

# Determining optimal disassembly and recovery strategies

Ruud H. Teunter\*

March 11, 2004

Econometric Institute Report EI2004-09

## Abstract

We present a stochastic dynamic programming algorithm for determining the optimal disassembly and recovery strategy, given the disassembly tree, the process dependent quality distributions of assemblies, and the quality dependent recovery options and associated profits for assemblies. This algorithm generalizes the one proposed by Krikke et al. [8] in two ways. First, there can be multiple disassembly processes. Second, partial disassembly is allowed. Both generalizations are important for practise.

## 1 Introduction

Disassembly is a systematic method for separating a product into its constituent modules, components, parts, etc. (all to be called assemblies from now on). Since assemblies usually have to be disassembled before they can be recovered, disassembly plays an important role in product recovery (Jovane et al. [7]). Driven by more rigid environmental legislation, societal pressure and economical incentives, many firms have started recovery and disassembly operations in recent years. For instance Air France, Lufthansa, BMW, Volkswagen, Daimler-Crysler, Nissan, Oce, Xerox, and Philips all operate large-scale disassembly/recovery plants.

Planning disassembly and recovery operations can be divided into three steps.

1. Determine all possible disassembly sequences and processes.

---

\*Erasmus University Rotterdam, Econometric Institute, PO Box 1738, 3000 DR Rotterdam, The Netherlands. E-mail: teunter@few.eur.nl.

2. Determine all possible recovery options and the associated profits for each assembly.
3. Determine the optimal disassembly and recovery strategy.

In step 1, the identification of all possible *disassembly sequences* and *disassembly processes* is based on technical and environmental restrictions. A convenient way to present all disassembly options is in a disassembly tree/graph. This tree contains arcs from each assembly to all its subassemblies (Arai et al. [1], Chen et al. [2], Dutta and Woo [3], Lambert [9, 10], Penev and de Ron [12], Pnueli and Zussman [13], Spengler et al. [14], Veerakamolmal et al. [15], Yan and Gu [16], Zussman et al. [17]).

Step 2 is to identify all recovery options (e.g. remanufacturing, recycling, and disposal) and determine the corresponding profits (revenues minus disassembly costs) for each assembly. The feasibility of a recovery options may depend on the quality of an assembly and on commercial and ecological feasibility criteria (Krikke et al. [8]).

In this paper, we will not consider these technical steps 1 and 2. It is assumed that a disassembly tree is given and that the recovery options are known for each assembly. We focus on step 3: determining the optimal disassembly and recovery strategy. Such a strategy specifies the disassembly sequence, the disassembly processes, and the recovery type for the disassembled assemblies.

A number of authors (Erdos et al. [4], Johnson and Wang [5, 6], Krikke et al. [8], Navin-Chandra [11], Penev and de Ron [12], Zussman et al. [17]) have dealt with the problem of finding the optimal disassembly and recovery strategy. The most general version of the problem is analyzed by Krikke et al. [8]. In fact, to the best of our knowledge, they are the only authors that consider variations in the quality of a returned product and its assemblies.

In our opinion, including quality considerations is essential for determining an optimal disassembly and recovery strategy. In most practical situations, the quality of an assembly determines its recovery options. For instance, PC components can typically only be remanufactured if a returned PC is less than 6 months old. Components of older cores have to be recycled or disposed of. In general, the quality distribution (over the different possible states) of an assembly depends on product characteristics such as age and usage as well as on the type of disassembly processes (e.g. destructive or non-destructive) that

are used to retrieve it.

In this paper, we therefore take the stochastic dynamic programming (DP) algorithm for finding the optimal, quality dependent disassembly and recovery strategy proposed by Krikke et al. [8] as a starting point. We generalize this algorithm in two ways. First, there can be multiple disassembly processes. Second, partial disassembly is allowed, i.e., it is not necessary to disassemble all possible subassemblies from an assembly. Both generalizations are important for practise. As mentioned before, there are often multiple processes for disassembly, e.g. non-destructive or destructive. Furthermore, partial disassembly of subassemblies with a high recovery value is often more profitable than complete disassembly.

The remainder of the paper is organized as follows. In Section 2 we present the DP algorithm for finding the optimal disassembly and recovery strategy. In Section 3 we illustrate the algorithm for a specific example. We end with conclusions in Section 4.

## 2 A stochastic dynamic programming algorithm

In this section we present a stochastic dynamic programming (DP) algorithm for calculating the optimal recovery/disposal strategy. This algorithm is a modified version of the one presented by Krikke et al. [8]. It allows partial disassembly and multiple disassembly processes (see Section 1).

The following information is assumed to be given.

- *Disassembly tree/graph.* The first level, i.e. the root, of the tree is a returned product (often referred to in practise as a "core"), and the other levels represent its modules, components, parts etc. Products as well as modules, components, parts etc. are all called *assemblies*. An assembly is called *atomic* if it can not be disassembled any further, and *non-atomic* otherwise. The tree contains arcs from each assembly to all its subassemblies. An example of a disassembly tree is presented in Figure 1 of the next section.
- *Process dependent quality distribution.* For each arc of the disassembly tree, the quality (distribution) for the subassembly conditional on the quality of the assembly

is given for each type of disassembly process.

- *Quality dependent recovery options and profits.* For each assembly, the quality dependent recovery options (remanufacturing, material recycling, energy recycling, disposal) and the associated profits (recovery revenues minus disassembly costs) are given for the assembly as a whole as well as after disassembling any possible set of subassemblies.

The notations that are used are listed in Table 1.

INSERT TABLE 1

The DP algorithm starts in the lowest level  $L$  of the disassembly tree, which contains atomic assemblies only. For all those assemblies  $j$  (and the corresponding quality classes  $q1 \in Q(j)$ ), it finds the optimal recovery option and the associated profit  $f^L(j, q1)$ . It then moves up to level  $L - 1$ . Again, for all the atomic assemblies  $j$  of that level, it finds the optimal recovery option and the associated profit  $f^{L-1}(j, q1)$ . For all the non-atomic assemblies  $j$  of level  $L - 1$ , it finds the optimal combination of a disassembly set (the set  $S$  of subassemblies that is disassembled) and a recovery option (after disassembly of the before mentioned set). This combination maximizes the profit  $f^{L-1}(j, q1, S)$ , which is the sum of the profits associated with all the disassembled subassemblies plus the profit associated with recovery minus the disassembly costs. The algorithm then moves up to level  $L - 2$ . And so on, until the entire recovery/disposal strategy is determined.

In formulas, this means that the profit of an atomic assembly  $j$  with quality  $q1$  in disassembly level  $l$  is

$$f^l(j, q1) = C(j, q1), \quad (1)$$

where

$$C(j, q1) = \max_{r \in R(j, q1)} p(j, q1, r) \quad (2)$$

and the maximizer for  $r$  is the optimal recovery option.

The profit of a non-atomic assembly  $j$  with quality  $q1$  in disassembly level  $l$  is

$$f^l(j, q1) = \max \left\{ C(j, q1), \max_{d \in D, S \subseteq S(j)} C(j, q1|d, S) \right\}, \quad (3)$$

where  $C(j, q1)$  is defined in (2) and

$$C(j, q1|d, S) = \sum_{s \in S} \sum_{q1 \in Q(s)} \Pr(s, q2|j, q1, d) f^{l+1}(s, q2) + \max_{r \in R(j, q1, S, d)} \{p(j, q1, S, r)\} - c(j, q1, S, d).$$

If (3) is maximized by  $C(j, q1)$ , then it is optimal to recover assembly  $j$  as a whole, and the maximizer for  $r$  is the optimal recovery option. Otherwise, the maximizers for  $d$  and  $S$  in (3) and for  $r$  in  $C(j, q1|d, S)$  are the optimal disassembly process, the optimal disassembly set and the optimal recovery option, respectively.

### 3 Example

We consider a product with 5 assemblies (including the product itself) and 3 levels. The disassembly tree is presented in Figure 1.

INSERT FIGURE 1

There are three recovery options: disposal ( $r = 1$ ), recycling ( $r = 2$ ), and remanufacturing ( $r = 3$ ). However, the recycling and remanufacturing option are not available for all assemblies and remanufacturing is quality (1 = high, 2 = low) dependent. Table 2 gives the profits for all available recovery options.

INSERT TABLE 2

There are two types of disassembly for both assembly 1 (the product itself) and for assembly 3: destructive ( $d = 1$ ) and non-destructive ( $d = 2$ ). For both disassembly processes, the quality distribution (high/low) of subassemblies conditional on the quality of the assembly is given in Table 3. The disassemble costs are given in Table 4.

INSERT TABLES 3 and 4

The results of applying the DP algorithm are given in Table 5.

INSERT TABLE 5

So, the optimal policy is as follows (quality 1=high 2=low; disassembly process 1=destructive 2=non-destructive; disposal option 1=disposal 2=recycling 3=remanufacturing).

- Always disassemble 2 and 3 from 1 (returned product) and dispose of what remains. Use non-destructive disassembly if 1 has high quality and destructive disassembly otherwise.
- Always recycle 2.
- If 3 has high quality, disassemble 4 and 5 and recycle the remainder. If 3 has low quality, recycle it as a whole.
- Remanufacture 4 if its quality is high and recycle it otherwise.
- Remanufacture 5 if its quality is high and dispose of it otherwise.

## 4 Conclusion

We considered the problem of determining the optimal disassembly and recovery strategy, given the disassembly tree and information on quality, available disassembly processes, recovery options and recovery profits. A stochastic dynamic programming algorithm was presented that generalizes the one proposed by Krikke et al. [8] in two ways. First, there can be multiple disassembly processes. Second, partial disassembly is allowed. Both generalizations are important for practise.

When the algorithm is applied in practise, it is important that the input information is regularly updated. This holds especially for the recovery options and the associated profits. In the PC industry, for instance, remanufacturing profits decline rapidly with the age of a core, and remanufacturing is typically no longer a (profitable) option if a core is more than 6 months old. So, in this case, updates are required at least once a month.

**Acknowledgement:** The research of Dr. Ruud H. Teunter has been made possible by a fellowship of the Royal Netherlands Academy of Arts and Sciences.

## References

- [1] E. Arai, N. Uchiyama, and M. Igoshi. Disassembly path generation to verify the assemblability of mechanical products. *JSME International Journal - Series C*, 38(4):805–810, 1995.
- [2] S.F. Chen, J.H. Oliver, S.Y. Chou, and L.L. Chen. Parallel disassembly by onion peeling. *Journal of Mechanical Design*, 119(2):267–274, 1997.
- [3] D. Dutta and T.C. Woo. Algorithm for multiple disassembly and parallel assemblies. *Journal of Engineering for Industry*, 117(1):102–109, 1995.
- [4] G. Erdos, T. Kis, and P. Xirouchakis. Modelling and evaluating product end-of-life options. *International Journal of Production Research*, 39(6):1203–1220, 2001.
- [5] M.R. Johnson and M.H. Wang. Planning product disassembly for material recovery opportunities. *International Journal of Production Research*, 33(11):3119–3124, 1995.
- [6] M.R. Johnson and M.H. Wang. Economic evaluation of disassembly operations for recycling, remanufacturing and reuse. *International Journal of Production Research*, 36(12):3227–3252, 1998.
- [7] F. Jovane, L. Alting, A. Armillotta, W. Eversheim, K. Feldmann, and G. Seliger. A key issue in product life cycle: disassembly. *Annals of the CIRP*, 42(2):640–672, 1993.
- [8] H.R. Krikke, A. van Harten, and P.C. Schuur. On a medium term product recovery and disposal strategy for durable assembly products. *International Journal of Production Research*, 36(1):111–139, 1998.
- [9] A.J.D. Lambert. Optimal disassembly of complex products. *International Journal of Production Research*, 35(9):2509–2523, 1997.
- [10] A.J.D. Lambert. Linear programming in disassembly/clustering sequence generation. *Computer & Industrial Engineering*, 36(4):723–738, 1999.

- [11] D. Navin-Chandra. The recovery problem in product design. *Journal of Engineering Design*, 5(1):65–86, 1994.
- [12] K.D. Penev and A.J. de Ron. Determination of a disassembly strategy. *International Journal of Production Research*, 34(2):495–506, 1996.
- [13] Y. Pnueli and E. Zussman. Evaluating the end-of-life value of a product and improving it by redesign. *International Journal of Production Research*, 35(4):921–942, 1997.
- [14] T. Spengler, H. Pueckert, T. Penkuhn, and O. Rentz. Environmental integrated production and recycling management. *European Journal of Operational Research*, 97(2):308–326, 1997.
- [15] P. Veerakamolmal and S.M. Gupta. High-mix/low-volume batch of electronic equipment disassembly. *Computer & Industrial Engineering*, 35(1-2):65–68, 1998.
- [16] X. Yan and P. Gu. A graph based heuristic approach to automated assembly planning. *Flexible Assembly Systems*, 73:97–106, 1994.
- [17] E. Zussman, A. Kriwet, and G. Seliger. Disassembly-oriented assessment methodology to support design for recycling. *Annals of the CIRP*, 43(1):9–14, 1994.



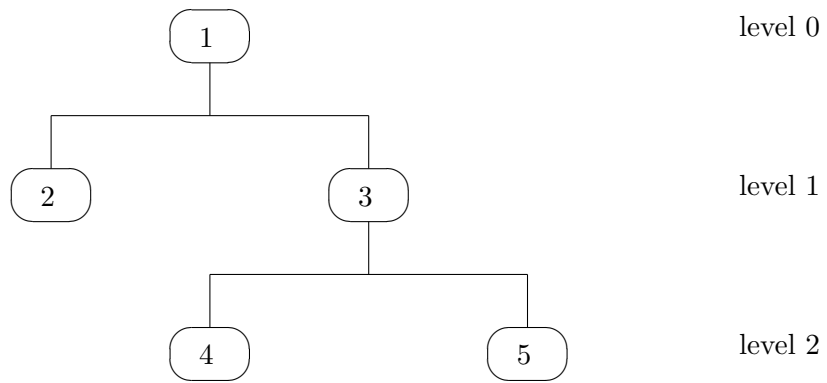


Figure 1: Example of a disassembly tree with 3 levels and 5 assemblies.

$l$	Disassembly level $l = 0, 1, \dots, L$ , $l = 0$ is the root that contains only the product itself
$j$	Assembly identification number $j = 0, 1, \dots, J$ $j = 0$ is the product itself
$l(j)$	Disassembly level of assembly $j$
$Q(j)$	Set of quality classes of assembly $j$
$D(j)$	Set of processes for disassembling assembly $j$
$S(j)$	Set of retrievable subassemblies of assembly $j$ Clearly, $S(j) = \emptyset$ for atomic assemblies
$R(j, q1)$	Set of recovery options for assembly $j$ with quality $q1 \in Q(j)$
$R(j, q1, S, d)$	Set of recovery options for assembly $j$ with quality $q1 \in Q(j)$ after disassembly of the non-empty set of subassemblies $S \subseteq S(j)$ using disassembly process $d$
$c(j, q1, S, d)$	Cost of disassembling from assembly $j$ the non-empty set of subassemblies $S \subseteq S(j)$ using disassembly process $d$
$p(j, q1, r)$	Net profit obtained from recovering assembly $j$ with quality $q1 \in Q(j)$ using recovery option $r \in R(j, q1)$
$p(j, q1, S, r)$	Net profit obtained from recovering assembly $j$ with quality $q1 \in Q(j)$ using recovery option $r \in R(j, q1, S)$ after disassembling the non-empty set of subassemblies $S \subseteq S(j)$ ,
$\Pr(s, q2 j, q1, d)$	Probability that subassembly $s \in S(j)$ of assembly $j$ has quality $q2$ if it is disassembled using process $d$ from an assembly $j$ with quality $q1$

Table 1: Notations.

$j$	$S$	$q1 = 1$			$q1 = 2$		
		$r = 1$	$r = 2$	$r = 3$	$r = 1$	$r = 2$	$r = 3$
1		-4	-	-	-4	-	-
1	{2}	-3	-	-	-3	-	-
1	{3}	-3	-	-	-3	-	-
1	{2, 3}	-2	-	-	-2	-	-
2		-1	2	-	-1	2	-
3		0	5	-	0	5	-
3	{4}	2	4	-	2	4	-
3	{5}	-3	3	-	-3	3	-
3	{4, 5}	-2	1	-	-2	1	-
4		-1	1	5	-1	1	-
5		2	1	10	2	1	-

Table 2: Profits  $p(j, q1, r)$  (no disassembly) and  $p(j, q1, S, r)$  (after disassembly of the set  $S$  of subassemblies) for the numerical example.

$s$	$q2$	$j$	$q1 = 1$		$q1 = 2$	
			$d = 1$	$d = 2$	$d = 1$	$d = 2$
2	1	1	0	0.7	0	0.5
2	2	1	1	0.3	0	0.5
3	1	1	0	0.9	0	0.6
3	2	1	1	0.1	0	0.4
4	1	3	0	0.9	0	0.5
4	2	3	1	0.1	0	0.5
5	1	3	0	0.8	0	0.4
5	2	3	1	0.2	0	0.6

Table 3: Probabilities  $\Pr(s, q2|j, q1, d)$  for the numerical example.

$j$	$S$	$d = 1$	$d = 2$
1	{2}	1	2
1	{3}	2	4
1	{2, 3}	3	5
3	{4}	1	3
3	{5}	1	4
3	{4, 5}	2	6

Table 4: Disassembly costs  $c(j, S, d)$  for the numerical example.

$C(4, 1) =$		$\max\{-1, 1, 5\} = 5$	$= f^2(4, 1)$
$C(4, 2) =$		$\max\{-1, 1, \} = 1$	$= f^2(4, 2)$
$C(5, 1) =$		$\max\{2, 1, 10\} = 10$	$= f^2(5, 1)$
$C(5, 2) =$		$\max\{2, 1, \} = 2$	$= f^2(5, 2)$
$C(2, 1) =$		$\max\{-1, 2, \} = 2$	$= f^1(2, 1)$
$C(2, 2) =$		$\max\{-1, 2, \} = 2$	$= f^1(2, 2)$
$C(3, 1) =$		$\max\{0, 5, \} = 5$	
$C(3, 1 1, \{4\}) =$	$0.0 \times 5 + 1.0 \times 1$	$- 1 + \max\{2, 4, \} = 4$	
$C(3, 1 1, \{5\}) =$	$0.0 \times 10 + 1.0 \times 2$	$- 1 + \max\{-3, 3, \} = 4$	
$C(3, 1 1, \{4, 5\}) =$	$0.0 \times 5 + 1.0 \times 1 + 0.0 \times 10 + 1.0 \times 2$	$- 2 + \max\{-2, 1, \} = 2$	
$C(3, 1 2, \{4\}) =$	$0.9 \times 5 + 0.1 \times 1$	$- 3 + \max\{2, 4, \} = 5.6$	
$C(3, 1 2, \{5\}) =$	$0.8 \times 10 + 0.2 \times 2$	$- 4 + \max\{-3, 3, \} = 7.4$	
$C(3, 1 2, \{4, 5\}) =$	$0.9 \times 5 + 0.1 \times 1 + 0.8 \times 10 + 0.2 \times 2$	$- 6 + \max\{-2, 1, \} = 8$	$= f^1(3, 1)$
$C(3, 2) =$		$\max\{0, 5, \} = 5$	$= f^1(3, 2)$
$C(3, 2 1, \{4\}) =$	$0.0 \times 5 + 1.0 \times 1$	$- 1 + \max\{2, 4, \} = 4$	
$C(3, 2 1, \{5\}) =$	$0.0 \times 10 + 1.0 \times 2$	$- 1 + \max\{-3, 3, \} = 4$	
$C(3, 2 1, \{4, 5\}) =$	$0.0 \times 5 + 1.0 \times 1 + 0.0 \times 10 + 1.0 \times 2$	$- 2 + \max\{-2, 1, \} = 2$	
$C(3, 2 2, \{4\}) =$	$0.5 \times 5 + 0.5 \times 1$	$- 3 + \max\{2, 4, \} = 4$	
$C(3, 2 2, \{5\}) =$	$0.4 \times 10 + 0.6 \times 2$	$- 4 + \max\{-3, 3, \} = 4.2$	
$C(3, 2 2, \{4, 5\}) =$	$0.5 \times 5 + 0.5 \times 1 + 0.4 \times 10 + 0.6 \times 2$	$- 6 + \max\{-2, 1, \} = 3.2$	
$C(1, 1) =$		$\max\{-4, , \} = -4$	
$C(1, 1 1, \{2\}) =$	$0.0 \times 2 + 1.0 \times 2$	$- 1 + \max\{-3, , \} = -2$	
$C(1, 1 1, \{3\}) =$	$0.0 \times 8 + 1.0 \times 5$	$- 2 + \max\{-3, , \} = 0$	
$C(1, 1 1, \{2, 3\}) =$	$0.0 \times 2 + 1.0 \times 2 + 0.0 \times 8 + 1.0 \times 5$	$- 3 + \max\{-2, , \} = 2$	
$C(1, 1 2, \{2\}) =$	$0.7 \times 2 + 0.3 \times 2$	$- 2 + \max\{-3, , \} = -3$	
$C(1, 1 2, \{3\}) =$	$0.9 \times 8 + 0.1 \times 5$	$- 4 + \max\{-3, , \} = 0.7$	
$C(1, 1 2, \{2, 3\}) =$	$0.7 \times 2 + 0.3 \times 2 + 0.9 \times 8 + 0.1 \times 5$	$- 5 + \max\{-2, , \} = 2.7$	$= f^0(1, 1)$
$C(1, 2) =$		$\max\{-4, , \} = -4$	
$C(1, 2 1, \{2\}) =$	$0.0 \times 2 + 1.0 \times 2$	$- 1 + \max\{-3, , \} = -2$	
$C(1, 2 1, \{3\}) =$	$0.0 \times 8 + 1.0 \times 5$	$- 2 + \max\{-3, , \} = 0$	
$C(1, 2 1, \{2, 3\}) =$	$0.0 \times 2 + 1.0 \times 2 + 0.0 \times 8 + 1.0 \times 5$	$- 3 + \max\{-2, , \} = 2$	$= f^0(1, 2)$
$C(1, 2 2, \{2\}) =$	$0.5 \times 2 + 0.5 \times 2$	$- 2 + \max\{-3, , \} = -3$	
$C(1, 2 2, \{3\}) =$	$0.6 \times 8 + 0.4 \times 5$	$- 4 + \max\{-3, , \} = -0.2$	
$C(1, 2 2, \{2, 3\}) =$	$0.5 \times 2 + 0.5 \times 2 + 0.6 \times 8 + 0.4 \times 5$	$- 5 + \max\{-2, , \} = 1.8$	

Table 5: Results of the DP algorithm for the numerical example.