A heuristic analysis for integrating disassembly planning and scheduling

Sundong Min, De-bi Tsao
Department of Industrial Engineering and Management
Tokyo Institute of Technology Tokyo 152-8550, JAPAN
+81-3-5734-2358, Email {minsdl}@pm.me.titech.ac.jp {cdb}@me.titech.ac.jp

Abstract. Disassembly planning determines how far to disassemble a product, while disassembly scheduling determines when and how much products should be disassembled to satisfy the demand. This paper is concerned with the means by which the two problems can be incorporated into an optimization model. For this purpose, we first generate the disassembly plan matrix (DPM) that includes information of feasible disassembly plans and their corresponding items disassembled. The DPM is then used to mathematically solve the integrated problem, in which the most attractive disassembly plan(s) and schedule are simultaneously determined for a capacitated disassembly facility. Due to the complexity of the mathematical model, we propose a heuristic algorithm that is capable of obtaining (near-) optimal solutions for reasonably sized problems in an acceptable calculation time. The effectiveness of the proposed algorithm is demonstrated by solving randomly generated problems.

Keywords: Disassembly plan matrix, Disassembly planning and scheduling

1. Introduction

The rapid development and improvement of technical products have promoted additional demands and shortened lifetime of products, resulting in increasing disposal of used products. In Europe, eighty hundred tons of old TV sets, computer equipments, radios, and measuring devices, and three million tons of automobile equipment are discarded each year [6].

Disassembly, defined as the process of physically separating a product into its constituent subassemblies and terminal parts, has been considered as an environmentally and economically sound way to achieve the goals of sustainable development. Disassembly planning determines the optimal disassembly sequences and level for single product and/or multiple products, while, disassembly scheduling determines the amount of products to be disassembled to satisfy the demands for disassembled items. While there is a strong interdependency between the two problems, most of the previous works have studied them independently and/or sequentially [2, 3, 5, 7].

To make the disassembly more viable, however, the disassembly planning and scheduling should be well defined and more incorporated especially for a capacitated disassembly facility designed for recovering multiple products. If the two decision problems are considered in an independent way, the generated disassembly plans may not guarantee the feasibility of the corresponding disassembly schedules if the capacity constraints are taken into account [4].

This paper deals with the problem of integrating the disassembly planning and scheduling, in which the used products that are purchased or collected are disassembled in different ways for individual products while satisfying the periodical demands for disassembled items under planned capacities. If the demands cannot be fulfilled by old parts due to the restricted capacity of the facility, the corresponding items new should be purchased to satisfy the demands. Therefore, the objective function of the problem is to minimize the costs required for purchasing used products and new items, disassembling the products, and holding inventories over the whole scheduling periods.

For this purpose, we first generate disassembly plan matrix (DPM) based on AND/OR graph that represents all feasible disassembly sequences for a given product. The DPM is then used to mathematically solve the integrated disassembly planning and scheduling problem considering capacity constraints. Due to the complexity of the mathematical model, we propose a heuristic algorithm that is capable of obtaining (near-) optimal solutions for reasonably sized problems in an acceptable calculation time.

The proposed algorithm can be used to obtain approximate solutions according to the change of market information such as demands and costs. It also can be utilized to support relatively short-term disassembly decisions.

2. Disassembly Representation

2.1 AND/OR graph

It is an important task to generate all feasible disassembly sequences of a product to be considered for disassembly planning and scheduling. For an illustration, consider the product shown in Fig. 1 that consists of four parts: a, b, c and d.

Figure 1. Simple product
Since part c is fixed on part a, we can first separate part d only, or c and d together. Due to the precedence relation between parts b and c, part b cannot be disassembled from part a without the removal of part c.

Homem de Mello and Sanderson [1] introduced an AND/OR graph to represent assembly processes, which is founded to be applicable also to the disassembly sequencing problem. In this paper, we utilize the AND/OR graph to represent the possible disassembly sequences. The two possible disassembly sequences of the simple product are illustrated in the AND/OR graph of Fig. 2. In the figure, square nodes represent the parent product, subassemblies and terminal parts, respectively. For illustration purposes, the items are numbered as 0, 1, 2, ..., 7. The numbered circular nodes represent technically possible disassembly operations denoted as O₁, O₂, ..., O₇, where DT is the disassembly time of each operation, including set up time, i.e., to change disassembly tools.

\[
A_{ij} = \begin{bmatrix}
  e_{11} & e_{12} & \cdots & e_{1j} \\
  e_{21} & e_{22} & \cdots & e_{2j} \\
  \vdots & \vdots & \ddots & \vdots \\
  e_{i1} & e_{i2} & \cdots & e_{ij}
\end{bmatrix}
\]

The row vectors of \( A_i \) represent each disassembly plan \( i \), on the other hand, the column vectors represent the items to be disassembled according to each disassembly plan. If a disassembly plan \( i \) creates an item \( j \), the element \( e_{ij} \) of \( A_i \) is one, otherwise zero. For the product shown in Fig. 1, for instance, we can consider a total of seven feasible disassembly plans: O₁ only, O₁→O₂, O₁→O₂→O₃, O₂ only, O₂→O₃, O₂→O₃→O₄, and O₂→O₃→O₄ (or O₂→O₃→O₄). Fig. 3 illustrates the DPM \( A_i \) of the product.

\[
A_{ij} = \begin{bmatrix}
  1 & 0 & 0 & 0 & 0 & 1 \\
  0 & 1 & 0 & 0 & 0 & 1 \\
  0 & 0 & 0 & 1 & 1 & 1 \\
  0 & 1 & 1 & 0 & 0 & 0 \\
  0 & 0 & 0 & 1 & 1 & 0 \\
  0 & 1 & 0 & 0 & 0 & 1 \\
  0 & 0 & 0 & 1 & 1 & 1 \\
\end{bmatrix}
\]

Figure 3. Disassembly Plan Matrix of the simple product

The \( A_{ij} \) can be then utilized to solve the integrated disassembly planning and scheduling problem for a capacitated disassembly facility. Let \( D_p \) be the demand for item \( j \) in period \( t \). Originally, the demands will be met by disassembling used products purchased. However, new items may be purchased if \( D_p \) cannot be met by disassembled items only, due to the restricted capacity \( MC \). Therefore, the objective function of the problem is to minimize the sum of the purchase cost \( CP \) of unit number of used product, the disassembly cost \( CD-DT \) of plan \( i \), the purchase cost \( Co \) of unit number of new item \( j \), and inventory holding cost \( CH \) of unit number for disassembled item \( j \). Here, \( CD \) is the average disassembly cost per unit operation time (i.e., $/min$). Based on these parameters and variables, the problem can be formulated as an IP model.

\[
\min \left\{ \sum_{t=1}^{T} \sum_{j=1}^{J} (CP+CD \cdot DT \cdot Z_{it}) + \sum_{t=1}^{T} \sum_{i=1}^{I} (CH \cdot Y_{it} + CQ \cdot N_{it}) \right\}
\]

subject to

\[
\sum_{i=1}^{I} A_{ij} \cdot X_{ij} + Y_{ij(t-1)} + N_{it} \geq D_{it} \quad \text{For all } j, t \quad (2)
\]

\[
\sum_{i=1}^{I} A_{ij} \cdot X_{ij} + Y_{ij(t-1)} + N_{it} - D_{it} = Y_{it} \quad \text{For all } j, t \quad (3)
\]

\[
\sum_{i=1}^{I} DT_i \cdot Z_{it} \leq MC \quad \text{For all } j, t \quad (4)
\]

\[
A_{ij} \cdot X_{ij} = Z_{it} \quad \text{For all } i, j, t \quad (5)
\]

\[
X_{it}, Y_{it}, Z_{it}, N_{it} = \{0,1,2,\ldots\} \quad \text{For all } i, j, t \quad (6)
\]

Where, all \( X_{ij} \), \( Z_{it} \), and \( N_{it} \) are decision variables. For each period \( t \), the \( X_{ij} \) denote the disassembled amount of item \( j \) according to disassembly plan \( i \) , the \( Y_{it} \) the inventory amount of disassembled item \( j \), the \( Z_{it} \) the amount of used products disassembled according to plan \( i \), and the \( N_{it} \) the amount of item \( j \) purchased new. Constraint (1) is the objective function of the problem. Constraints (2) imply that the demands should be satisfied by either old items disassembled or new items purchased. Constraints (3) represent the inventory balance of disassembled items. Constraints (4) restrict the disassembly capacity so that the summed operation time cannot exceed the available capacity.
Constraints (5) cover the material valance of disassembly plans and their corresponding items.

3. Heuristic Approach

Since the Integer Program (IP) model suggested in the previous section belongs to the class of NP-complete, we may fail in obtaining the optimal solutions for relatively large problems in a polynomial calculation time. Therefore, we propose a heuristic algorithm to obtain (near-) optimal solutions for the integrated disassembly planning and scheduling problem. Fig. 4 summarizes the framework of the algorithm that consists of two main modules: ‘Forward-schedule’ and ‘Backward-schedule’. The ‘Forward-schedule’ module determines the attractive disassembly plans to perform and the amount of used products to purchase and to disassemble according to the disassembly plans determined. The ‘Backward-schedule’ module checks the reschedule possibility if there are periods where the demands are not satisfied completely due to the restricted capacity of a certain period.

Input $A_i$ and the data

\[\text{Generate an initial solution and compute the objective function value}\]

\[\text{‘Forward-schedule’ module} \quad \text{‘Backward-schedule’ module}\]

\[\text{Repeat the above steps over the scheduling horizons.}\]

Figure 4. Framework of the algorithm.

The algorithm begins with the input of required data such as $A_i$, demand data, and cost parameters. Then an initial solution is generated. Because of the expensive prices of new items, to purchase them to meet all demands provides an upper bound with respect to the objective function value. Thus, the decision can be considered as the initial solution of the problem. Since the objective function of the problem should be minimized, the ‘Forward-schedule’ module first determines the best disassembly plan in reducing the objective function value, by ranking the multiple plans using a certain indicator. The indicator that is used in this study is to estimate the trade-off ratio between costs and benefits with respect to the reduction of the objective function value.

For an illustration, we refer to the example of the simple product and its DPM $A_i$. The first row vector $[1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1]$ of the $A_i$ implies that the first disassembly plan (plan 1) creates items 1 and 7. Suppose there are the demand for items $j$, i.e., $\{3, 5, 0, 9, 7, 8, 2\}$ for a certain period. If we disassemble one unit of the product according to the plan 1, two units of the demands are satisfied because the product is separated into $\{a/b/c\}$ and (d) by the disassembly. Although we have to pay for the cost for executing the disassembly plan and the purchase cost of one unit of the product, we can get the benefits from the execution of the disassembly plan, because we do not need to purchase the corresponding two items new. Similarly, for each disassembly plan $i$, we can compute the trade-off ratio between the costs required for reducing the objective function value and the corresponding. If $F_i$ denotes the value of the trade-off ratio, it can be represented as follows.

\[F_i = \frac{BP_i - \text{Costs for Plan } i}{\text{Benefits from Plan } i (= \text{BP}_{i})}\]

Table 1 summarizes the $F_i$ for all disassembly plans of the simple product. For the simplicity of the analysis, let us assume that the average disassembled cost per min. and the purchase costs of the product and new items are 1, 10 and 50(S), respectively. From Table 1, we consider plan 6 the most attractive plan referred to $i^*$, because it results in the largest value of $F_i(0.85^*)$.

\[\text{Table 1. Example of the } F_i\]

<table>
<thead>
<tr>
<th>Plan</th>
<th>Item 1</th>
<th>CP</th>
<th>$BP_i$</th>
<th>$F_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>1 0 0 0 0 0 1</td>
<td>10</td>
<td>100</td>
<td>0.80</td>
</tr>
<tr>
<td>P2</td>
<td>0 1 0 0 0 1 1</td>
<td>15</td>
<td>10</td>
<td>0.83</td>
</tr>
<tr>
<td>P3</td>
<td>0 0 0 1 1 1 1</td>
<td>30</td>
<td>200</td>
<td>0.80</td>
</tr>
<tr>
<td>P4</td>
<td>0 1 1 0 0 0 0</td>
<td>8</td>
<td>10</td>
<td>0.64</td>
</tr>
<tr>
<td>P5</td>
<td>0 0 1 1 1 0 0</td>
<td>25</td>
<td>100</td>
<td>0.67</td>
</tr>
<tr>
<td>P6</td>
<td>0 1 0 0 0 1 1</td>
<td>13</td>
<td>150</td>
<td>0.85*</td>
</tr>
<tr>
<td>P7</td>
<td>0 0 0 1 1 1 1</td>
<td>28</td>
<td>200</td>
<td>0.81</td>
</tr>
<tr>
<td>Dj</td>
<td>3 5 0 9 7 8 2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The ‘Forward-schedule’ model then determines the amount $Z_i$ of the product to disassemble according to the plan $i^*$. We set $Z_i$ equal to the minimum unit of the demand items that can be satisfied by the execution of the plan $i^*$ (for example, $Z_2^* = 2$ among 5, 8, and 2). This is because we can avoid unnecessary inventory and operation costs that may be occurred by the over-disassembly of the product. Based on these decisions, the demand, inventory, and objective function value are updated. After that, $F_i$ are computed again with respect to the updated demand data. For each period, these procedures are repeated until either all demands are satisfied or the consumed capacity $\sum DT_iZ_i$ exceeds the available capacity $MC_i$ of the period.

On the other hand, if there are any periods that the demands are not completely satisfied due to the restricted capacities, the ‘Backward-schedule’ module then checks the reschedule possibility for their precedent periods that have residual capacity sufficient to disassemble the products more. This reschedule procedure examines from the precedent period that is the nearest to the current period where the shortage of capacity occurred. If the precedent periods have not residual capacity available for the reschedule, the module terminates the reschedule and goes to the next period. Finally, the above procedures of ‘Forward-schedule’ and ‘Backward-
4. Calculation Experiments

To show the effectiveness of the algorithm proposed in this study, computational experiments were carried out. Due to the difficulty of obtaining extensive real data, randomly generated test problems were used in the experiments. A total of five test problems were generated based on the normally distributed demand data. Each of the test problems was examined with respect to two levels of periods (5, and 10) and capacity scenarios (S: short, R: restrictive, and A: abundant).

The capacity scenario ‘S’ corresponds to the set of problems where there may be periods that the demand cannot be fully satisfied due to the limited capacity, while ‘R’ denotes the set of problems where the demands can be fully satisfied by doing the backward-schedule. Finally, ‘A’ implies the set of problems where the demands of all periods can be satisfied without the backward-schedule because of the abundant capacity per each period.

The experiments were run on a SUN Ultrasparc with 7200 MHz CPU with 512 MB of RAM. The IP problems were solved to obtain the optimal solutions using a commercial software package Xpress-MP, while the heuristic algorithm was simulated using Visual Basic language. The response variables examined are the percentage errors of the solutions obtained from the heuristic algorithm, and the CPU times of the algorithm and the IP model.

Table 2 summarizes the results of the experiments. The CPU time of obtaining the optimal solutions for IP problems is ranged from 1 to 5630 CPU seconds depending on the problem characteristics. On the other hand, the time for the heuristic is less than 10 CPU seconds for all test problems. Summarizing these results, the proposed heuristic solution procedure is capable of obtaining approximate solutions in a linear calculation time.

Table 2. Results of the experiments

<table>
<thead>
<tr>
<th>Problem</th>
<th>Period 5</th>
<th>Period 10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>S  R  A</td>
<td>S  R  A</td>
</tr>
<tr>
<td>1</td>
<td>7.04 1.86 0.53</td>
<td>1.08 0.87 0.18</td>
</tr>
<tr>
<td>2</td>
<td>1.29 0.81 0.00</td>
<td>7.11 4.80 0.90</td>
</tr>
<tr>
<td>3</td>
<td>0.38 0.00 0.00</td>
<td>3.64 2.15 2.10</td>
</tr>
<tr>
<td>4</td>
<td>0.00 0.00 0.00</td>
<td>1.83 1.25 0.00</td>
</tr>
<tr>
<td>5</td>
<td>5.74 0.00 0.00</td>
<td>2.62 0.44 0.00</td>
</tr>
</tbody>
</table>

Ave. 2.89 0.53 0.11 3.26 1.90 0.64

* Percentage error between IP solution and heuristic solution

5. Conclusions

This paper has studied the problem of disassembling used products at the end of their useful lives. An attempt to integrate the disassembly planning and scheduling, the disassembly plan matrix (DPM) was generated based on the AND/OR graph for a certain product. The DPM is used to the integrated disassembly problem mathematically. To overcome the complexity of the mathematical model, we proposed a heuristic algorithm for obtaining approximate solutions in an acceptable run time. The algorithm consists of two main modules of 'Forward-schedule' and 'Backward-schedule' and can be used to determine the attractive disassembly plans and the amount of products to disassemble to satisfy the demands at a minimum cost. Comprehensive calculation experiments demonstrated the effectiveness of the solution procedure presented in this paper.

References