fitness function in the case that no modular is configured. Table 4 shows that fitness function of the optimal modular configuration is higher than that of non-modular configuration. Figure 10 shows obtained optimal modular configurations. In figure 10, each group of gray nodes surrounded by dashed line shows a module. These results show that the proposed method can obtain modular configurations suitable for each set of weighting coefficients.

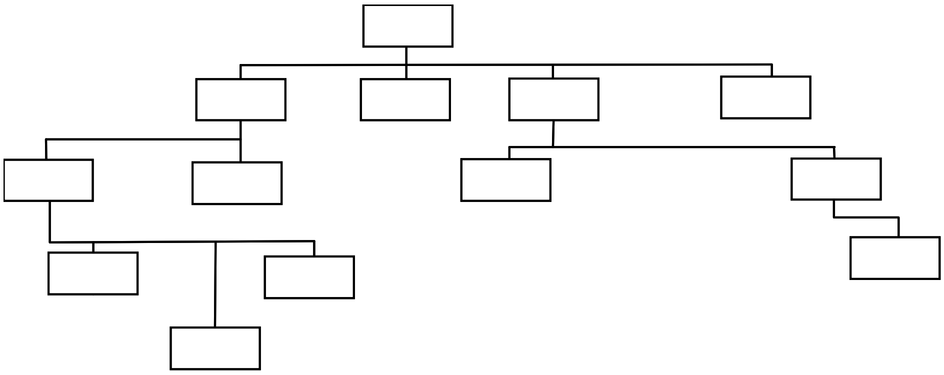


Figure 9. Search space of modular optimization

Table 3. Examples of components' attributes

Component	Material	Longevity	Frequency of maintenance
Tray	ABS	10	—
Motor 1	Aluminum, Copper	5	—
Gear 1	ABS	5	3
Gear 2	ABS	6	4
Gear 3	ABS	6	4
Rubber	EPM	5	—
Roller 1	ABS	10	—
Roller 2	ABS	10	—
Wall 1	ABS	5	—
Wall 2	ABS	8	—
Guide 1	ABS	10	—
Guide 2	ABS	10	—
Output Paper Tray	ABS	5	—
Control Circuit	Copper	5	—
Ink Cartridge	$C_2H_6O_2$, $C_3H_5(OH)_3$	0.5	0.5
Cover	ABS	8	0.5
Guide Rail	ABS	8	—
Print Head	ABS	5	—

Table 4. Optimal results

Weight	EV (recycle)	EV (reuse)	EV (maintenance)	Fitness (Modularized)	Fitness (Not modularized)
(0.4, 0.3, 0.3)	1	0.72	1	0.91	0.35
(0.8, 0.1, 0.1)	1	0.67	1	0.97	0.37
(0.1, 0.8, 0.1)	0.67	0.85	1	0.85	0.56
(0.1, 0.1, 0.8)	0	0.1	1	0.81	0.11

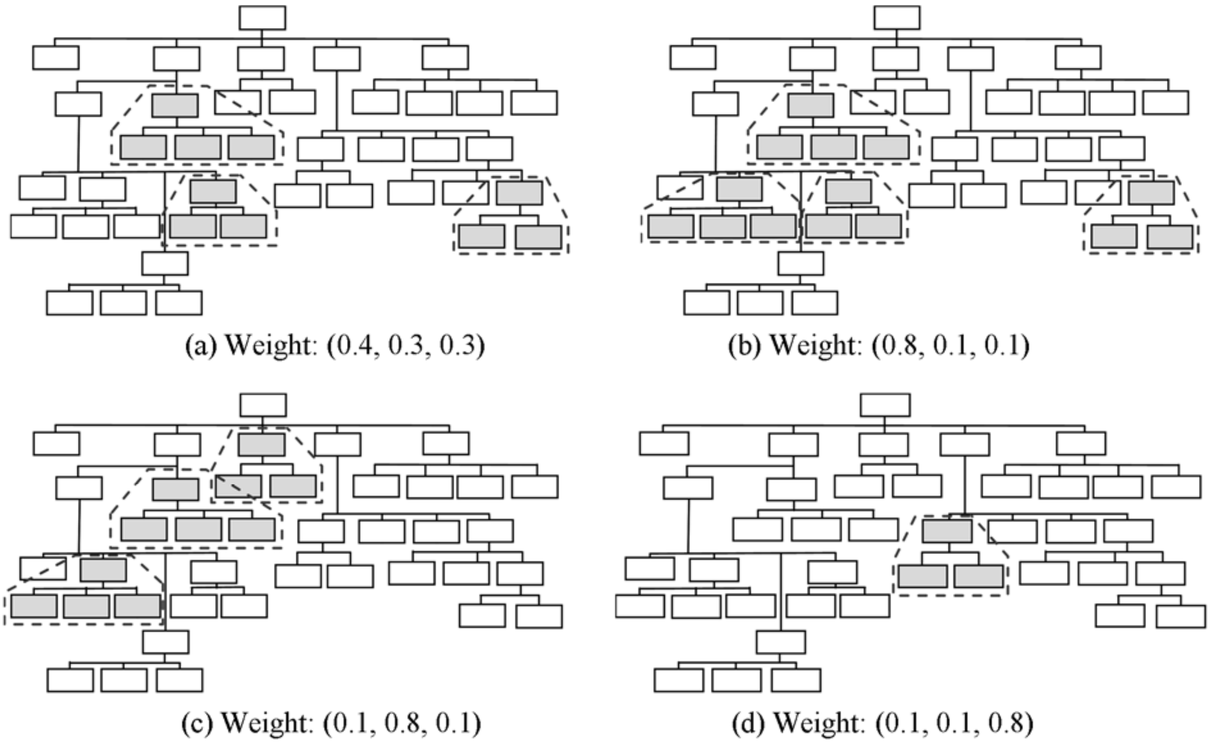


Figure 10. Optimal modular configurations

5. Conclusion

This paper proposes a new optimal design method of modular configuration considering hierarchy of product function structure. In the proposed method, a module is represented as a subtree of a product function structure unlike the exiting methods. Whether each node is selected as a root node of a subtree that represents a modular or not is handled as de sign variables and optimized by using HGA (hierarchical genetic algorithm). Recyclability, reusability and maintainability are evaluated as criteria of optimization and fitness function is calculated based on them. By using the proposed method, a modular configuration that balances recyclability, reusability and maintainability can be obtained. In the case study, the proposed method is applied to a modular design of an inkjet printer and the results show its effectiveness.

As for future works, relaxation of limitation of modular configuration is planned. Although the proposed method only allows a subtree of a function structure as a module due to its method for representing a module, there is no problem from a functional viewpoint that components of sibling nodes is configured as a module. Thus the future method will allow such modular configuration.

Acknowledgement

This study was supported in part by a grant of Strategic Research Foundation Grant-aided Project for Private Universities from Ministry of Education, Culture, Sport, Science, and Technology, Japan (MEXT), 2008-2012 (S0801058).

References

Gu, P., Sosale, S., "Product Modularization for Lifecycle Engineering", Product Modularization for Lifecycle Engineering, Vol.15, No.5, 1999, pp 387-401.

Hansen, C. T., Andreasen, M. M., "Two approaches to synthesis based on the domain theory", In Chakrabarti, C., Engineering Design Synthesis: Understanding, Approaches, and Tools, Springer, 2002, pp. 99.

Hubka, V., "Der Grundlegender Algorithmus für die Lösung von Konstruktionsaufgaben", In: XII. Internationales Wissenschaftliches Kolloquium der Technischen Hochschule Ilmenau "Konstruktion", 1967, pp.69-74.

Ishii, K., Juengel, C., Eubanks, C. F., "Design for Product Variety: Key to Product Line Structuring", Proceedings of 1995 Asme Design Engineering Technical Conferences- 9th International Conference on Design Theory and Methodology, ASME, Vol.83, 1995, pp 499-506.

Koga, T., Aoki, H., Aoyama, K., "A Modular Design Method for Scenario Embedded Product", New World Situation: New Directions in Concurrent Engineering, Part 3, 2010, pp 203-212.

Marks, M. D., Eubanks, C. F., Ishii, K., "Life-Cycle Clumping of Product Designs for Ownership and Retirement", Proceedings of 1993 Asme Design Engineering Technical Conferences- 5th International Conference on Design Theory and Methodology, ASME, Vol.53, pp 83-90.

Shimabukuro, A., Umeda, Y., Kondoh, S., "Development of a Modular Design Method for Inverse Manufacturing", Proceedings of Conference of the Japan Society of Mechanical Engineers, Design and Systems Division, JSME, 2005, pp 98-101.

Yoshimura, M., Izui, K., "Smart Optimization of Machine Systems Using Hierarchical Genotype Representations", Transaction of ASME, Journal of Mechanical Design, Vol.124,No.3, 2002, pp 375-384.

Dr. Masakazu Kobayashi
Associate professor
Toyota Technological Institute
2-12-1, Hisakata, Tempaku, Nagoya, 468-8511, Japan
Telephone: +81-52-809-1784
Email: kobayashi@toyota-ti.ac.jp