ICIC Express Letters ICIC International ©2018 ISSN 1881-803X
Volume **12**, Number **6**, June **2018** pp. **583–590**

OPTIMAL MODULAR DECOMPOSITION FOR MINIMIZING COST OF PREVENTIVE MAINTENANCE UNDER TARGET AVAILABILITY

Changmuk Kang

Department of Industrial and Information Systems Engineering
Soongsil University
369 Sangdo-Ro, Dongjak-Gu, Seoul 06978, Korea
changmuk.kang@ssu.ac.kr

Received December 2017; accepted March 2018

Abstract. *This study considers a system that consists of multiple deteriorating and replaceable components, and assumes that all of them are replaced in a preventive maintenance mode. A modular replacement unit is considered rather than replacing individual component independently; where any component in a module needs to be replaced, the whole module is replaced all together. Small modules are economic, but degrade availability of the system due to frequent maintenance stops. Large modules are expensive, but enhance availability by reducing maintenance frequency. This study proposes a linear-integer programming model to find the optimal modules that minimize the long-run average cost under a target availability. It explores a solution under flexible decomposition structures which has never been allowed in the previous studies.*

Keywords: Replacement module, Preventive maintenance, System availability, Transition matrix

1. **Introduction.** A system with deteriorating components periodically replaces those components with new ones. This study considers a system that consists of multiple deteriorating and replaceable components, and assumes that all of them are replaced in a preventive maintenance mode. In other words, maintenance personnel replace a component before it really fails. Where this replacement is far earlier than the expected life time, a chance of actual failure is negligible. Such maintenance practice is common in mission-critical systems like industrial equipment and jet engines. Whereas a system can regularly inspect the components and replace them on a certain condition [1,2], this study assumes that they are placed only by a planned replacement interval [3]. A system owner strongly requires her system available for most of the time.

This study formulates a problem of decomposing replacement *modules* in this system. A replacement module (or simply a module) is referred to as a collection of components that can be detached, replaced and reassembled as a whole. A straightforward and the cheapest way of maintenance is to replace every single component independently. In this way, however, the replacement job requires much time for disassembly and reassembly of a small component. Moreover, the system has to frequently shut down where many components alternately require replacement operations. It does not satisfy a system owner who strongly wants high availability. Another extreme solution is to renew a whole system whenever any component requires replacement. It is trivially impractical because of the cost, though it does minimize loss of the availability. Between these two extremes, modular replacement can effectively compromise the availability and cost objectives. It can avoid frequent maintenance operations while keeping the cost at a moderate level. For this reason, modular architecture is known as a key strategy for achieving better serviceability and reliability [4]. In order to fully exploit this benefit, a system has to be decomposed into appropriate modules, which is a challenging task for design engineers [5].

This paper proposes a deterministic optimization model to balance the tradeoff between maintenance cost and system availability.

An issue of modular decomposition can be found in the engineering design literature. They mainly concern how to maximize interactions among components within a module while minimizing between-module interactions [6]. Maintenance oriented modular decomposition problem has rarely been discussed in this literature.

In the system reliability literature, it is basically a maintenance problem of a multi-component system. This problem has been widely studied in the reliability literature. Two comprehensive reviews can be found in Cho and Parlar [7] and Wang [8]. According to their common classifications, the module replacement is a kind of opportunistic maintenance. Replacement of a certain component yields an opportunity of replacing other components with a marginal cost. For this kind of maintenance, threshold-based policies have been suggested; whenever a component needs to be replaced for a preventive or corrective purpose, other components that have aged more than a predefined threshold are replaced together [7,8]. A common fixed threshold for all components [9] or variable thresholds for different components [10,11] has been proposed.

Another stream of reliability studies adopts the modular concept for multi-level redundancy allocation. Yun and Kim [12] formulated a problem of maximizing availability of a system by allocating redundant modules for occasional failures rather than redundant components. Yun et al. [13] extended this problem by allowing variable hierarchy levels of different modules. It also explores an optimal module levels for allocating redundancy. A series of following studies [5,14-16] proposed various meta-heuristic methods to solve this problem, and Chung [17] and Han et al. [18] alternatively formulated a problem of minimizing life-cycle cost under target availability. As far as the author studied, all those studies assume a single hierarchical structure and determine module levels in this fixed hierarchy.

This study contributes to the literature by two folds. It first expands the solution space of the problem by allowing flexible decomposition structures. Specifically, this study accommodates multiple choices of decomposition hierarchies and considers it as a decision variable rather than a constraint. It is possible to group components with similar replacement intervals together as a module in this expanded space.

Second, this study enables instant evaluation of the cost and availability measures by deriving their closed-form functions. The previous studies evaluated those performance metrics through simulation of maintenance schedules. Long simulation time hinders to find an optimal decomposition solution in our expanded solution space. The closed-form functions, however, enable to efficient exploration of solutions.

This study is organized as follows. Variables, parameters and constraint and objective functions are mathematically formulated and the final linear-integer programming model is presented in Section 2. This model is applied to a numerical example in Section 3, and the conclusions are made in Section 4.

2. **Problem Statement.** This study assumes mission-critical systems like jet engines that have to seldom fail due to component failure. In addition, the system is assumed a series system; fail of any component makes the whole system fail. In order to satisfy this requirement, every component must be preventively replaced whenever it reaches a predefined age, which is far shorter than the expected lifetime. Under this assumption, maintenance schedule becomes pretty deterministic ignoring possibility of corrective maintenance, by which a component is replaced after failure. Instead, frequent maintenance may degrade availability of the system. The system has to shut down during maintenance of any component because it is assumed as a series system. Maintenance time is determined by setup, disassembly, replacement and reassembly times, and no waiting time due to lack of inventory is involved.

2.1. **Decision variables and parameters.** The objective of the problem is to minimize long-run average maintenance cost while meeting a target availability, which is defined by fraction of operating time over total time of operating and maintenance. A decomposition solution is represented by binary variables x_i's, each of which is 1 if a module i is selected as a replacement unit and 0 otherwise. A module can be a single component, a group of components or a whole system. A set of possible modules with size m is preliminary given based on structural analysis on the system. A feasible decomposition should include all components in any one of modules without duplication.

For module i, replacement cost, maintenance interval (usage life) and maintenance time for replacement are denoted by C_i, L_i and M_i, respectively. For C_i's, a module is more expensive than all its constituting components. If a module is a replacement unit, its L_i has to follow the most frequent interval among the constituting components.

Maintenance time M_i's are obtained by an optimal selective disassembly and reassembly time. This study assumes a flexible disassembly and reassembly structure. Possible disassembly operations are constrained rather than a rigid disassembly sequence, adopting a *transition matrix*. It represents disassembly operations, among the possible modules (including a whole system and individual components) by a transition vector of which values are −1 and 1 for an original module and disassembled modules, respectively. Other elements are all 0. Thus, the matrix rows and columns correspond to possible modules and disassembly operations, respectively. Table 1 illustrates a well-known example of a transition matrix, which is called AFI (assembly from industry) in [19]. Lambert [19,20] presented mathematical formulation of feasible disassembly sequences using this matrix.

Table 1. Transition matrix of AFI

i	Feasible modules	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	$ABCDEFGHJKL$	1	−1	−1	0	0	0	0	0	0	0	0	0	0	0
2	$ABCDEGHJKL$	0	0	1	−1	−1	0	0	0	0	0	0	0	0	0
3	$ABCDGHJKL$	0	0	0	0	1	−1	−1	0	0	0	0	0	0	0
4	$ABCGHJKL$	0	0	0	0	0	0	1	−1	−1	0	0	0	0	0
5	$AGHJKL$	0	0	0	0	0	0	0	0	1	−1	0	0	0	0
6	$ABCDEF$	0	1	0	0	0	0	0	0	0	0	−1	0	0	0
7	$ABCDE$	0	0	0	1	0	0	0	0	0	0	1	−1	0	0
8	$ABCD$	0	0	0	0	0	1	0	0	0	0	0	1	−1	0
9	ABC	0	0	0	0	0	0	0	1	0	0	0	0	1	−1
10	A	0	0	0	0	0	0	0	0	0	1	0	0	0	1
11	B	0	0	0	0	0	0	0	0	1	0	0	0	0	1
12	C	0	0	0	0	0	0	0	0	1	0	0	0	0	1
13	D	0	0	0	0	0	0	1	0	0	0	0	0	1	0
14	E	0	0	0	0	1	0	0	0	0	0	0	1	0	0
15	F	0	0	1	0	0	0	0	0	0	0	1	0	0	0
16	G	0	1	0	1	0	1	0	1	0	1	0	0	0	0
17	H	0	1	0	1	0	1	0	1	0	1	0	0	0	0
18	J	0	1	0	1	0	1	0	1	0	1	0	0	0	0
19	K	0	1	0	1	0	1	0	1	0	1	0	0	0	0
20	L	0	1	0	1	0	1	0	1	0	1	0	0	0	0
Transition time (t_j)		5	5	4	5	5	5	2	5	3	5	1	5	3	3

In this study, maintenance time M_i is defined by $M_i = z^*(i)$ where $z^*(i)$ is the optimal value of following problem DP(i):

$$\underset{u_j \in \{0,1\}}{\text{Min}} \quad z(i) = \sum_{j=1}^{n} t_j \cdot u_j \tag{1}$$

$$\text{s.t.} \sum_{j=1}^{n} T_{kj} u_j \geq 0, \quad \forall k \neq i \tag{2}$$

$$\sum_{j=1}^{n} T_{ij} u_j = 1 \tag{3}$$

where t_j, which is listed in the last row of Table 1, is time for performing disassembly and reassembly transition j and T_{kj} is an element of the transition matrix T for row i and column j. T is an $m \times n$ matrix. The positive value of t_1 is a constant setup and replacement time charged for any module, since u_1 is always 1 in feasible solutions. The objective function (1) minimizes time of all transitions. The transitions satisfying inequality (2) guarantee their feasibility since no module can appear without disassembling a bigger module. By Equation (3), target module i is selectively obtained without further disassembly.

2.2. **Cost and availability during system maintenance cycle.** This study adopts common definitions for maintenance cost and availability in reliability theory [3]. In this field, maintenance cost is often defined by cost per time unit, availability is defined as fraction of time for which a system operates (uptime) over total time. This study computes time-averaged cost and availability over system maintenance cycle τ, after which the whole system is renewed with new modules and the same maintenance schedule is repeated.

A module maintenance cycle is simply $L_i + M_i$. In a system level, however, it is not as simple as a module. In Figure 1, let τ be the system maintenance cycle of an l-module system. Without loss of generality, the modules are indexed form 1 to l in an order of increasing L_i. By the assumption of a series system, all modules stop during any module is replaced (the shaded intervals). Thus, a later module's replacement point is delayed by prior modules' maintenance. Where a_i is the number of replacements of module i before module l's first maintenance, it actually occurs at $L_l + \sum_{i=1}^{l-1} a_i M_i$, as shown in Figure 1.

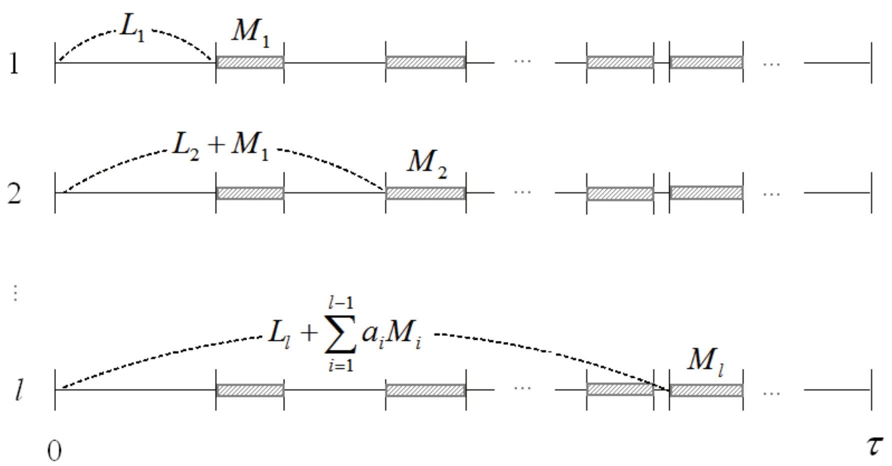

FIGURE 1. System maintenance cycle

Fortunately, such complicated schedule is analytically derived, and the cycle time τ is defined by a closed-form function following Proposition 2.1.

Proposition 2.1. *Where L_i and M_i are maintenance interval and maintenance time of module i, respectively, system maintenance cycle τ and the number of times b_i to replace module i during τ satisfy the following equation where the system consists of l modules.*

$$\tau = \prod_{i=1}^{l} L_i + \sum_{i=1}^{l} b_i M_i, \ \textit{where } b_i = \prod_{\forall k \neq i} L_k, \ \forall i. \tag{4}$$

The first and second terms of τ are total uptime and downtime, respectively.

Proof: For module i, if τ is a system cycle time, it repeats its own maintenance cycle integer-multiple times. Let this number of maintenance cycles be b_i. Since the module also stops during other modules' maintenance time $b_k M_k$ for each $k \neq i$, time τ should be summation of its own cycles and this time. Therefore, there exist positive integers $b_1, \ldots, b_l$ that satisfy the following equation.

$$b_i(L_i + M_i) + \sum_{\forall k \neq i} b_k M_k = b_i L_i + \sum_{k=1}^{l} b_k M_k = \tau, \ \forall i. \tag{5}$$

By equating (5) for all i's, the equation $b_1 L_1 = b_2 L_2 = \cdots = b_l L_l$ holds. It is also total uptime of the system during the cycle τ. Such integer b_i's always exist where $b_i L_i$ is a common multiple of L_i's. By substituting $b_i L_i$ in Equation (5) with a trivial common multiple, which is the product of L_i's, Equation (4) is derived. □

The long-run average cost function $TC(\mathbf{x})$ of $\mathbf{x}$, which is a module decomposition vector $\mathbf{x} = \{x_1, \ldots, x_m\}$, is derived from Equation (4). Because maintenance schedule repeats for any integer multiple of a cycle time, a τ_0 defined by Equation (6) is also a cycle time where only l modules are selected for decomposition among m candidates.

$$\tau_0 = \prod_{i=1}^{m} L_i + \sum_{i=1}^{m} \left(\prod_{k=1}^{m} L_k \Big/ L_i \right) M_i x_i. \tag{6}$$

It is equivalent to multiply L_i's, for which $x_i = 0$, to Equation (4). Where module placement cost is C_i, total cost during τ_0 is the summation of $b_i C_i$'s for i's where $x_i = 1$. Then, $TC(\mathbf{x})$ is

$$TC(\mathbf{x}) = \sum_{i=1}^{m} (C_i/L_i)\, x_i \Big/ 1 + \sum_{i=1}^{m} (M_i/L_i) x_i. \tag{7}$$

Next, availability function $A(\mathbf{x})$ is also derived from Equation (6), of which the first and second terms are uptime and downtime of the system, respectively. Then, $A(\mathbf{x})$ is

$$A(\mathbf{x}) = 1 \Big/ 1 + \sum_{i=1}^{m} (M_i/L_i) x_i. \tag{8}$$

2.3. **Feasible module decomposition.** For any solution $\mathbf{x} = \{x_1, \ldots, x_m\}$, it should be a feasible decomposition; every component belongs to one and only one module. The feasibility is guaranteed by constraining $\mathbf{x}$ with the former transition matrix T. By Lemma 2.1, $\mathbf{x}$ constrained by Equation (9) is a feasible decomposition.

Lemma 2.1. *A solution $\mathbf{x} = \{x_1, \ldots, x_m\}$ is a feasible decomposition if and only if there exists a vector $\mathbf{y} = \{y_1, \ldots, y_n\}$, $y_j \in \{0, 1\}$ satisfying the following equation,*

$$y_1 = 1 \ \text{and} \ \sum_{j=1}^{n} T_{ij} y_j = x_i, \ \forall i. \tag{9}$$

Proof: It is proved by dividing if and only if cases.

i) only if: If $\mathbf{x}$ is a feasible decomposition, there exists an assembly sequence by which the modules are assembled into the original system. Where a column vector $T_{\cdot j}$ corresponds to an operation assembling modules with value 1 into a module with value -1, let y_j be 1 if $T_{\cdot j}$ consists of the assembly sequence, and 0 otherwise. Then, summation of $T_{ij} y_j$'s is 1 for a leaf module i. On the other hand, the summation is 0 if module i is an assembly of other modules because one y_j element is 1 for $T_{ij} = -1$. It is also zero for nonparticipating ones.

ii) if: Suppose $\mathbf{x}$ and $\mathbf{y}$ satisfying Equation (9). Let R be a subset of $\{j |\ y_j = 1\}$ where v_i, which is summation of T_{ij}'s for $j \in R$, be a feasible decomposition. A trivial R is $\{1\}$.

Take i where $x_i = 0$ and $v_i = 1$. There exists $k \notin R$ such that $y_k = 1$ and $T_{ik} = -1$. By expanding $R = R \cup \{k\}$, v_i becomes 0. The redefined v_i's are also a feasible decomposition because each modules h where $T_{hk} = 1$ is a disjoint subset of the components in module i. Likewise, repeat expansion of R until there is no $v_i = 1$ for i's where $x_i = 0$.

If the resulting v_i's are 1 for i's where $x_i = 1$, x_i's that equal v_i's respectively are also feasible. Suppose any $v_i = 0$. If $R = \{j | \ y_j = 1\}$, it contradicts Equation (9). Otherwise, if $R' = \{j | \ y_j = 1\} \backslash R$ is not empty, summation of $T_{\cdot j}$'s for R' has at least one negative element since R trivially contains y_1. Then, some v_i's are decreased by expanding R to $R \cup R'$, and then, some v_i's are unequal to x_i's. This case also contradicts Equation (9). □

2.4. **Optimal modular decomposition model.** The optimization model explores a feasible modular decomposition $\mathbf{x}$ that minimizes the long-run average cost $TC(\mathbf{x})$ while assuring $A(\mathbf{x}) \geq A_T$ and satisfying Equation (9). The cost function $TC(\mathbf{x})$ is nonlinear while the other inequalities are linear. Fortunately, however, the optimization model can formulate a linear objective function by Lemma 2.2.

Lemma 2.2. *For two non-zero functions f and g with the same sign, if $g(\mathbf{x}) \leq g(\mathbf{x}')$ for any two vectors $\mathbf{x}$ and $\mathbf{x}'$ satisfying $f(\mathbf{x}) \geq f(\mathbf{x}')$, $\mathbf{x}^*$ minimizing $f(\mathbf{x})$ also minimizes $f(\mathbf{x})/g(\mathbf{x})$.*

Proof: Suppose that there exist $\mathbf{x}'$ such that $\frac{f(\mathbf{x}')}{g(\mathbf{x}')} < \frac{f(\mathbf{x}^*)}{g(\mathbf{x}^*)}$ where $\mathbf{x}^*$ minimizes $f(\mathbf{x})$.

$$1 \leq \frac{f(\mathbf{x}')}{f(\mathbf{x}^*)} < \frac{g(\mathbf{x}')}{g(\mathbf{x}^*)}. \tag{10}$$

By the condition, however, $g(\mathbf{x}') \leq g(\mathbf{x}^*)$ should hold because $f(\mathbf{x}') \geq f(\mathbf{x}^*)$. Since Equation (10) contradicts this condition, such $\mathbf{x}'$ never exists. □

By Lemma 2.2, where a more expensive decomposition may require less disassembly and reassembly time. From these properties, the modular decomposition problem MP is formulated as a linear-integer program as follows:

$$\min_{x_i \in \{0,1\}, y_j \in \{0,1\}} \quad w = \sum_{i=1}^{m} \frac{C_i}{L_i} x_i \tag{11}$$

$$\text{s.t.} \ \sum_{i=1}^{m} \frac{L_i}{M_i} x_i \leq \frac{1}{A_T} - 1, \tag{12}$$

$$x_i - \sum_{j=1}^{n} T_{ij} y_j = 0, \ \forall i \tag{13}$$

$$y_1 = 1, \tag{14}$$

where variable y_j's, which are also decision variables, are used only for guaranteeing feasible decomposition without involving any cost.

3. **Numerical Example.** As a numerical example, recall the example of AFI in Table 1. All DP(i)'s and MP are solved by CPLEX v12.6.2, and the optimal solutions are obtained in a very short time, less than a second.

This system consists of 11 components, and total 20 modules are candidates for replacement units. For each module i, parameter C_i, L_i and M_i values are listed in Table 2. Let C_i's be in thousand dollars, and L_i's and M_i's be hours. M_i's are derived from the optimal solutions of DP(i)'s. This analysis applied two different L_i value sets: L_i^1 and L_i^2, where L_i^1 has diverse operating life time, and L_i^2 has common life time, which is the average of L_i^1 values. The diversity of life time may affect the solution.

Table 2. Parameter values of AFI

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
C_i	65	55	45	37	24	41	36	24	15	3	3	5	4	4	3	1	2	4	2	4
L_i	500	500	500	500	500	500	500	600	600	600	700	900	600	500	900	800	500	600	700	500
M_i	5	9	14	16	19	10	11	16	19	22	19	19	16	14	9	10	10	10	10	10

Table 3. Optimal decompositions

Target Availability A_T	Optimal Decomposition $\mathbf{x}$	Total Cost in \$/hour $TC(\mathbf{x})$	Attained Availability $A(\mathbf{x})$
81%	$A/B/C/D/E/F/G/H/J/K/L$ (11 modules)	55.62	81.10%
84%	$ABC/D/E/F/G/H/J/K/L$ (9 modules)	65.77	84.76%
87%	$ABCD/E/F/G/H/J/K/L$ (8 modules)	74.11	97.10%
90%	$ABCGHJKL/D/E/F$ (4 modules)	92.00	91.19%
93%	$ABCDGHJKL/E/F$ (3 modules)	101.33	93.81%
96%	$ABCDEGHJKL/F$ (2 modules)	113.33	97.28%
99%	$ABCDEFGHJKL$ (1 module)	130.00	99.01%

The problem MP is solved for these values varying the target availability values from 81% to 99% by 3%. Table 3 shows the optimal solutions. The smallest cost (\$55.62/hour) but lowest availability (81.10%) is attained by decomposing the system into individual components, and the largest cost (\$130.00/hour) but highest availability (99.01%) is attained by replacing the system as a whole. Component $F(i = 15)$ remains as an independent module until the system is decomposed into only two modules. It is because its maintenance interval is much longer than other components while its replacement requires the least time among all modules except the whole system.

4. **Conclusions.** In this study, a linear-integer programming model for minimizing the long-run average maintenance cost of modular-replaceable systems while achieving a given availability is proposed. The replacement modules take a trade-off between the cost and availability. Smaller modules achieve lower costs, but lose availability because of the frequent maintenance stops. Vice versa, larger modules achieve higher availability while spending more cost for replacing all the components that still remain life. In this sense, the optimal decomposition may group the components having similar maintenance intervals. It is a combinatorial problem to explore such groups under a constrained disassembly structure.

The proposed model formulates those objectives and constraints in linear and closed-form functions, and solve it as an ordinary integer-programming problem. Because the existing studies simulated maintenance schedules for deriving the system availability and the long-run average cost, a special algorithm was needed to solve this problem, and the simulation time became a hurdle to solve it efficiently. The closed-form functions resolve this trouble. In addition, a transition matrix, which is usually utilized for exploring assembly or disassembly sequences, is first introduced for guaranteeing physically feasible decomposition of replacement modules. These improvements make the problem much more tractable.

This model could be further extended to stochastic problems involving random failure and corrective maintenance. Since module composition changes failure rates, a more complicated model is needed for taking account of the random nature.

Acknowledgment. This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2017R1D1A1B03032176).

REFERENCES

[1] W. Li and H. Pham, An inspection-maintenance model for systems with multiple competing processes, *IEEE Transactions on Reliability*, vol.54, no.2, pp.318-327, 2005.

[2] A. Grall, C. Bérenguer and L. Dieulle, A condition-based maintenance policy for stochastically deteriorating systems, *Reliability Engineering & System Safety*, vol.76, no.2, pp.167-180, 2002.

[3] R. E. Barlow and F. Proschan, *Mathematical Theory of Reliability*, Wiley, New York, USA, 1996.

[4] K. Ishii, Modularity: A key concept in product life-cycle engineering, in *Handbook of Life Cycle Engineering: Concepts, Models and Technologies*, Kluwer, Dordrecht, 1999.

[5] R. Kumar, K. Izui, M. Yoshimura and S. Nishiwaki, Optimal multilevel redundancy allocation in series and series-parallel systems, *Computers & Industrial Engineering*, vol.57, no.1, pp.169-180, 2009.

[6] J. Gershenson, G. Prasad and Y. Zhang, Product modularity: Definitions and benefits, *Journal of Engineering Design*, vol.14, no.3, pp.295-313, 2003.

[7] D. I. Cho and M. Parlar, A survey of maintenance models for multi-unit systems, *European Journal of Operational Research*, vol.51, no.1, pp.1-23, 1991.

[8] H. Wang, A survey of maintenance policies of deteriorating systems, *European Journal of Operational Research*, vol.139, no.3, pp.469-489, 2002.

[9] A. Haurie and P. L'ecuyer, A stochastic control approach to group preventive replacement in a multicomponent system, *IEEE Transactions on Automatic Control*, vol.27, no.2, pp.387-393, 1982.

[10] P. L'ecuyer and A. Haurie, Preventive replacement for multicomponent systems: An opportunistic discrete-time dynamic programming model, *IEEE Transactions on Reliability*, vol.32, no.1, pp.117-118, 1983.

[11] S. Özekici, Optimal periodic replacement of multicomponent reliability systems, *Operations Research*, vol.36, no.4, pp.542-552, 1988.

[12] W. Y. Yun and J. W. Kim, Multi-level redundancy optimization in series systems, *Computers & Industrial Engineering*, vol.46, no.2, pp.337-346, 2004.

[13] W. Y. Yun, Y. M. Song and H.-G. Kim, Multiple multi-level redundancy allocation in series systems, *Reliability Engineering & System Safety*, vol.92, no.3, pp.308-313, 2007.

[14] W.-C. Yeh, A two-stage discrete particle swarm optimization for the problem of multiple multi-level redundancy allocation in series systems, *Expert Systems with Applications*, vol.36, no.5, pp.9192-9200, 2009.

[15] W. C. Yeh, C. Y. Luo, C. M. Lai, C. T. Hsu, Y. Y. Chung and J. S. Lin, Simplified swarm optimization with modular search for the general multi-level redundancy allocation problem in series-parallel systems, *2016 IEEE Congress on Evolutionary Computation*, 2016.

[16] T.-J. Hsieh, Hierarchical redundancy allocation for multi-level reliability systems employing a bacterial-inspired evolutionary algorithm, *Information Sciences*, vol.288, pp.174-193, 2014.

[17] I.-H. Chung, Optimization of redundancy allocation in multi-level system under target availability, *Journal of Korean Society of Quality Management*, vol.41, no.3, pp.413-422, 2013.

[18] Y. J. Han, W. Y. Yun and J. Y. Lee, A simulation-based multi-level redundancy allocation for a multi-level system, *International Journal of Performability Engineering*, vol.11, no.4, pp.357-367, 2015.

[19] A. J. D. Lambert, Determining optimum disassembly sequences in electronic equipment, *Computers & Industrial Engineering*, vol.43, no.3, pp.553-575, 2002.

[20] A. J. D. Lambert, Linear programming in disassembly/clustering sequence generation, *Computers & Industrial Engineering*, vol.36, no.4, pp.723-738, 1999.