



**HAL**  
open science

## **A real time solution to control disassembly processes.**

Luminata Duta, Jean-Michel Henrioud, Ioan Caciula

► **To cite this version:**

Luminata Duta, Jean-Michel Henrioud, Ioan Caciula. A real time solution to control disassembly processes.. 4th IFAC Conference on Management and Control of Production and Logistics, MCPL'2007., Sep 2007, Sibiu, Romania. pp.789-794. hal-00181100

**HAL Id: hal-00181100**

**<https://hal.science/hal-00181100>**

Submitted on 23 Oct 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## A REAL TIME SOLUTION TO CONTROL DISASSEMBLY PROCESSES

L. Duta <sup>(1)</sup>, J. M. Henrioud <sup>(2)</sup>, I. Caciula <sup>(3)</sup>

<sup>(1)</sup> *Computer Science Department  
Valahia State University  
18-24, Unirii Av. Targoviste, ROMANIA  
Fax: +40 (0) 245 21 76 83, E-mail: [duta@valahia.ro](mailto:duta@valahia.ro)*

<sup>(2)</sup> *Automation Laboratory  
Institute of Production Engineering  
25, Alain Savary, Besançon, FRANCE  
E-mail: [henrioud@ens2m.fr](mailto:henrioud@ens2m.fr)*

<sup>(3)</sup> *Control Engineering Student  
Valahia State University  
18-24, Unirii Av. Targoviste, ROMANIA  
E-mail : [viorel2004\\_ro@yahoo.com](mailto:viorel2004_ro@yahoo.com)*

Abstract: Disassembly of manufactured products induces both disassembly costs and revenues from the parts saved by the process. At the planning stage a good trade-off has to be found between the costs of disassembly and the final profit. At the control stage it is important to assure an optimal balance of the line as well as the complete disassembly processing during the rest of the working time. A real time control method based on modeling of disassembly by the precedence graph and on a stochastic algorithm is presented in this article. *Copyright © 2007 IFAC*

Keywords: production control, real time systems, decision making, stochastic programming

### 1. INTRODUCTION

The disassembly process is the main stage in the recycling of manufactured products at the end of their life. The aim of this process is to extract the reusable parts of the product, materials to recycle, as well as the dangerous materials. Products submitted to the disassembly process are out of use so we have to take into account their physical state.

Disassembly is a non-destructive technique: it implies the extraction of the desired components and/or materials. If parts are not reusable after reconditioning, partial or total destructive operations are applied: drilling, cutting, wrenching, and shearing. These techniques are used in view of material or energy recovery.

The aim of the valorization process is to save the value of parts and materials by repairing and recovering operations.

Thus a trade off has to be made between the cost of disassembly operations and the revenue brought by the parts or the material retrieved. This problem is made more complex by the occurrence, during the disassembly process, of uncertainty in the possibility of components separations (Penev and De Ron, 1996).

Deteriorations and deformations of some elements, absence of one or more components, presence of corrosion and rust are perturbations often encountered in the disassembly process. Therefore, some operations cannot be carried out due to the physical degradations of the components and other

operations are not performed if they are not profitable. So, a choice has to be made, between applying an alternative disassembly destructive operation (dismantling), and abandoning the disassembly procedure (Salomonski and Zussman, 1999).

Thus, control of disassembly process involves two essential decision variables: disassembly level and disassembly mode: clean or destructive. A third decision variable may be added: the task assignment to the station. This decision variable (the only one available in assembly) is far more constrained than the two former ones. This makes from the control of disassembly systems a far more complex problem than the control of assembly systems.

As mentioned above, an essential criterion for a disassembly system is the benefit it brings, that is the revenue brought by the retrieved parts and material, decreased by the cost of their retrieval. Another important criterion, especially for manual systems is the line balancing. One way to deal with this multi-criteria problem is to use a multi-criteria method, the problem is that this will require a weighting of the two criteria (revenue and line balance) and there is no satisfying way to choose the weights. In order to avoid this problem, the authors have proposed a two stages method.

At the first stage, which is the Disassembly Line Design, disassembly level is obtained by the optimization of the revenue produced by the line. This is resolved by modeling the set of valid disassembly sequences using a Petri Net and Linear Programming (Addouche *et al*, 2002). At this stage, disassembly costs are supposed to be simply proportional to the disassembly time.

At the second stage, of the Disassembly System Control, decisions are made in order to optimize the line balance and guarantee that the production plan will be realized at the end of the plan horizon (usually the week). Assumption of disassembly costs proportional to disassembly time is no longer relevant. While working, as soon as the line gets unbalanced, time becomes highly valuable at overloaded workstations and without any value at idle ones.

Few authors have addressed the control problem of disassembly lines (few had addressed before the control problem of assembly lines). One first attempt was made by (Kizilkaya and Gupta, 1998) who used the method "just-in-time" and a flexible Kanban system to control the flow of the disassembly process. But his approach is not suited to the automatic disassembly lines. Chevron in his doctoral research (Chevron, 1999) proposed the Colored Petri Nets for modeling the disassembly process and a dedicated computer to perform the on-line control. The lack of his method is that he didn't take into account the application of the destructive operations in the case of the disassembly failure. (Wiendahl *et al*, 1999) proposed a special architecture of the

disassembly system control. This one includes a decision module which gives the possible tasks assignments on the line. They proposed the linear programming to program this module but they never gave an example. Recently (Kopacek and Kopacek, 2005) describes a control system for mobile phones disassembly cells. Unfortunately, this system can not be functional in the case of destructive operations. In fact, even the author specified that the system is suited for small series of products that haven't suffered great structure modifications in their functional life.

## 2. THE OPTIMISATION PROBLEM

### 2.1. Preliminary considerations

The problem considered in this paper is the real time control of a Disassembly Line whose design has been optimized so that the Line produces, on an average production, the best trade-off between disassembly cost and revenue brought by the retrieved parts and material. The result is:

- the disassembly level,
- the choice of disassembly operations : destructive or not destructive,
- the design of the line and the assignment of operations to workstations.

At the On Line Control Level, it is very important to assure the Line Balance, more specially, but not only, if the Line is mostly manual. This is far more critical in disassembly than in assembly because disassembly times present an important dispersion.

What follows will be based on the following assumptions.

- the Disassembly Line is multi-product and all the operations to perform and their precedence relations may be represented by a single precedence graph, where nodes represent generic operations. Operative times take the zero value for operations missing on some products (when optional components are involved).
- The sequencing of the product on the line has been fixed and cannot be changed.
- The objective of the control is to make sure that the products that are to be disassembled in the current period will be as completely treated as possible during the remaining time and that the line keeps a good balance in the dynamic sense.

### 2.2. The objective function

#### Notations:

$n$  - the number of workstations

$m_i$  - the total number of operations on the workstation  $W_i$

$t_{cy}$  - the cycle time

$t_j$  - the operational time for the task  $j$

Considering the *balancing function* presented in (Duta *et al*, 2005) which gives the difference between the operational times and the cycle time the next formula gives one objective function to minimise:

$$f(t) = \sum_{i=1}^n \left( t_{cy} - \sum_{j \in W_i}^{m_i} t_j \right)^2 \quad (1)$$

In a *static approach* of the disassembly process the cycle time is defined as the ratio between the duration of the planning period  $H$  and the number  $S$  of products to be disassembled (Duta, 2006; Lambert and Gupta, 2006):

$$t_{cy} = \frac{H}{S} \quad (2)$$

It is supposed that the products to disassemble are similar (with the same structure), so the operational times are equal from one product to another. Thus, the objective function has the same form and the operational times are multiplied by  $S$ .

Equation (1) becomes

$$f(t) = S \cdot \sum_{i=1}^n \left( t_{cy} - \sum_j^{m_i} t_j \right)^2 = \sum_{i=1}^n \left( H - S \cdot \sum_j^{m_i} t_j \right)^2 \quad (3)$$

A problem occurs in the case of the real time disassembly: the operation can be fulfilled or not or there are destructive operations to accomplish. The aim is find the form of the objective function in real time.

Notations:

- $n$  is the number of workstations,
- $Tr$  is the time remaining for the disassembly process
- $Tr_i$  is the time left for workstation  $W_i$  to complete its remaining workload

In a *dynamic approach*, the objective defined above may be expressed by formula (4) which is the objective function to be minimized:

$$F = \sum_{i=1}^n [Tr - Tr_i]^2 \quad (4)$$

$Tr_i$  can be expressed as a sum of the operational times performed by workstation  $W_i$ .

### 2.3. Real time coefficients

To compute  $Tr_i$ , we have introduced *three binary variables*:

$\phi_{ij}$  **the assignment coefficient** that defines assignment of tasks to stations for different products.

$\phi_{ij} = 1$  when the operation  $O_j$  can be assigned to workstation  $W_i$

$\phi_{ij} = 0$  otherwise

$\psi_{ij}$  **the state coefficient** that defines which operation has already been performed and which one is still to be done.

$\psi_{ij} = 1$  if operation  $O_j$  has not still been performed on product  $P_i$ . ( $P_i$  being the  $i^{\text{th}}$  product of the sequence)

$\psi_{ij} = 0$  otherwise.

$\theta_{ij}$  **the control coefficient** that defines the modality of performance for the operations.

$\theta_{ij} = 1$  when the operation  $O_j$  is to be performed without damaging the product  $i$

$\theta_{ij} = 0$  when the operation  $O_j$  has to be performed in a destructive way on product  $i$

Then,  $Tr_i$  is given by the next formula (Duta, 2006):

$$Tr_i = \sum_{j=1}^m \phi_{ij} \psi_{ij} \cdot [\theta_{ij} \cdot t_{ij} + (1 - \theta_{ij}) \cdot t'_{ij}] \quad (5)$$

Where

$m$  is the total number of operations,

$P$  is the number of products to be disassembled

$t_{ij}$  is the average duration attributed to the operation  $O_j$  on a product  $P_i$  when it is performed in a non destructive way.

$t'_{ij}$  is the average duration attributed to operation  $O_j$  on a product  $P_i$  when it is performed in a destructive way.

Remarks:

1. An operation  $O_j$  is still to be performed on some products yet to be disassembled, if  $\exists \square \square i$  so as  $\psi_{ij} = 1$ . Thus,  $\psi_{ij}$  has two functions: first it defines the operations still to be performed, hence the disassembly level, and second it defines the assignment of tasks to workstations.

2. Table  $[\varphi_{ij}]$  is subject to two kinds of constraints:

- Assignment constraints that define for each operation the workstations that are able to perform it. Mostly workstations are dedicated to specific tasks, due to the tooling, and there are few possibilities of reassignment.

- Precedence constraints that are represented by a precedence graph

A possible combination of the three binary coefficients is given in the Table 1.

Table 1. Binary coefficients variants

No	$\varphi_{ij}$	$\psi_{ij}$	$\theta_{ij}$	Meaning
1	1	0	0	Operation j made in a destructive way on the workstation i
2	1	0	1	Operation j made in a non-destructive way on the workstation i
3	1	1	0	Operation j has to be made in a destructive way on the workstation i
4	1	1	1	Operation j has to be made in a non-destructive way on the workstation i

#### 2.4. Defining the solution space search

A solution for the optimisation problem stated above is given by tables  $[\varphi_{ij}]$ ,  $[\psi_{ij}]$  and  $[\theta_{ij}]$  and it defines a valid partition of the precedence graph where no operation in a workstation with rank i can precede an other operation on a workstation with rank less than i. For any suppressed operation  $O_j$  all operations,  $O_h$  such that  $O_i$  precedes  $O_h$ , are also suppressed. The optimal tasks assignment as optimal solution must be searched in the neighbourhood of the current solution.

**Definition 1:** Two solutions are neighbours if one of them can be transformed into the other one, by one of the four transformations T1 - T4, given below.

**Definition 2:** The upper (or lower) border of a workstation  $W_i$  is the set of all tasks that have no predecessor (or successor) in  $W_i$ .

To obtain a neighbourhood four transformations can be possible:

T1. One task belonging to the upper border of a workstation is moved to the lower border of the next workstation

T2. Suppression of a task. This transformation applies only to tasks that have no other successor than the last task of the precedence graph (the unloading of what is left of the product to disassemble).

T3. The introduction of a task that has been suppressed before. This transformation applies only to tasks that have no predecessor in the precedence graph (the loading of the product to disassemble on the disassembly line).

T4. Transformation of a task that may be performed either in the destructive mode or in the non destructive one: if it was planned in the destructive mode, it becomes non destructive and reverse.

### 3. THE ALGORITHM

The algorithm used for the optimisation problem formulated in the previous section is a Kangaroo algorithm that belongs to the stochastic algorithms class. Kangaroo algorithm has already been used in the closed problem of Assembly Line by V. Minzu (Minzu and Henrioud, 1998). The first step in the building of a stochastic algorithm was to define the solution space and a neighbourhood in this space. This algorithm searches the optimal solution in the neighbourhood of the current solution. The steps of the Kangaroo algorithm are described as it follows:

**Step1** Let fix a value  $A = a \cdot \left[ \frac{N}{M} \right]$

where  $a \in \{1, 2\}$ , N=the number of tasks and M=the number of workstations;

**Step 2** Let choose as a valid solution the initial partition named  $u$

**Step 3**

A better solution (partition)  $u^*$  is searching so as to minimize the objective function  $f_{obj}(u)$

Let  $c \leftarrow 1$ ;  $u^* \leftarrow u$ ;

**Step 4**

execute

if  $c < A$  then call  $descent(u, u^*, c)$

else call  $jump(u, u^*, c)$ ;

until (an end criteria is fulfilled);

**STOP**

The two procedures  $descent$  and  $jump$  are described as it follows:

procedure  $descent(u, u^*, c)$

Begin

Generate a solution  $v$  in the neighbourhood of  $u$

$c \leftarrow c+1$ ;

if  $f(v) \leq f(u)$  then

begin if  $f(v) < f(u)$  then

begin

$c \leftarrow 0$ ; // a better value of the function has been obtained;

if  $f(v) < f(u^*)$  then  $u^* \leftarrow v$ ; //a lowest value of the objective function has been obtained;

end;

$u \leftarrow v$ ; //the old partition is replaced with the new one

```

end;End.
procedure jump(u,u*,c)
Begin Generate v in the neighbourhood of u
  c ← c+1;
  if  $f(v) \neq f(u)$  then
  begin
    if  $f(v) < f(u^*)$  then  $u^* \leftarrow v$ ;
    c ← 0;
  end;u ← v;
End.

```

#### 4. APPLICATION

The method was simulated on a case study: disassembly of a cell phone which precedence graph is taken from (Lambert and Gupta, 2006).

There are 25 disassembly operations and 30 disassembly sequences.

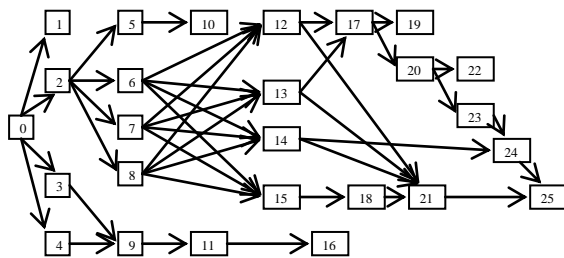


Fig. 1. Precedence graph of the cell phone

The operational times  $t_j$  are given in (Gupta and McGovern, 2004).

Table 2. Cell phone operational times

Op	1	2	3	4	5	6	7	8	
$t_j$	3	2	3	10	10	15	15	15	
Op	9	10	11	12	13	14	15	16	
$t_j$	15	2	2	2	2	2	2	2	
Op	17	18	19	20	21	22	23	24	
$t_j$	2	3	18	5	1	5	15	2	
Op	25								
$t_j$	2								

We considered 6 workstations that can accomplish destructive or non-destructive disassembly operations.

Assumptions:

- the supply of product is continuous
- a single type of product is disassembled
- a task can not be divided between two stations
- disassembly tasks are assigned so as not to violate the precedence relationships among them
- complete disassembly is performed
- destructive operations are taken into account

Input data:

$H$ : 8 h,  $S$ : 1000,  $t_{cy}$ : 30 s, operational times  $t_j$

The initial assignment is given in Table 3

Table 3. Initial tasks assignment

Workstation	Assigned tasks
1	1,2,3,4, 5
2	6,7
3	8,9
4	10,11, 12, 13, 14, 15, 16, 17, 18, 20
5	19, 23
6	21, 22, 24, 25

The Kangaroo algorithm defines a neighbourhood of the solution in case of a perturbation occurrence and permits the minimisation of the balancing function given by the equation (4). Perturbations can be provided by the impossibility to perform a disassembly operation or by the change of its type – destructive or not. Moving an operation to one station to another, a new neighbourhood is obtained and the search of a new solution is performed in real time.

The new assignment solution is provided in real time taking into account the precedence relationships between disassembly operations. The program was implemented in the C++ language. On a 64 AMD Athlon processor the computational time is 10.12 ms.

#### 5. CONCLUSIONS

A method to control the disassembly processes is presented in this paper. Disassembly processes are submitted to perturbations and uncertainties derived from the used state of the product. Once a perturbation occurs, a fast computation has to be made so as to determine the optimal disassembly sequence and the optimal assignment of the tasks to workstations.

A stochastic algorithm is proposed in order to manage the optimisation problem in real time control. The optimised function is dependent of three discrete variables.

The algorithm does not optimize the balance of the disassembly line, but gives a solution that improves this balance at each computation moment.

Applying a stochastic algorithm in discrete optimisation the quantity of data decreases and the calculus speed rises. In the disassembly process a local and quickly solution for the optimal disassembly sequence is preferred to the complex and slower algorithms.

Future work will be centred on the deduction of the general form of the function (4) that is the balancing function for a product family. In this case the operational times are dependent of the product type. Feasible methods have to be found to control the flow on a family product disassembly line.

#### REFERENCES

- Addouche S., Perrard C. and Henrioud J.M. (2002), Linear Programming Model To Find The Optimal Disassembly Sequence, *The 3rd CIRP International Seminar on Intelligent Computation in Manufacturing Engineering* , (ICME2002), Italy.
- Chevron D., (1999), *Contribution à l'étude de la supervision d'une cellule de démontage de produits techniques en fin de vie*, Thesis, Institut National Polytechnique de Grenoble
- Duta, L., F.G. Filip and J.M. Henrioud (2003). A method for dealing with multi-objective optimization problem of disassembly processes, *Proceedings of The IEEE International Symposium of Assembly and Task Planning – ISATP03*, pg. 163-168, July, 2003, Besançon, France.
- Duta, L., Filip, Gh. F., Henrioud J. M. (2005) Applying equal piles approach to disassembly line balancing problem, *Proceedings of the 16<sup>th</sup> IFAC World Congress*, Prague, July 2005, on CD
- Duta, L., (2006). *Contribution à l'étude de la conduite des systèmes de désassemblage*, Thesis, Franche-Comté University, Besançon, France, september 2006
- Gupta, S., McGovern S.M., (2004), Disassembly Sequencing problem: a case study of a cell phone, *Proceedings of SPIE, Bellingham, WA*, **vol. 5583**, pp 43-52
- Kizilkaya E., Gupta S., (1998), Material flow control and scheduling in a disassembly environment, *Computers Industrial Engineering*, **vol. 35**, no. 1, pp. 93-96
- Kopacek P, Kopacek B, (2005), Intelligent, flexible disassembly, *International Journal of Advanced Manufacturing Technology*, **vol 1**, pp. 424-531
- Lambert, A. J. D., Gupta S., (2006) Optimum disassembly sequencing, *Proceedings of SPIE, USA*, **vol. 6385**, pp 71-78
- Minzu V., J.M. Henrioud (1998) Stochastic algorithm for tasks assignments in single or mixed-model assembly lines, *APII-JESA*, **vol. 32**, n°7-8, 1998, pp. 831-851.
- Penev K. D. and A. J. de Ron (1996), Determination of a disassembly strategy, *International Journal of Production Research*, **vol. 34**, pp. 495-506, 1996.
- Salomonski, N., E. Zussman (1999), On-line Predictive Model for Disassembly Process Planning Adaptation. *Robotics and Computer Integrated Manufacturing*, **vol. 15**, pp. 211-220
- Wiendhal H.P., Seliger G., Perlewitz H., Brukner S., (1999), A general approach to disassembly