Journal of the Korean Institute of Industrial Engineers
Vol. 34, No. 4, pp. 414-424, December 2008.

An Exact Algorithm for Two-Level Disassembly Scheduling

Hwa-Joong Kim[1†] · Dong-Ho Lee[2] · Paul Xirouchakis[3]

[1]Graduate School of Logistics, Inha University, Num-gu, Incheon 402-751, KOREA
[2]Department of Industrial Engineering, Hanyang University, Seongdong-gu, Seoul 133-791, KOREA
[3]Institute of Production and Robotics (STI-IPR-LICP), Swiss Federal Institute of Technology at Lausanne (EPFL), Lausanne, CH-1015, SWITZERLAND

2 수준 분해 일정계획 문제에 대한 최적 알고리듬

김화중[1] · 이동호[2] · Paul Xirouchakis[3]

[1]인하대학교 물류전문대학원 / [2]한양대학교 산업공학과 / [3]스위스 로잔 연방기술원 기계공학과

Disassembly scheduling is the problem of determining the quantity and timing of disassembling used or end-of-life products while satisfying the demand of their parts or components over a given planning horizon. This paper considers the two-level disassembly structure that describes a direct relationship between the used product and its parts or components. To formulate the problem mathematically, we first suggest an integer programming model, and then reformulate it to a dynamic programming model after characterizing properties of optimal solutions. Based on the dynamic programming model, we develop a polynomial exact algorithm and illustrate it with an example problem.

Keywords: Production, Planning and Scheduling, Disassembly System, Dynamic Programming

1. Introduction

Disassembly is defined as a systematic method for separating a product into its constituent parts or components, subassemblies, or other groupings with necessary sorting operations in a non-destructive or marginally destructive way. Due to its importance in the recovery of used or end-of-life products and even in the disposal of hazardous materials, much attention has been given to disassembly. In the meantime, various decision problems, such as design for disassembly, disassembly process planning, and disassembly scheduling, have been emerged in the disassembly area. For literature reviews on these, see Boothroyd and Alting (1992), Jovane *et al*. (1993), Gupta and McLean (1996), Xirouchakis and Kiritsis (1996, 1997), O'Shea *et al*. (1998), Erdös *et al.* (2001), Kang *et al.* (2001, 2002, 2003), Lambert (2003), Lee *et al*. (2001), Moore *et al*. (2001), Santochi *et al*. (2002), Lambert and Gupta (2005), and Kang and Xirouchakis (2006).

This paper focuses on disassembly scheduling, one of the important operational problems in disassembly systems. In general, disassembly scheduling can be defined as the problem of determining the quantity and timing of disassembling used or end-of-life products while satisfying the demand of their parts or components over a given planning horizon. In fact, dis-

Work carried out while at Swiss Federal Institute of Technology at Lausanne.
The partial financial support from the Swiss National Science Foundation project "Planning and Scheduling in Remanufacturing Systems" with contract no 2000-066640 is gratefully acknowledged.
† Corresponding author : Hwa-Joong Kim, Graduate School of Logistics, Inha University, Num-gu, Incheon 402-751, KOREA,
Tel : +82-32-860-8238, Fax : +82-32-860-8223, E-mail : hwa-joong.kim@inha.ac.kr
Received January 17, 2008; Accepted July 2, 2008.

assembly scheduling corresponds to production planning in assembly systems. However, due to the difference in the number of demand sources, disassembly scheduling is known to be more complicated than the ordinary production planning problems (Gupta and Taleb 1994). In other words, parts or components converge into a single demand source of the product in the assembly environment, while in the disassembly environment, the product diverges into multiple demand sources of parts or components. This limits the applicability of the ordinary production planning algorithms to disassembly systems.

The previous research on disassembly scheduling can be classified according to the number of product types and parts commonality. (See Kim *et al*. (2007) for a review of models, algorithms, and future research topics for disassembly scheduling.) Here, the parts commonality implies that used products or their subassemblies share their parts and/or components. For the problem with single product type without parts commonality, Gupta and Taleb (1994) suggest a material requirement planning (MRP) based algorithm without explicit objective function after showing that it is a reversed form of the MRP. Also, Lee *et al.* (2004) suggest a two-stage heuristic algorithm for the objective of minimizing various costs related to disassembly systems, in which an initial solution is obtained using the MRP based algorithm suggested by Gupta and Taleb (1994) and it is improved iteratively while considering the trade-offs among different cost factors. Lee *et al*. (2002) suggest an integer programming model for the case with resource capacity constraints to represent and solve the problem optimally, based on the fact that the problem can be regarded as a reversed form of the multi-level capacitated lot sizing problem. (See Kim *et al*. (2006b, c) for other capacitated models.) For the problem with single product type and parts commonality, Taleb *et al.* (1997) suggest another MRP-like algorithm for the objective of minimizing the number of products to be disassembled. Also, Neuendorf *et al.* (2001) give an improved solution for the example of Taleb *et al.* (1997) using their Petri-net based algorithm. Finally, for the problem with multiple product types with parts commonality, Taleb and Gupta (1997) suggest a heuristic algorithm for the objective of minimizing the disassembly costs of used products, and Kim *et al.* (2003) suggest a linear programming (LP) relaxation heuristic for the objective of minimizing the sum of setup, inventory holding and disassembly operation costs, in which the solutions obtained from the LP relaxation are modified considering the cost changes, and later it is improved by Kim *et al.* (2006a). Also, Lee *et al*. (2004) suggest integer programming models for all the cases and the computational results show that their performance highly depends on problem data such as the number of items and periods, and Langella (2007) recently suggested a heuristic algorithm that modifies the algorithm of Taleb and Gupta (1997) with respect to the limited purchase quantity and disposal options. For a special case of this case, Inderfurth and Langella (2006) consider a single-period version of the problem with two-level disassembly structure as well as random yields of parts obtained by disassembling used products. Here, the two-level structure describes a direct relationship between the used product to be disassembled and its parts or components obtained from disassembling the used product without considering the intermediate subassemblies explicitly.

In this paper, we consider a deterministic version of the problem, which was considered by Inderfurth and Langella (2006). Also, we consider the problem with single product type and therefore, the parts commonality does not exist in the disassembly structure. To solve the problem, we suggest an exact algorithm that can give optimal solutions in polynomial time. Fundamentally, the disassembly scheduling problem considered in this paper is closely related to the economic lot-sizing problem. (For example, see Aggarwal and Park (1993), Federgruen and Tzur (1991), Wagelmans *et al.* (1992), and Wagner and Whitin (1958) for more details on the economic lot-sizing problem.) Compared with the economic lot-sizing problem, however, the problem considered here has some special characteristics. First, as stated earlier, demand occurs in the level of part and/or component, not in the level of product, which results in multiple demand sources. Second, in the disassembly scheduling problem, there are interdependencies between parts (or components) obtained from disassembling the used products. That is, disassembly operations are done to satisfy the demand of parts and/or components, and hence the parts and/or components affect each other.

To describe the problem mathematically, an integer programming model is suggested in the next section. In Section 3, the integer programming model is reformulated as a dynamic programming model after characterizing the properties of optimal solutions, and based on the dynamic programming model, an exact algorithm is suggested by which the optimal solutions can be obtained in polynomial time. Also, the algorithm incorporates an additional property that can reduce the computation time. To illustrate the exact algorithm, an example problem is solved in Section 4. Finally, Section 5 concludes the paper with a summary and discussion of future research.

2. Problem Description

We begin with explaining the two-level disassembly structure. The first level represents a single root item, i.e., the used product itself to be disassembled (ordered), while the second one represents leaf items, i.e., the items not to be disassembled further. Each leaf item is a part, a component or a subassembly that results from disassembling the product. Note that those leaf items are multiple demand sources as described earlier. An example of the two-level disassembly structure is given in <Figure 1>, in which the number in each parenthesis represents the yield of the corresponding leaf item obtained from the root item. In this case, the root item is disassembled into two units of leaf item 1, three units of leaf item 2, and one unit of leaf item 3.

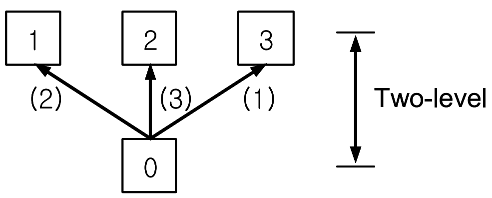

Figure 1. Two-level disassembly structure

For a given two-level disassembly structure, the problem is to determine *the disassembly schedule of the root item while satisfying the demand of leaf items over a given planning horizon for the objective of minimizing the sum of setup and inventory holding costs*. Here, the planning horizon consists of discrete planning periods. Like the production planning problem in assembly systems, the basic trade-off between the setup and the inventory holding costs is an important consideration in disassembly systems. Note that in disassembly systems, the amount of inventory is high due to the divergence nature of the disassembly process, and the setup cost is an important consideration due to the batch process with large manual setup time (Kang *et al.*, 2001, 2002, 2003). The setup cost, which is specific for the root item, implies the cost required when preparing its disassembly. In this paper, the setup cost for the root item in a period is assumed to occur if any disassembly operation is performed in that period. The inventory holding costs occur when leaf items are held in order to satisfy future demand, and they are computed based on the end-of-period inventory. Also, it is assumed that the disassembly operation cost, which is proportional to the labor or machine processing time required for performing the disassembly operation, is constant over the planning periods, and hence it can be eliminated from the problem.

Other assumptions made in this paper are summarized as follows : (a) there is no shortage of the used products ordered, i.e., used products can be obtained whenever they are ordered; (b) demands of parts or components are given and deterministic; (c) backlogging is not allowed, and hence demand should be satisfied on time; and (d) parts and/or components are perfect in quality, i.e., no defective parts or components are considered.

For a clear definition of the problem, the problem considered in this paper is formulated as an integer programming model. Note that the model is not used to solve the problem because of its complexity. Instead, as stated earlier, we reformulate it to a dynamic programming model. Before describing the integer programming model, the notations are summarized below.

i : index for leaf items, $i = 1, 2, \cdots, N$ (The root item is not specified to simplify the notations.)
t : index for periods, $t = 1, 2, \cdots, T$
s_t : setup cost for disassembling the root item in period t
h_{it} : inventory holding cost of one unit of leaf item i in period t
d_{it} : demand of leaf item i in period t
a_i : yield of leaf item i obtained from disassembling one unit of the root item
M : an arbitrary large number
Y_t = 1 if setup for disassembling the root item occurs in period t, and 0 otherwise
I_{it} : inventory level of leaf item i at the end of period t
X_t : disassembly quantity of the root item in period t
l_t : last setup period in t-period subproblem

Now, the integer programming model is given below.

[P] Minimize $\sum_{t=1}^{T} s_t \cdot Y_t + \sum_{i=1}^{N}\sum_{t=1}^{T} h_{it} \cdot I_{it}$

subject to

$$I_{it} = I_{i,t-1} + a_i \cdot X_t - d_{it} \quad \text{for } i = 1, 2, \cdots, N \text{ and } t = 1, 2, \cdots, T \tag{1}$$

$$X_t \le M \cdot Y_t \quad \text{for } t = 1, 2, \cdots, T \tag{2}$$

$$Y_t \in \{0, 1\} \quad \text{for } t = 1, 2, \cdots, T \tag{3}$$

$$I_{it} \ge 0 \quad \text{for } i = 1, 2, \cdots, N \text{ and } t = 1, 2, \cdots, T \tag{4}$$

$$X_t \ge 0 \text{ and integer} \quad \text{for } t = 1, 2, \cdots, T \tag{5}$$

The objective function denotes the sum of setup and inventory holding costs. Constraint (1) defines the inventory level of leaf items at the end of each period, called the inventory flow conservation constraint, and ensures that at the end of each period, the inventory is equal to that at the end of the previous period, increased by the disassembly quantity of the root item multiplied by the yield from the root item in that period, and decreased by the demand quantity in that period. A schematic description of the inventory flow conservation constraint is shown in <Figure 2>. Note that no inventory flow conservation constraint is needed for the root item because its surplus-inventory results in unnecessary increase in the total cost. Constraint (2) guarantees that a setup cost in a period is incurred whenever there is at least one disassembly operation at that period. Constraints (3), (4), and (5) represent the conditions on decision variables. In particular, constraint (4) ensures that backlogging is not allowed.

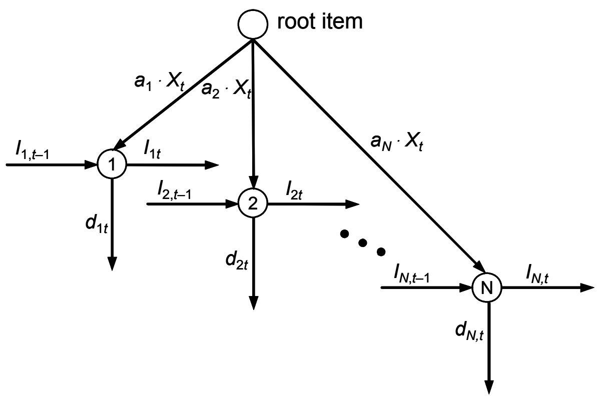

Figure 2. Inventory flow conservation in period t

3. Polynomial Exact Algorithm

This section presents the exact algorithm that can give optimal solutions in polynomial time. Before describing the exact algorithm, two properties that characterize the optimal solutions are derived to reduce the solution space, and using the two properties, the integer programming model [P] described in Section 2 is transformed into a simple dynamic program. Then, based on the dynamic program, the exact algorithm is explained which incorporates an additional property that can reduce the computation times.

The two theorems that characterize the optimal solutions are given below.

Theorem 1 : *For the disassembly scheduling problem [P], there exists an optimal solution that satisfies*

$$X_t \cdot \min_{i=1,2,\cdots,N} \left\lfloor \frac{I_{i,t-1}}{a_i} \right\rfloor = 0 \quad \textit{for } t = 1, 2, \cdots, T, \tag{6}$$

where $\lfloor \bullet \rfloor$ *denotes the largest integer that is less than or equal to* $\bullet$.

Proof : Suppose that there is an optimal solution X_j for $j = 1, 2, \cdots, T$ such that

$$X_t \cdot \min_{i=1,2,\cdots,N} \left\lfloor \frac{I_{i,t-1}}{a_i} \right\rfloor > 0,$$

i.e., $X_t > 0$ and $I_{i,t-1} \ge a_i$ for all $i = 1, 2, \cdots, N$, in period t. Let $k < t$ be the period such that $X_k > 0$ and $X_j = 0$ for $j = k + 1, \cdots, t - 1$ and set

$$L = \min_{i=1,2,\cdots,N} \left\lfloor \frac{I_{i,t-1}}{a_i} \right\rfloor.$$

Consider an alternative solution X'_j, $j = 1, 2, \cdots, T$, in which $X'_k = X_k - L$ and $X'_j = X_j$ for all $j \neq k$, such that

$$\min_{i=1,2,\cdots,N} \left\lfloor \frac{I'_{i,t-1}}{a_i} \right\rfloor = 0$$

and hence the alternative solution satisfies (6) in period t. Here, I'_{it} is the inventory level of item i in period t corresponding to the alternative solution. In other words, we reduce the disassembly quantity in period k as much as L so that (6) is satisfied. Then, the alternative solution reduces the sum of the inventory holding costs of the leaf items from period k to $t - 1$ corresponding to L, i.e.,

$$\sum_{i=1}^{N}\sum_{u=k}^{t-1} h_{iu} \cdot a_i \cdot L .$$

From this, we can see that the current solution X_j, j = 1, 2, ⋯, T, can be improved and hence is not optimal. This leads to a contradiction. ■

Theorem 1 is an extension of the zero-inventory property of Wagner and Whitin (1958). Note that this theorem can be applied to the case with zero initial inventories or with $\min_{i=1,2,\cdots,N} \lfloor I_{i0} / a_i \rfloor = 0$. However, in the case with positive initial inventories satisfying

$$\min_{i=1,2,\cdots,N} \left\lfloor \frac{I_{i0}}{a_i} \right\rfloor > 0 ,$$

the method suggested by Zabel (1964) can be used after a simple modification. The basic idea is as follows : (a) for leaf item i with $I_{i0} \le d_{i1}$, set its initial inventory to zero and its demand in period 1 to d_{i1}-I_{i0}; (b) for leaf item i with $I_{i0} > d_{i1}$, there is a period t such that $\sum_{k=1}^{t} d_{ik} < I_{i0} \le \sum_{k=1}^{t+1} d_{ik}$. Then, set its initial inventory and its demands over periods 1, 2, ⋯, t to zero, and its demand in period t + 1 to $\sum_{k=1}^{t+1} d_{ik} - I_{i0}$. Then, the original problem with positive initial inventories can be transformed into the one with zero initial inventories.

Theorem 1 implies that the disassembly quantity in period t, X_t, is either zero or the quantity to satisfy the demands of all leaf items from period t to some period k, $t \le k \le T$. Note that the demands of the leaf items can be satisfied by those obtained from disassembling the root item by the amount to satisfy the largest one among the demands of all leaf items. More specifically, the exact disassembly quantity in a certain period can be specified using the following theorem.

Theorem 2 : *In the optimal solution* of *the disassembly scheduling problem* [P],

$$X_t = \max_{i=1,2,\cdots,N} \left\lceil \frac{\sum_{k=t}^{T} d_{ik} - I_{i,t-1}}{a_i} \right\rceil \text{ and}$$
$$X_j = 0, \text{ for } t < j \le T,$$

where $t = \max_{k=1,2,\ldots,T} \{k \mid X_k > 0\}$ *and* $\lceil \bullet \rceil$ *denotes the smallest integer that is greater than or equal to* $\bullet$.

Proof : The net demand of leaf item i that should be satisfied from period t to T is

$$\sum_{k=t}^{T} d_{ik} - I_{i,t-1}.$$

Then, the optimal disassembly quantity in period t, X_t, can be determined as the maximum among the values obtained from dividing the net demand of all leaf items by the yield of each leaf item from the root item and then rounding up. More formally,

$$X_t = \max_{i=1,2,\cdots,N} \left\lceil \frac{\sum_{k=t}^{T} d_{ik} - I_{i,t-1}}{a_i} \right\rceil .$$

Note that other values of X_t (except for the above) do not satisfy Theorem 1. Therefore, we can see that the optimal disassembly quantities in the other periods should be all zero, i.e., $X_j = 0$ for $t < j \le T$. ■

Theorem 2 specifies the disassembly quantity to satisfy the demands from period t to T. In other words, if we set X_t to the quantity specified in Theorem 2, the demands of all leaf items from period t to T can be satisfied automatically. However, it may result in indispensable surplus-inventories of the leaf items since the requirement of each leaf item should be divided by the yield of the corresponding leaf item. Note that the existence of the surplus-inventories is one of the differences between the problem considered here and the conventional economic lot-sizing problem.

Now, from Theorem 2, we can see that the problem can be decomposed into subproblems. For a given problem, suppose that the last setup occurs in period t, $2 \le t \le T$, i.e.,

$$X_t = \max_{i=1,2,\cdots,N} \left\lceil \frac{\sum_{k=t}^{T} d_{ik} - I_{i,t-1}}{a_i} \right\rceil \text{ and}$$
$$X_j = 0 \text{ for } j = t+1, \cdots, T.$$

Then, the entire problem can be solved after decomposing it into one from period 1 to t-1 and the other from period t to T. Here, according to Theorem 1, the subproblem from period 1 to t-1 should satisfy

$$\min_{i=1,2,\cdots,N} \left\lfloor \frac{I_{i,t-1}}{a_i} \right\rfloor = 0 \text{ so that } X_t \min_{i=1,2,\cdots,N} \left\lfloor \frac{I_{i,t-1}}{a_i} \right\rfloor = 0.$$

To determine the optimal solution of the entire problem, T alternatives of the decomposition should be compared, and the best one should be selected since the last setup may occur in one of the periods from 1 to T.

Based on the decomposition described above, we can transform the integer programming model [P] into a dynamic programming model. Note that the straightforward method to obtain the optimal solution is to enumerate 2^T combinations of either performing disassembly or not in each period. Instead, in this paper, we develop an exact algorithm that can solve the problem in polynomial time. That is, each subproblem obtained from the decomposition is solved recursively, starting from the 1-period subproblem and ending at the T-period subproblem. Here, the t-period subproblem denotes the subproblem from period 1 to t.

Now, the dynamic programming model is explained. Suppose that demands of the leaf items from period j to t ($1 \leq j \leq t$) are satisfied in period j. According to Theorem 2, the disassembly quantity in period j, X_j, can be determined as

$$X_j = \max_{i=1,2,\cdots,N} \left\lceil \frac{\sum_{k=j}^{t} d_{ik} - I_{i,j-1}}{a_i} \right\rceil, \tag{7}$$

where the inventory $I_{i,j-1}$ can be obtained from solving the (j-1)-period subproblem. Also, the inventory level of leaf item i at the end of period k ($j \leq k \leq t$) can be represented as

$$I_{ik} = \begin{cases} I_{i,k-1} + a_i \cdot X_j - d_{ik} & \text{for } k = j \\ I_{i,k-1} - d_{ik} & \text{for } k = j+1, j+2, \cdots, t. \end{cases} \tag{8}$$

Given (7) and (8), the dynamic programming model can be represented as follows. In the formulation, $F(t)$ denotes the optimal cost function for period 1 through t.

[P'] $$F(t) = \min_{1 \le j \le t} \left\{ \sum_{i=1}^{N} \sum_{k=j}^{t} h_{ik} \cdot I_{ik} + s_j + F(j-1) \right\},$$

where $F(0) = 0$. This recursive function consists of the inventory holding costs (of all leaf items) occurred from period j to t, the setup cost (of the root item) occurred in period j, and the optimal cost function for period 1 through j-1. Therefore, the optimal solution can be obtained by computing $F(t)$ recursively, starting from period 1 and ending at period T.

Now, we can derive another property that can reduce the amount of computation when obtaining the optimal solution. The following theorem summarizes this property.

Theorem 3 : *In the dynamic program* [P'], *suppose that an optimal solution for the u-period subproblem, i.e., the subproblem from period* 1 *to u, is obtained with the last setup in period* l_u. *Then, for the v-period subproblem* ($v > u$), *there exists an optimal solution that has the last setup at period* $l_v \geq l_u$,

Proof : Suppose that an optimal solution for a v-period subproblem is obtained with the last setup in period $j < l_u$. Then, from the definition of optimal function,

$$\sum_{i=1}^{N} \sum_{k=j}^{v} h_{ik} \cdot I_{ik} + s_j + F(j-1) \\ \leq \sum_{i=1}^{N} \sum_{k=l_u}^{v} h_{ik} \cdot I_{ik} + s_{l_u} + F(l_u - 1).$$

Separating these functions by period u,

$$\sum_{i=1}^{N} \sum_{k=j}^{u} h_{ik} \cdot I_{ik} + \sum_{i=1}^{N} \sum_{k=u+1}^{v} h_{ik} \cdot I_{ik} + s_j + F(j-1) \\ \leq \sum_{i=1}^{N} \sum_{k=l_u}^{u} h_{ik} \cdot I_{ik} + \sum_{i=1}^{N} \sum_{k=u+1}^{v} h_{ik} \cdot I_{ik} + s_{l_u} + F(l_u - 1).$$

Taking out the second term in each function and rewriting results in

$$\sum_{i=1}^{N} \sum_{k=j}^{u} h_{ik} \cdot I_{ik} + s_j + F(j-1) \\ \leq \sum_{i=1}^{N} \sum_{k=l_u}^{u} h_{ik} \cdot I_{ik} + s_{l_u} + F(l_u - 1). \tag{9}$$

However, (9) violates the optimality that the last setup of the u-subproblem occurs in period l_u, which leads to a contradiction. ■

Theorem 3, which is an extension of the planning horizon theorem of Wagner and Whitin (1958), can reduce the amount of computation in such a way that we need to consider only periods l_u, $l_u + 1$, $\cdots$, v (without periods 1, 2, $\cdots$, l_u-1) when the optimal solution is obtained for the v-period subproblem. Here, l_u is the last setup period for the u-period subproblem ($u < v$). For example, suppose that the last setup occurs in period 3, i.e., $l_4 = 3$, in the 4-period subproblem. Then, when solving the 5-period subproblem, it is needed to consider only periods 3, 4 and 5 and the remaining periods 1 and 2 can be excluded from consideration.

Now, the exact algorithm is summarized below.

Procedure 1. (Exact algorithm)

Step 1 : Set $t = 1$, $F(0) = 0$, and $l_0 = 1$.

Step 2 : For a t-period subproblem, do the following steps :

1) Set $j = l_{\tilde{t}1}$ (by Theorem 3). (Demands of all leaf items from period j to t are satisfied in period j.)
2) Calculate the disassembly quantity X_j in period j using

$$X_j = \max_{i=1,2,\cdots,N} \left\lceil \frac{\sum_{k=j}^{t} d_{ik} - I_{i,j-1}}{a_i} \right\rceil$$

3) Calculate the inventory levels of all leaf items from period j to t using

$$I_{ik} = \begin{cases} I_{i,k-1} + a_i \cdot X_j - d_{ik} & \text{for } k = j \\ I_{i,k-1} - d_{ik} & \text{for } k = j+1, j+1, \cdots, t \end{cases}$$

4) Calculate the total cost C_{jt} when a setup occurs in period j for the t-period subproblem using

$$C_{jt} = \sum_{i=1}^{N} \sum_{k=j}^{t} h_{ik} \cdot I_{ik} + s_j + F(j-1)$$

5) Set $j = j + 1$. If $j > t$, go to Step 3. Otherwise, go (2).

Step 3 : Calculate the optimal solution value and the last setup period l_t using

$$F(t) = \min_{k=l_{t-1}, l_{t-1}+1, \cdots, t} C_{kt} \text{ and } l_t = \min_{k=l_{t-1}, l_{t-1}+1, \cdots, t} \{k \mid C_{kt}\},$$

and update I_{it}, $i = 1, 2, \cdots, N$.

Step 4 : Set $t = t + 1$. If $t \leq T$, go to Step 2, and otherwise, stop.

The exact algorithm given above has a polynomial time bound. More specifically, in a t-period subproblem, the followings can be calculated in $O(N \cdot T^2)$: disassembly quantity X_j in period j; the total cost C_{jt} when a setup occurs in period j; and the inventory levels for all leaf items from period j to t. Here, N and T denote the number of leaf items and the number of periods, respectively. Also, the optimal solution value $F(t)$ and the last setup period l_t are calculated in $O(T)$ time. Therefore, the complexity of the algorithm becomes $O(N \cdot T^3)$ since there are T subproblems in total.

In fact, the exact algorithm given above extends the well-known Wagner-Whitin algorithm according to the characteristics of the disassembly problem considered in this paper, i.e., multiple demand sources and interdependency among leaf items. Note that the Wagner-Whitin algorithm is only for the assembly systems and cannot be applied directly to the problem considered here. Richer and Sombrutzki (2000) and Richer and Weber (2001) suggest the reverse Wagner-Whitin algorithms for the problems in which used products are returned to be stored and remanufactured to satisfy demands of the product over a planning horizon. Their algorithms, however, use the Wagner-Whitin algorithm as it is or after a simple modification, since their problems have the characteristics of the assembly systems, i.e., single demand source and hence no interdependency among items. Unlike these, in this paper, we derive an extended zero-inventory property according to the unique characteristics of the disassembly systems, and suggest a new exact algorithm. Note that the reverse Wagner-Whitin algorithms could be applied to the problems with single demand source.

4. An Example

In this section, the exact algorithm is illustrated using an example. The example has the disassembly structure described in <Figure 1> and the demand of leaf items (d_{it}), setup costs (s_t), and inventory holding costs of leaf items (h_{it}) are summarized in <Table 1>. The initial inventory level of each leaf item is set to 0.

We begin the exact algorithm from the 1-period subproblem. First, the disassembly quantity X_1 can be determined as

$$X_1 = \max_{i=1,2,3} \left\lceil \frac{d_{i1} - I_{i0}}{a_i} \right\rceil = \max\left\{ \left\lceil \frac{50-0}{2} \right\rceil, \left\lceil \frac{37-0}{3} \right\rceil, \left\lceil \frac{45-0}{1} \right\rceil \right\} = 45.$$

Then, the inventory levels of all leaf items, i.e., I_{i1}, for $i = 1, 2$, and 3, can be calculated as

$$I_{1,1} = I_{1,0} + a_1 \cdot X_1 - d_{1,1} = 0 + 2 \cdot 45 - 50 = 40,$$

Table 1. Problem data of the example

	Planning period (t)									
	1	2	3	4	5	6	7	8	9	10
d_{1t}	50	153	36	61	176	26	134	67	121	67
d_{2t}	37	46	198	134	68	155	29	130	44	149
d_{3t}	45	34	44	28	43	67	48	34	66	56
s_t	1050	1020	1020	1010	1080	1140	1050	1060	1190	1100
h_{1t}	1	5	3	4	2	3	1	1	3	1
h_{2t}	3	2	3	5	5	1	3	2	2	4
h_{3t}	3	1	2	4	3	2	5	3	1	2

$I_{2,1} = I_{2,0} + a_2 \cdot X_1 - d_{2,1} = 0 + 3 \cdot 45 - 37 = 98$, and
$I_{3,1} = I_{3,0} + a_3 \cdot X_1 - d_{3,1} = 0 + 1 \cdot 45 - 45 = 0$.

Now, the optimal solution value $F(1)$ of this subproblem, i.e., the total cost C_{11} of the 1-period subproblem, is as follows.

$$F(1) = C_{1,1} = \sum_{i=1}^{3} h_{i1} \cdot I_{i1} + s_1 + F(0)$$
$$= 1 \cdot 40 + 3 \cdot 98 + 3 \cdot 0 + 1050 + 0 = 1384$$

In the 1-period subproblem, the last setup period becomes 1, i.e., $l_1 = 1$, since disassembly can take place only at period 1.

Next, we consider the 2-period subproblem. In this problem, two alternatives are available : (a) disassembly operation is done only at period 1; and (b) disassembly operations are done at periods 1 and 2, respectively. In the former case, the disassembly quantity in period 1 (X_1) should cover the demands in periods 1 and 2, and can be determined as

$$X_1 = \max\left\{ \left\lceil \frac{50+153}{2} \right\rceil, \left\lceil \frac{37+46}{3} \right\rceil, \left\lceil \frac{45+34}{1} \right\rceil \right\}$$
$$= 102 .$$

Also, the resulting inventory levels of all leaf items in periods 1 and 2 can be calculated as

$I_{1,1} = 0 + 2 \cdot 102 - 50 = 154$,
$I_{1,2} = 154 - 2 \cdot 0 - 153 = 1$,
$I_{2,1} = 0 + 3 \cdot 102 - 37 = 269$,
$I_{2,2} = 269 - 3 \cdot 0 - 46 = 223$,
$I_{3,1} = 0 + 1 \cdot 102 - 45 = 57$, and
$I_{3,2} = 57 - 1 \cdot 0 - 34 = 23$.

In the latter case, the disassembly quantity in period 1, X_1, is the same as that of the 1-period subproblem, and the disassembly quantity in period 2, X_2, can be determined as

$$X_2 = \max\left\{ \left\lceil \frac{153-40}{2} \right\rceil, \left\lceil \frac{46-98}{3} \right\rceil, \left\lceil \frac{34-0}{1} \right\rceil \right\} = 57,$$

where 40, 98, and 0 are the surplus-inventories resulting from solving the 1-period problem. Also, the resulting inventory levels become

$I_{1,2} = 40 + 2 \cdot 57 - 153 = 1$,
$I_{2,2} = 98 + 3 \cdot 57 - 46 = 223$, and
$I_{3,2} = 0 + 1 \cdot 57 - 34 = 23$.

Finally, the optimal policy in the 2-period subproblem is to choose the period with the lowest total cost between the two options. That is,

$$F(2) = \min \begin{cases} C_{1,2} = \sum_{i=1}^{3} \sum_{t=1}^{2} h_{it} \cdot I_{it} + s_1 + F(0) \\ \quad = 1 \cdot 154 + 3 \cdot 269 + 3 \cdot 57 + 5 \cdot 1 \\ \quad\quad + 2 \cdot 223 + 1 \cdot 23 + 1050 + 0 = 2656, \\ C_{2,2} = \sum_{i=1}^{3} h_{i2} \cdot I_{i2} + s_2 + F(1) \\ \quad = 5 \cdot 1 + 2 \cdot 223 + 1 \cdot 23 \\ \quad\quad + 1020 + 1384 = 2878. \end{cases}$$

From the above, we can see that the optimal policy is to select the case that disassembly operation takes place at period 1. Therefore, the last setup period in the 2-period subproblem is 2, i.e., $l_2 = 1$.

The subproblems for the remaining periods, i.e., from 3-period to 10-period subproblems, were solved using the same procedure described above, and the results are summarized in <Table 2> that shows the total costs, the optimal solution values, and the last setup period for each subproblem, i.e., C_{jt}, $F(t)$, and l_t for $j \leq t = 1, 2, \cdots, T$. Note that the dashes in the table imply the unnecessary calculations to obtain the optimal solution (Theorem 3).

The optimal solution of the example is given in <Table 3>, which shows the disassembly schedule of the root item and the resulting inventory levels of the leaf items. In the optimal solution, we can see that setups occur in periods 1, 3, 5, 7, and 9 with the corresponding disassembly quantities 102, 49, 110, 91 and 113, respectively. Note that there are surplus-inventories at the end of the planning horizon. They can be disposed or used for the next planning horizon, which is one of important practical problems in disassembly systems.

5. Concluding Remarks

We considered the problem of determining the quantity and timing of disassembling used or end-of-life products while satisfying the demands of their parts or components over a given planning horizon. The main focus was the case with a direct relationship between a product and its parts or components, i.e., two-level disassembly structure, for the objective of minimizing the sum of setup and inventory holding costs. The problem was formulated as an integer programming model, which was reformulated as a dynamic programming model after the properties of optimal solutions were characterized. Then, based on the dynamic programming model, an exact algorithm has been demonstrated in which the optimal solutions can be obtained in polynomial time.

Table 2. Solution value of each subproblem for the example

Setup period (j)	t-period subproblem									
	1	2	3	4	5	6	7	8	9	10
1	1384(45)[†]	2656(102)	3592(123)	–[‡]	–	–	–	–	–	–
2		2878(57)	3520(78)	4670(106)	–	–	–	–	–	–
3			3961(21)	4635(49)	9750(136)	–	–	–	–	–
4				4728(28)	8364(115)	–	–	–	–	–
5					7002(87)	7713(110)	10500(167)	–	–	–
6						8347(23)	9880(80)	–	–	–
7							9669(57)	10759(91)	13172(151)	–
8								11275(34)	12728(94)	15659(147)
9									12742(60)	15090(113)
10										15487(53)
F(t)	1384	2656	3520	4635	7002	7713	9669	10759	12728	15090
l_t	1	1	2	3	5	5	7	7	8	9

Note) [†] : total cost(C_{jt}) and disassembly quantity(in parenthesis) when a setup in period j occurs in the t-period subproblem.
[‡] : unnecessary calculations by Theorem 3.

Table 3. Optimal solution of the example

	Planning period(t)									
	1	2	3	4	5	6	7	8	9	10
I_{1t}	154	1	63	2	46	20	68	1	106	39
I_{2t}	269	223	172	38	300	145	389	259	554	405
I_{3t}	57	23	28	0	67	0	43	9	56	0
X_t	102	0	49	0	110	0	91	0	113	0

This research can be extended in several ways. First, it is needed to consider more general problems, such as those with multi-level product structure, multiple product types and/or resource capacity constraints, in which the exact algorithm suggested in this paper may be used to solve their subproblems. In particular, the parts commonality is an important consideration in the case of multiple product types. Second, like other disassembly problems, uncertainties, such as defective parts or components, and stochastic demands and lead times, are important considerations.

References

Aggarwal, A. and Park, J. K. (1993), Improved Algorithms for Economic Lot Size Problems, *Operations Research*, **41**, 549-571.

Boothroyd, G. and Alting, L. (1992), Design for Assembly and Disassembly, *CIRP Annals-Manufacturing Technology*, **41**, 625-636.

Erdös, G., Kis, T., and Xirouchakis, P. (2001), Modeling and Evaluating Product End-of-life Options, *International Journal of Production Research*, **39**, 1203-1220.

Federgruen, A. and Tzur, M. (1991), A Simple Forward Algorithm to Solve General Dynamic Lot sizing Models with n periods in $O(n \log n)$ or $O(n)$ Time, *Management Science*, **37**, 909-925.

Gupta, S. M. and McLean, C. R. (1996). Disassembly of Products, *Computers and Industrial Engineering*, **31**, 225-228.

Gupta, S. M. and Taleb, K. N. (1994), Scheduling Disassembly. *International Journal of Production Research*, **32**, 1857-1886.

Inderfurth, K. and Langella, I. M. (2006), Heuristics for Solving Disassemble-to-Order Problems with Stochastic Yields. *OR Spectrum*, **28**, 71-97.

Jovane, F., Alting, L., Armoillotta, A., Eversheirm, W., Feldmann, K., Seliger, G., and Roth, N. (1993), A Key Issue in Product Life Cycle : Disassembly, *CIRP Annals-Manufacturing Technology*, **42**, 651-658.

Kang, J-G., Lee, D-H., Xirouchakis, P., and Lambert, A. J. D. (2002), Optimal Disassembly Sequencing with Sequence Dependent Operation Times Based on the Directed Graph of Assembly States, *Journal of the Korean Institute of Industrial Engineers*, **28**, 264-273.

Kang, J-G., Lee, D-H., Xirouchakis, P., and Persson, J-G. (2001), Parallel Disassembly Sequencing with Sequence-Dependent Operation Times, *CIRP Annals-Manufacturing Technology*, **50**, 343-346.

Kang, J-G., Lee, D-H., and Xirouchakis, P. (2003), Disassembly Sequencing with Imprecise Data : A Case Study, *International Journal of Industrial Engineering*, **10**, 407-412.

Kang, J-G. and Xirouchakis, P. (2006), Disassembly Sequencing for Maintenance : A Survey, *Proceedings of the Institution of Mechanical Engineers : Journal of Engineering Manufacture-Part B*, **220**, 1697-1716.

Kim, H-J., Lee, D-H., Xirouchakis, P., and Züst, R. (2003), Disassembly Scheduling with Multiple Product Types, *CIRP Annals-Manufacturing Technology*, **52**, 403-406.

Kim, H-J., Lee, D-H., and Xirouchakis, P. (2006a), Two-Phase Heuristic for Disassembly Scheduling with Multiple Product Types and Parts Commonality, *International Journal of Production Research*, **44**, 195-212.

Kim, H-J., Lee, D-H., and Xirouchakis, P. (2006b), A Lagrangean Heuristic Algorithm for Disassembly Scheduling with Capacity Constraints, *Journal of the Operational Research Society*, **57**, 1231-1240.

Kim, H-J., Lee, D-H., and Xirouchakis, P. (2007), Disassembly Scheduling : Literature Review and Future Research Directions, *International Journal of Production Research*, **45**, 4465-4484.

Kim, J-G., Jeon, H-B., Kim, H-J., Lee, D-H., and Xirouchakis, P. (2006c), Disassembly Scheduling with Capacity Constraints : Minimizing the Number of Products Disassembled. *Proceedings of the Institution of Mechanical Engineers : Journal of Engineering Manufacture-Part B*, Journal of Engineering Manufacture, **220**, 1473-1481.

Kongar, E. and Gupta, S. M. (2006), Disassembly to Order System Under Uncertainty, *Omega-International Journal of Management Science*, **34**, 550-561.

Lambert, A. J. D. (2003), Disassembly Sequencing : A Survey. *International Journal of Production Research*, **41**, 3721-3759.

Lambert, A. J. D. and Gupta, S. M. (2005), *Disassembly Modeling for Assembly, Maintenance, Reuse and Recycling*, CRC Press, Florida, USA.

Langella, I. M. (2007), Heuristics for Demand-Driven Disassembly Planning, *Computers and Operations Research*, **34**, 552-577.

Lee, D-H. (2005), Disassembly Scheduling for Products with Assembly Structure, *International Journal of Management Science*, **11**, 63-78.

Lee, D-H., Kang, J-G., and Xirouchakis, P. (2001), Disassembly Planning and Scheduling : Review and Further Research, *Proceedings of the Institution of Mechanical Engineers : Journal of Engineering Manufacture-Part B*, **215**, 695-710.

Lee, D-H., Kim, H-J., Choi, G., and Xirouchakis, P. (2004), Disassembly Scheduling : Integer Programming Models. *Proceedings of the Institution of Mechanical Engineers : Journal of Engineering Manufacture-Part B*, **218**, 1357-1372.

Lee, D-H. and Xirouchakis, P. (2004), A Two-Stage Heuristic for Disassembly Scheduling with Assembly Product Structure, *Journal of the Operational Research Society*, **55**, 287-297.

Lee, D-H., Xirouchakis, P., and Züst, R. (2002), Disassembly Scheduling with Capacity Constraints, *CIRP Annals-Manufacturing Technology*, **51**, 387-390.

Moore, K. E., Güngör, A., and Gupta, S. M. (2001), Petri Net Approach to Disassembly Process Planning for Products with Complex AND/OR Precedence Relationships, *European Journal of Operational Research*, **135**, 428-449.

Neuendorf, K-P., Lee, D-H., Kiritsis, D., and Xirouchakis, P. (2001), Disassembly Scheduling with Parts Commonality Using Petri-nets with Timestamps, *Fundamenta Informati-*

cae, **47**, 295-306.

O'Shea, B., Grewal, S. S., and Kaebernick, H. (1998), State of the Art Literature Survey on Disassembly Planning, *Concurrent Engineering* : *Research and Applications*, **6**, 345-357.

Richter, K. and Sombrutzki, M. (2000), Remanufacturing Planning for the Reverse Wagner/Whitin Models. *European Journal of Operational Research*, **121**, 304-315.

Richter, K. and Weber, J. (2001), The Reverse Wagner/Whitin Model with Variable Manufacturing and Remanufacturing Cost, *International Journal of Production Economics*, **71**, 447-456.

Santochi, M., Dini, G., and Failli, F. (2002), Computer Aided Disassembly Planning : State of the Art and Perspectives, *CIRP Annals- Manufacturing Technology*, **51**, 1-23.

Taleb, K. N. and Gupta, S. M. (1997), Disassembly of Multiple Product Structures, *Computers and Industrial Engineering*, **32**, 949-961.

Taleb, K. N., Gupta, S. M., and Brennan, L. (1997), Disassembly of Complex Product Structures with Parts and Materials Commonality, *Production Planning and Control*, **8**, 255-269.

Wagelmans, A., Hoesel, S. V., and Kolen, A. (1992), Economic Lot Sizing : An $O(n \log n)$ Algorithm that Runs in Linear Time in the Wagner-Whitin Case, *Operations Research*, **40**, 145-156.

Wagner, M. H. and Whitin, T. M. (1958), Dynamic Version of the Economic Lot-Sizing Model, *Management Science*, **5**, 89-96.

Xirouchakis, P. and Kiritsis, D. (1996), A Generic Petri Net Model for Disassembly Planning Processes, *Proceedings of Tools and Methods for Concurrent Engineering*, 523-529.

Xirouchakis, P. and Kiritsis, D. (1997), Petri Net Model for Disassembly Process Planning, *Proceedings of ASME Mechanical Engineering*, 255-262.

Zabel, E. (1964), Some Generalizations of an Inventory Planning Horizon Theorem, *Management Science*, **10**, 465-471.