ALGORITHMS FOR DISASSEMBLY SCHEDULING

Karim N. Taleb, Northeastern University, Boston, MA 02115, (617)-286-4121
Surendra M. Gupta, Northeastern University, Boston, MA 02115, (617)-373-4846

ABSTRACT

This paper presents the basic structure for two disassembly scheduling algorithms applied to a single product structure. The first algorithm addresses the case when all items in the product structure are unique. The second algorithm accounts for common items.

INTRODUCTION

Disassembly is a relatively new and fast growing trend in manufacturing. BMW, the German car manufacturer, has already opened a dismantling plant in Orlando, Florida, a number of plants in Europe and is planning to open fifteen more by 1995. Volkswagen has also opened a dismantling plant in Europe in 1990 and is planning to open more in the future. American car manufacturers, namely, Chrysler, Ford Motors and General Motors, are also starting to get involved in disassembly and recycling but are still at early stages compared to their European counterpart [Berry, 1991].

Over the last few years, environmental awareness and recycling regulations have been putting pressure on many manufacturers and consumers, forcing them to produce and dispose of products in an environmentally responsible manner. The regulations are becoming more stringent [Berry, 1991] and manufacturers are required to use recycled materials whenever possible [Chynoweth and Rotman, 1993]. Furthermore, they are sometimes required to recycle their products at the end of their useful lives. Product recycling requires disassembly first, so that individual components and materials can be reused or recycled. As a matter of fact, BMW is considering adding $1000 to the price of its cars to cover expenses related to dismantling and recycling [Berry, 1991].

Another current application of disassembly is in dismantling weapons. It is no longer environmentally acceptable to discard weapons and ammunitions by blowing them up, burning them or dumping them in the ocean. The U.S. Defense Department is now hiring military contractors to dismantle and recycle unwanted weapons [Meier, 1993]. Evidently, disassembly has already affected the plants’ applications are projected to grow in the future. operations of some leading industrial firms and its

We address the operational aspect of planning the material requirements for disassembly operations. While MRP is a widely used procedure for production planning, it is assembly oriented and cannot be used for the planning of disassembly operations. As Panisset [1988] pointed out, "Most MRP logic (and the supporting bill of materials) do not provide facilities to plan disassembly."

In an assembly setting, the parts tend to converge to a single demand source as they are moving on the manufacturing floor. This single demand source is the final product, and the material management’s governing principles are constrained by this "convergence" property. Under a disassembly setting, as the parts start moving away from their source of origin, they tend to diverge from each other and we have to deal with a “divergence” property. While these two classes of problems may sound similar, dealing with them requires different approaches.

The structure of the algorithms in the disassembly case is considerably more complicated than that of the assembly case (regular MRP.) This can be attributed to the multiple demand sources in the case of disassembly as opposed to a single demand source in the assembly case.

Commonality is also a prevalent characteristic of products nowadays, due to its proven benefits for manufacturing operations. The two most typical forms of commonality are parts and materials. The benefits of commonality include lower inventory costs, lower unit costs (due to quantity discounts), and the alternative use of parts/materials (new or reclaimed) across several end products.

From an operational perspective, materials commonality has also proved itself to have benefits similar to the ones produced by parts commonality. While new materials are constantly being developed, the trend has been towards a consolidation of materials for most products. This has become a common aspect of new product design since it allows the recycling operations to be much more efficient and less expensive [Noller, 1992]. Sorting and collecting fewer types of materials in higher volumes is more efficient than sorting and collecting more types of materials in lower volumes. The same concept applies to the cost of recycling. Car manufacturers are currently trying to reduce the total number of different plastic types from over thirty to fewer than ten [Berry, 1992]. At BMW, for example, researchers and engineers are hoping to have only five different types of plastic in their cars [Anonymous, 1991]. In fact, some manufacturers overdesign some elements in a product, just to achieve the benefits produced by materials commonality.

Commonality is therefore a major issue when dealing with disassembly and should not be overlooked. Any effort that deals with disassembly should directly address the problems posed by the two forms of commonality. Commonality makes the problem of scheduling disassembly more complex since it adds some freedom to the problem while
creating dependencies between the components of the product structure.

This paper presents two algorithms for scheduling disassembly. The first algorithm assumes that all the parts in the product structure are unique. This algorithm can be applied to a product structure where there is a certain demand for components and a need to know the number of root items to disassemble in order to fulfill the demand for those components. The algorithm determines the quantity and schedule of disassembly of the large item to fulfill the demand for its various parts. The second algorithm addresses the issue of commonality. Unlike the first algorithm, it involves a decision process. A decision rule has been developed and is used in the algorithm.

**DESCRIPTION OF THE FIRST ALGORITHM**

The objective of this algorithm is to determine an ordering schedule for the root item and generate a disassembly schedule for all parent items over the planning horizon. Figure 1 represents a disassembly product structure where the demand occurs at the component levels (items D, E, H, I, J, K, and L) and where only item A is procured. Figure 1 is interpreted as follows: the root item, A, can be disassembled into four units of B and two units of C. The procedure will require one time unit, which is the disassembly lead time denoted by DLT. Similarly, B can be further disassembled into three units of component, D, and one unit of component, E. This operation will take zero time units, etc. The number at the upper right corner above each box in the product structure denotes the item number. We define a module as a collection of a parent item and all its siblings. For example, G-J,K,L is the right upper module in figure 1. The basic idea of the algorithm is to translate the demand occurring at the component level(s) into an equivalent demand for the root item.

This is achieved by translating the requirement from level to level, proceeding one module at a time. Figure 2 shows the basic logic of the algorithm. There are two loops: the module loop and the time loop. The module loop is executed as many times as there are modules in the product structure and the time loop is executed as many times as the length of the planning horizon. For every module, given the net requirements for the sibling items for a certain time period, the disassembly requirement of the parent item is determined such that all the requirements of the siblings are fulfilled. Since the requirements of the siblings are not usually in the proportions of their respective yields from the parent item, some accumulations will occur. The "on hand inventory after disassembly" is therefore determined. Next, the disassembly schedule of the parent item is determined by phasing the disassembly requirements by the DLT of the parent. This disassembly schedule will serve as a gross requirement at the next level where the parent item will become a sibling item in the module. This procedure is repeated until the root item is reached. At that point the MRP logic is used to determine the ordering schedule of the root item, by taking into account the ordering lead time (OLT).

The objective of this algorithm is to determine an ordering schedule for the root item and generate a disassembly schedule for all parent items over the planning horizon when parts commonality is present.
If commonality were not present, then the procurement source for every item will be the parent item. Under commonality however, a decision problem arises that adds complexity to the problem. When the demand for a certain common part is to be satisfied, we are faced with the problem of selecting the procurement source, since a common part has more than one parent, and hence more than one potential procurement source. For every common part, we need to determine the proportion according to which the demand and inventories will be allocated.

**DESCRIPTION OF THE SECOND ALGORITHM**

These proportions are calculated using the “Allocation Percentage” rule. The “Allocation Percentage” for a certain item is the proportion of that item as it occurs in the yield of the product structure. A part which is not common has an “Allocation Percentage” of 100%.

There are three nested loops in the algorithm. The first loop is the “depth loop” and is the outermost loop in figure 3. Once the algorithm moves to a lower level, the previous level will never be visited again. All the information needed, mainly the disassembly schedule, is carried from the previous level. A second loop is nested in the first one and is the “time loop”. Modules at the same level cannot be processed independently of one another for all time periods since they may contain common parts whose inventories have to be adjusted before proceeding to the next time period.

The “time loop” is therefore needed to ensure that the modules are processed “in parallel” and not in a serial manner. Finally, for a certain level and a certain time period, we want to perform the same operations for all the modules. A third loop is therefore needed and the “modules loop” is nested within the time loop (see figure 3).

**CONCLUSIONS**

This paper presented two algorithms for scheduling the disassembly of discrete products. The first algorithm does not allow for parts commonality while the second algorithm accounts for common parts. These algorithms should have useful applications in planning the disassembly of discrete products where the demand occurs on the component level of the product structure. The applications of these algorithms will increase with the rise of public environmental awareness and the increase in governmental regulations requiring disassembly.

**REFERENCES**