공통 부품을 포함한 복수 제품들에 대한 분해 일정계획

Disassembly Scheduling with Multiple Product Types and Parts Commonality

Hwa-Joong Kim[1], Dong-Ho Lee[2], and Paul Xirouchakis[1]

[1]Institute of Production and Robotics (STI-IPR-LICP)
Swiss Federal Institute of Technology (EPFL)
Lausanne, CH-1015
SWITZERLAND
e-mails:{hwa-joong.kim, paul.xirouchakis}@epfl.ch

[2]Department of Industrial Engineering
Hanyang University
Sungdong-gu, Seoul 133-791
KOREA
e-mail:leman@hanyang.ac.kr

Abstract

This paper focuses on the problem of determining the quantity and timing of disassembling used products while satisfying the demand of their parts or components over a planning horizon. The case of multiple product types with parts commonality is considered for the objective of minimizing the sum of setup, disassembly operation, and inventory holding costs. A heuristic is suggested, in which an initial solution is obtained using a linear programming relaxation method, and then improved by perturbing the given solution using a dynamic programming approach and a look-ahead check while considering the trade-offs among different cost factors.

1. Introduction

Disassembly has become a major issue for countries and companies due to the obligation to the environment and the society as well as its profitability incurred from reusing or remanufacturing used or end-of-life products. In general, disassembly is an important step in the product recovery process since used or end-of-life products are disassembled before they are recycled or even disposed (Lee *et al.* 2001).

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 planning horizon. In this paper, we consider the case of multiple product types with parts commonality. Here, the parts commonality implies that products or subassemblies share their parts/components. In general, parts commonality gives lower inventory holding costs, operation costs, and setup costs since it allows an alternative use of parts/components across different product types or subassemblies. However, the existence of parts commonality makes the problem more complex since it creates interdependencies among different parts or components (Taleb *et al.* 1997, Taleb and Gupta 1997).

There are two previous research articles on the case of multiple product types with parts commonality. Taleb and Gupta (1997) suggest a heuristic for the problem with two independent objectives of minimizing the number of products to be disassembled and disassembly costs of products. Later, Kim *et al.* (2003) suggest a heuristic algorithm for the problem with the objective of minimizing the sum of the costs occurred during the disassembly process. For previous research on other cases, see Gupta and Taleb (1994), Taleb *et al.* (1997), Neuendorf *et al.* (2001), Lee *et al.* (2002), and Lee and Xirouchakis (2004).

As stated earlier, we consider the case of multiple product types with parts commonality. The objective is to minimize the sum of setup, disassembly operation, and inventory holding costs. To solve the problem, we suggest a two-phase heuristic, in which the first phase generates an initial solution using the linear programming relaxation heuristic suggested by Kim *et al.* (2003), and then the initial solution is improved by perturbing the current solution while considering the trade-offs among cost factors.

2. Problem Description

This section begins with the disassembly structure. In the structure, the root item represents the product

itself to be disassembled and a leaf item is an item not to be disassembled further. Also, a child item denotes an item that has at least one parent and a parent item is an item that has more than one child item. Note that in the structure considered here, there may be two or more root items (multiple product types) and each item may have two or more parents (parts commonality).

Now, the problem considered in this paper can be defined as follows: *for a given disassembly structure with multiple product types and parts commonality, the problem is to determine the timing and quantity of disassembling parent items, while satisfying the demand of leaf items over a planning horizon.* As stated earlier, the objective is to minimize the sum of setup, disassembly operation, and inventory holding costs. The inventory holding cost is assumed to be computed based on the end-of-period inventory. Other assumptions made in this paper are: (a) products are available whenever ordered; (b) demands of parts/components are given and deterministic; (c) backlogging is not allowed; and (d) parts/components are perfect in quality.

The problem considered here can be formulated as an integer program. In the formulation, without loss of generality, all items are numbered with integers 1, 2,…, i_r, …, i_l,…, N. Here, i_r denotes the index for the last root item, and i_l denotes the index for the first leaf item. The notations used are summarized below.

s_i	setup cost of item i
p_i	disassembly operation cost of item i
h_i	inventory holding cost of item i
d_{it}	demand of item i in period t
a_{ij}	yield of item j obtained from disassembling one unit of the item i
I_{i0}	initial inventory of item i
l_i	disassembly lead time of item i
$\Phi(i)$	set of parents of item i
M	arbitrary large number
Y_{it}	= 1 if there is a setup for item i in period t, and 0 otherwise
X_{it}	disassembly quantity of item i in period t
I_{it}	inventory level of item i at the end of period t

Now, the integer program is given below.

[P] Minimize $\sum_{i=1}^{i_l-1}\sum_{t=1}^{T} s_i \cdot Y_{it} + \sum_{i=1}^{i_l-1}\sum_{t=1}^{T} p_i \cdot X_{it} + \sum_{i=i_r+1}^{N}\sum_{t=1}^{T} h_i \cdot I_{it}$

subject to

$$I_{it} = I_{i,t-1} + \sum_{k\in\Phi(i)} a_{ki} \cdot X_{k,t-l_k} - X_{it} \quad \text{for } i = i_r+1,\ldots, i_l-1 \text{ and all } t \quad (1)$$

$$I_{it} = I_{i,t-1} + \sum_{k\in\Phi(i)} a_{ki} \cdot X_{k,t-l_k} - d_{it} \quad \text{for } i = i_l,\ldots, N \text{ and all } t \quad (2)$$

$$X_{it} \le M \cdot Y_{it} \quad \text{for } i = 1,\ldots, i_l-1 \text{ and all } t \quad (3)$$

$$Y_{it} = \{0, 1\} \quad \text{for } i = 1,\ldots, i_l-1 \text{ and all } t \quad (4)$$

$$X_{it} \ge 0 \text{ and integer} \quad \text{for } i = 1,\ldots, i_l-1 \text{ and all } t \quad (5)$$

$$I_{it} \ge 0 \text{ and integer} \quad \text{for } i = i_r+1,\ldots, N \text{ and all } t \quad (6)$$

The objective function denotes the sum of setup, disassembly operation, and inventory holding costs. Constraints (1) and (2) describe the inventory level of non-root items at the end of each period. Constraint (3) guarantees that a setup cost in a period is incurred if there is any disassembly operation at that period.

3. Two-Phase Heuristic

Phase 1: *Solution construction*

An initial solution is obtained using the linear programming relaxation approach (Kim *et al.*, 2003). The LP relaxation algorithm consists of two main steps. In the first step, the problem [P] is solved directly after relaxing the integral constraints (4), (5), and (6), and then the LP solution (with real values) is rounded down. In the second step, the rounded-down solution is modified so that all the original constraints of [P] are satisfied, while considering the changes of setup, disassembly operation, and inventory holding costs. See Kim *et al.* (2003) for more details.

Phase 2: *Improvement*

Given the initial disassembly schedule obtained in the first phase, an improvement is made by changing and evaluating the disassembly schedule of parent items, repeatedly, using a dynamic program and a look-ahead check. Here, the dynamic program is similar to that of Wagner and Whitin (1958) for single item lot-sizing problem, i.e., the problem is decomposed in T subproblems and a new solution is obtained by solving each subproblem recursively in the forward direction. Also, the look-ahead check is performed to consider the cost changes of other items affected by changing the schedule.

Before explaining the dynamic program and look-ahead check in more detail, we first explain the method to change the current disassembly schedule of a parent item. Suppose that the last setup for the item occurs in period u ($u < v$) in the v-period subproblem of parent item k. The change in the current disassembly schedule is done as follows:

$$X'_{ku} = \sum_{j=u}^{v} X_{kj} \text{ and } X_t = 0 \text{ for } t = u+1,\ldots, v.$$

Then, this change results in the changes of the inventory levels of (parent) item k and its child items. Also, it may result in backlogging, i.e., an infeasible solution. This requires methods to check the feasibility of the new disassembly schedule and to calculate the new inventory levels of the corresponding items. The proposition given below describes the

feasibility condition for the new disassembly schedule obtained by the method to change the current disassembly schedule.

Proposition 1. *For a given disassembly schedule of problem [P], a new disassembly schedule of (parent) item k, obtained by the method to change the current disassembly schedule, is feasible if*

$$X'_{ku} - \sum_{j=u}^{t} X_{kj} \le I_{kt} \quad \text{for } t = u, u+1, \ldots, v-1.$$

Proof: The new inventory level of item k, I'_{kt} for $t = u, \ldots, v$, if the last setup for the v-period subproblem is done in period u, can be calculated as follows. (The others remain the same.)

For $t = u$,

$$I'_{ku} = I_{k,u-1} + \sum_{j\in\Phi(k)} a_{jk} \cdot X_{j,u-l_j} - X'_{ku} = I_{ku} - (X'_{ku} - X_{ku}).$$

For $t = u+1, \ldots, v-1$,

$$\begin{aligned} I'_{kt} &= I'_{k,t-1} + \sum_{j\in\Phi(k)} a_{jk} \cdot X_{j,t-l_j} \\ &= (I_{k,t-1} - (X'_{ku} - \sum_{j=u}^{t-1} X_{kj})) + \sum_{j\in\Phi(k)} a_{jk} \cdot X_{j,t-l_j} \\ &= I_{kt} - (X'_{ku} - \sum_{j=u}^{t} X_{kj}). \end{aligned}$$

For $t = v$,

$$\begin{aligned} I'_{kv} &= I'_{k,v-1} + \sum_{j\in\Phi(k)} a_{jk} \cdot X_{j,v-l_j} \\ &= I_{k,v-1} + \sum_{j\in\Phi(k)} a_{jk} \cdot X_{j,v-l_j} - X_{kv} = I_{kv} \end{aligned}$$

Then, the feasibility condition follows from $I'_{kt} \ge 0$ for $t = j, \ldots, t-1$. ■

Also, we explain the method to calculate the inventory levels after the disassembly schedule of parent item k is changed. Note that the change in the disassembly schedule of parent item k affects the inventory levels of the item itself and its child items. First, the new inventory level of parent item k can be calculated as follows:

$$\begin{aligned} I'_{kt} &= I_{kt} - (X'_{ku} - \sum_{j=u}^{t} X_{kj}) \quad \text{for } t = u, \ldots, v-1, \\ I'_{kt} &= I_{kt} \qquad\qquad\quad \text{for } t = 1, \ldots, u-1, v, \ldots, T, \end{aligned}$$

The above equations result from the proof of Proposition 1. Second, the new inventory levels of the child items can be calculated using the following proposition. Note that the benefit of this method is that it can significantly reduce the computation time to calculate the new inventory level without recalculating the inventory levels of all items.

Proposition 2. *The inventory level (I'_{it}) of each child item i, $i \in H(k)$, after the disassembly schedule of item k is changed according to the method to change the current disassembly schedule is*

$$\begin{aligned} I'_{it} &= I_{it} + a_{ki} \cdot (X'_{ku} - \sum_{j=u}^{t-l_k} X_{kj}) \\ &\qquad \text{for all } t = u+l_k, \ldots, v+l_k-1, \\ I'_{it} &= I_{it} \quad \text{for all } t = 1, \ldots, u+l_k-1, v+l_k, \ldots, T, \end{aligned}$$

where $H(k)$ denotes the set of children of item k.

Proof: For each child item i of a parent item k, $i \in H(k)$, the new inventory level (I'_{it}) in each period, after the last setup for the v-period subproblem of the parent item k is done in period u ($u < v$), can be calculated as follows.

For $t = 1, \ldots, u+l_k-1$, $I'_{it} = I_{it}$.

For $t = u+l_k$,

$$\begin{aligned} I'_{i,u+l_k} &= I_{i,u+l_k-1} + \sum_{j\in\Phi(i), j\ne k} a_{ji} \cdot X_{j,u+l_k-l_j} + a_{ki} \cdot X'_{ku} - X_{i,u+l_k} \\ &= (I_{i,u+l_k-1} + \sum_{j\in\Phi(i)} a_{ji} \cdot X_{j,u+l_k-l_j} - X_{i,u+l_k}) + \\ &\quad a_{ki} \cdot (X'_{ku} - X_{ku}) = I_{i,u+l_k} + a_{ki} \cdot (X'_{ku} - X_{ku}) \end{aligned}$$

For $t = u+l_k+1, \ldots, v+l_k-1$,

$$\begin{aligned} I'_{it} &= I'_{i,t-1} + \sum_{j\in\Phi(i), j\ne k} a_{ji} \cdot X_{j,t-l_j} - X_{it} \\ &= (I_{i,t-1} + \sum_{j\in\Phi(i)} a_{ji} \cdot X_{j,t-l_j} - X_{it}) + a_{ki} \cdot (X'_{ku} - \sum_{j=u}^{t-l_k} X_{kj}) \\ &= I_{it} + a_{ki} \cdot (X'_{ku} - \sum_{j=u}^{t-l_k} X_{kj}). \end{aligned}$$

For $t = v+l_k$,

$$\begin{aligned} I'_{i,v+l_k} &= I'_{i,v+l_k-1} + \sum_{j\in\Phi(i), j\ne k} a_{ji} \cdot X_{j,v+l_k-l_j} - X_{i,v+l_k} \\ &= I_{i,v+l_k-1} + \sum_{j\in\Phi(i), j\ne k} a_{ji} \cdot X_{j,v+l_k-l_j} + a_{ki} \cdot X_{kv} - X_{i,v+l_k} \\ &= I_{i,v+l_k}. \end{aligned}$$

Then, $I'_{it} = I_{it}$ for $t = v+l_k+1, \ldots, T$. ■

According to the method to change the current disassembly schedule, we can see that there are v alternatives to change the current disassembly schedule of parent item k, i.e., the last setup can occur in any of the periods $1, \ldots, v$ in the v-period subproblem. To determine the best one, we specify the increase and decrease in the total cost. First, the decrease in the total cost can be represented as

$$B(u,v) = \sum_{t=u}^{v-1} h_k \cdot (X'_{ku} - \sum_{j=u}^{t} X_{kj}) + \sum_{t=u-1}^{v} s_k \cdot \delta(X_{kt})$$

where $\delta(\bullet) = 1$ if $\bullet > 0$, and 0 otherwise. The first term represents the decrease in the inventory holding cost of item k from period u to $v-1$, while the second one represents the decrease in the setup cost of parent item k from period $u-1$ to v. Second, the increase in the total cost can be represented as

$$C(u,v) = \sum_{i\in H(k)} \sum_{t=u+l_k}^{v+l_k-1} h_i \cdot a_{ki} \cdot (X'_{ku} - \sum_{j=u}^{t-l_k} X_{kj}) + s_k \cdot (1 - \delta(X_{ku}))$$

where the first term represents the increase in the inventory holding cost of its child items, while the second term represents the increase in the setup cost of parent item k in period u if $X_{ku} = 0$.

Now, the dynamic program for improving the current disassembly schedule of parent item k can be formulated as

$$\textbf{[DP1]}\ F_k(v) = \min_{1\le u\le v} \left[\max\{0, B(u,v) - C(u,v) + F_k(u-1)\} \right]$$

where $F_k(0) = 0$. In the formulation, $F_k(v)$ denotes the recursive function for the v-period subproblem for parent item k. This function consists of the decrease ($B(u, v)$) and increase ($C(u, v)$) in total cost occurred when the last setup is done in period u, and the

recursive function for the (u–1)-period subproblem.

Now, the look-ahead check is explained. As stated earlier, the look-ahead check is done to consider the cost changes of other items affected by changing the current disassembly schedule. Suppose that the last setup of parent item k is done (fixed) at period u in the v-period subproblem. Then, the disassembly schedule of item k and the inventory levels of item k and its child items can be calculated using the methods described earlier. Also, the set of items affected by changing the disassembly schedule of item k is specified (denoted by $O(k)$), and then, the new disassembly schedules of each of the items in $O(k)$ can be calculated using [DP1]. Finally, the change in the total cost for item k that includes the cost change of the set of items, $i \in O(k)$, can be represented as $B(u,v) - C(u,v) + \sum_{i \in O(k)} A_i$,

where A_i denotes the maximum cost decrease of item i calculated after fixing the last setup period of item k to u. Note that A_i can be obtained by solving [DP1]. Now, [DP1] can be reformulated as follow.

[DP2] $$L_k(v) = \max_{1 \le u \le v} \left[\max \left\{ 0, B(u,v) - C(u,v) + \sum_{i \in O(k)} A_i + L_k(u-1) \right\} \right]$$

where $L_k(0) = 0$. Based on [DP2], the best new disassembly schedule of the parent item k can be obtained by solving each subproblem recursively, starting from period 1 and ending at period T.

The following procedure summarizes the improvement algorithm using [DP2].

Procedure. (Improvement)

Step 1. Let the current solution be the disassembly schedule obtained in the first phase (the LP relaxation heuristic).

Step 2. For each parent item, do the following steps:

1) Set $k = 1$ (Start from the root item 1).
2) Solve the following dynamic program in forward direction (starting from period 1 and ending at period T), and find the new disassembly schedule of item k.

$$L_k(v) = \max_{1 \le u \le v} \left[\max \left\{ 0, B(u,v) - C(u,v) + \sum_{i \in O(k)} A_i + L_k(u-1) \right\} \right]$$

where $L_k(T) = 0$. Here, if the new disassembly schedule makes an improvement, i.e., $L_k(T) > 0$, save the new disassembly schedule.

3) Set $k = k+1$. If $k < i_l$ (the index for the first leaf item), go to (2). Otherwise, go to Step 3.

Step 3. If $L_k(T) = 0$, for all parent items, $k = 1, \ldots, i_l - 1$, i.e., no further improvements can be made for all parent items, stop. Otherwise, go to Step 2.

5. Concluding Remarks

We considered the disassembly scheduling problem, which is the problem of determining the disassembly schedule of used products while satisfying the demands of their individual parts or components over a given planning horizon. The case of multiple product types and parts commonality is considered for the objective of minimizing the sum of setup, disassembly operation, and inventory holding costs. To solve the problem, a two-phase heuristic is suggested in this paper, in which an initial solution is obtained using a linear programming relaxation approach, and then it is improved using a dynamic programming approach and a look-ahead check.

Although the results are not reported in this paper (due to the space limitation), a case study on end-of-life inkjet printers was performed, and the results show that the heuristic can give near optimal solutions within short computation time.

References

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

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

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. 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, *Annals of the CIRP* 51(1), 387-390.

Neuendorf, K.-P., Lee, D.-H., Kiritsis, D., and Xirouchakis P. (2001), Disassembly Scheduling with Parts Commonality using Petri-Nets with Timestamps, *Fundamenta Informaticae* 47, 295-306.

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

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