Complications in Disassembly Line Balancing

Askiner Güngör[†], Surendra M. Gupta[‡*], Kishore Pochampally[‡] and Sagar V. Kamarthi[‡]

[†]Department of Industrial Engineering
Pamukkale University
20017 Denizli, Turkey

[‡]Laboratory for Responsible Manufacturing
334 SN, Department of MIME
Northeastern University
360 Huntington Avenue
Boston, MA 02115 U.S.A.

ABSTRACT

Disassembly line is, perhaps, the most suitable way for the disassembly of large products or small products in large quantities. In this paper, we address the disassembly line balancing problem (DLBP) and the challenges that come with it. The objective of balancing the disassembly line is to utilize the disassembly line in an optimized fashion while meeting the demand for the parts retrieved from the returned products. Although, the traditional line balancing problem for assembly has been studied for a long time, so far, no one has formally talked about the DLBP. In this work, our primary objective is to address the DLBP related issues. However, we also present a heuristic to demonstrate how several important factors in disassembly can be incorporated into the solution process of a DLBP. An example is considered to illustrate the use of the heuristic.

Keywords: Environment; Product Recovery; Disassembly; Line Balancing; Heuristic

1. INTRODUCTION

The current state of the environment is being seriously threatened by the extraordinary growth in the advancement of technology. Take, for example, the case of computers. According to a forecast, by the year 2005, there will be almost 50 million computers becoming obsolete every year in the U.S. alone. The rest of the developed countries will experience a similar phenomenon. The life cycles of other products will most likely have a similar fate. Governmental regulations and customer perspective on environmental issues have further fueled this trend.[4] All this has an effect on the waste management infrastructure. Many researchers and industry executives have started to realize the economic opportunities that lie ahead in the area of end-of-life (EOL) processing of products. Among the desirable alternatives for EOL processing of products are remanufacturing, reusing and recycling. Although disposal and incineration are also possible EOL alternatives, they are less desirable and should be kept to a minimum. In order to remanufacture, reuse or recycle, often the product has to be disassembled first. Disassembly has proven its role in material and product recovery by allowing selective separation of desired parts and materials.[6] However, disassembly, though crucial, is an expensive process. Therefore, performing disassembly in a cost effective manner is important.

Many researchers have already focused on minimizing the resources invested in the disassembly process. Some, for example, have focused on the disassembly leveling problem, which targets disassembly to a level to which the product of interest is disassembled such that the profitability and environmental features of the product recovery (PR) process are kept at a desired level.[2, 12, 14] Another important issue in disassembly is the generation of efficient disassembly sequence plans (DSP). A DSP is a sequence of disassembly tasks that begins with a product to be disassembled and terminates in a state

---

*Correspondence: e-mail: gupta@neu.edu; URL: http://www.coe.neu.edu/~smgupta/
Phone: (617)-373-4846; Fax: (617)-373-2921

where all of the *parts of interest* are disconnected (thus, it could be either partial or complete disassembly). In the disassembly literature, although the disassembly sequence planning has found a large following[1, 7, 8, 9, 10, 13, 18, 19], only a handful of researchers have emphasized the shortcomings of the existing disassembly systems and suggested any improvements. Current disassembly systems are generally manual and labor intensive. Therefore, designing and improving disassembly systems which optimize the use of resources (labor, money and time) are important and worth investigating.

In this paper, we address the disassembly line balancing problem (DLBP) and the challenges that come with it. The objective of balancing the disassembly line is to utilize the disassembly line in an optimized fashion while meeting the demand for the parts retrieved from the returned products. Although, the traditional line balancing problem for assembly (ALBP) has been around for a long time[15], so far, no one has formally talked about the DLBP. In this work, our primary objective is to address the DLBP related issues. However, we also present a heuristic to demonstrate how several important factors in disassembly can be incorporated into the solution process of a DLBP. The heuristic is based on a priority function, which is instrumental in identifying the "best" task to assign to a particular workstation.

2. THE DLBP AND RELATED COMPLICATIONS

The disassembly of returned products can be performed at a single workstation, in a disassembly cell or on a disassembly line.[18, 19] Even though a single workstation or the disassembly cell provides the most flexible environment for sorting parts according to their quantity and quality, the disassembly line provides the highest productivity rate. The disassembly line setting is most suitable for disassembly of large products or small products in large quantities. Furthermore, the disassembly line is the best choice for automated disassembly process, a feature that will be essential in the future disassembly systems.[16, 17] It is, therefore, important that the disassembly line be designed and balanced so that it works as efficiently as possible.

In disassembly, unlike assembly, there are serious inventory problems, much more complicated flow process, a high degree of uncertainty in the structure and the quality of the products, and uncertainty factors associated with the reliability of the workstations. Let us take a closer look at the various disassembly line balancing complications.

2.1. Product Complications

Changing characteristics of products complicate the operations on a disassembly line. Balancing the disassembly line used in such cases can be very complex. Such a line may be balanced for a group of products yet may become unbalanced when a new type of product is received.

2.2. Disassembly Line Complications

Various line configurations may be possible. They are proposed to cope with the irregularities and product variability in the disassembly system. One important consideration is the line speed. It can be dynamically modified to minimize the effects of varying demands for subassemblies and/or parts on the disassembly line.

2.3. Part Complications

Quality of Incoming Products: There is a high level of uncertainty in the quality of the products received and their constituent parts. They may be either physically defective or functionally defective or both.

Quantity of Parts in Incoming Products: Due to upgrading (or downgrading) of the product during its use, the actual number of parts in it may be more (or less) than expected when the product is received.

2.4. Operational Complications

Variability of Disassembly Task Times: The disassembly task times may vary depending on several factors that are related to the condition of the product and the state of the disassembly workstation (or worker). Dynamic learning is possible, which allows systematic reduction in disassembly times.

Early Leaving Work-pieces (EWP): If one or more (not all) tasks of a work-piece, which have been assigned to the current workstation, cannot be completed due to some defect (that might be related to one or more of the tasks), the work-piece might leave the workstation early. We term this phenomenon as the *early-leaving work-piece (EWP).* Due to EWP, the workstation experiences an unscheduled idle time for the duration of the tasks that causes the work-piece to leave early.

Self-Skipping Work-pieces (SSWP): If all tasks of a work-piece, which have been assigned to the current workstation, are disabled due to some defect of their own and/or precedence relationships, the work-piece leaves the workstation early without being worked on. We term this phenomenon as *self-skipping work-piece (SSWP).*

Skipping Work-pieces (SWP): At workstation m, if one or more defective tasks of a work-piece directly or indirectly precede **all** the tasks of workstation m+1 (i.e., the workstation immediately succeeding workstation m), the work-piece "skips" workstation m+1 and moves on to workstation m+2. We term this phenomenon as *skipping work-piece (SWP).* In addition to unscheduled idle time, both SSWP and SWP experience added complexities in material handling and the status of the downstream workstation.

Disappearing Work-pieces (DWP): If a defective task disables the completion of all the remaining tasks on a work-piece, the work-piece may simply be taken off the disassembly line before it reaches any downstream workstation. In another words, the work-piece "disappears"! Therefore, we term this phenomenon as the *disappearing work-piece (DWP)*. DWP may result in starvation of subsequent workstations leading to a higher overall idle time.

Revisiting Work-pieces (RWP): Work-piece currently at workstation w, may *revisit* a preceding workstation (w-a), where (w-a) ≥ 1 and a ≥ 1 and integer, to perform task f if the completion of current task i enables one to work on task f which was originally assigned to workstation (w-a), and was, however, disabled due to the failure of another preceding task. We term this *revisiting work-pieces* (*RWP).* An RWP results in overloading one of the previous workstations.

Exploding Work-pieces (EWP): A work-piece may split into two or more work-pieces (subassemblies) as it moves on the disassembly line because of the disassembly of certain parts that hold the work-piece together. Each of these subassemblies acts as an individual work-piece on the disassembly line. We term this phenomenon as the *exploding work-pieces (EWP).* The EWP complicates the flow mechanism of the disassembly line.

2.5. Demand Complications

In disassembly, the following demand scenarios are possible: Demand for one part only (single part disassembly **-** a special case of partial disassembly); demand for multiple parts (partial disassembly); and demand for all parts (complete disassembly). Possible physical and functional defects in the demanded parts or the parts preceding the demanded parts may complicate the situation further.

2.6. Assignment Complications

Certain tasks must be grouped and assigned to a specific workstation for reasons like requirement of similar operating conditions for them and availability of special machining and tooling at certain workstations.

2.7. Other Complications

There are additional uncertainty factors associated with the reliability of the disassembly workstations. For example, hazardous parts may require special handling, which can also influence the utilization of the workstations. Some of the assembly line balancing factors, which are presented by Ghosh and Gagnon[3] in their comprehensive literature survey, can also be important in the disassembly line balancing case.

3. OBJECTIVE AND CONSTRAINTS OF THE DLBP

The objective of the DLBP is to utilize the resources of the disassembly line as efficiently as possible while meeting the demand. Efficient utilization of resources consists of finding the minimum number of disassembly workstations required, optimally assigning the disassembly tasks to the workstations, and improving the layout and material handling features of the disassembly line.

Some of the precedence relationships that need to be considered in the disassembly case include AND, OR and complex AND/OR.[11] In order to understand these terms, let p_i represent part i in a product to be disassembled. An *AND relationship* exists between p_1 and p_2 in relation to p_3, if both p_1 and p_2 must be removed prior to p_3. An *OR relationship* exists between p_1 and p_2 in relation to p_3, if either p_1 or p_2 must be removed prior to p_3. A *complex AND/OR relationship* exists between p_1, p_2, and p_3, in relation to p_4, if p_1 along with either p_2 or p_3 must be removed prior to p_4.

In the DLBP, to represent the precedence relationship, we utilize a disassembly precedence matrix (DPM) which strictly represents the geometrically based relationships among the parts. The DPM consists of the binary elements plus another entity (d) to represent OR and complex AND/OR relationships among parts; where d denotes the disassembly movement in x-y-z directions; $d \in D = \{x, -x, y, -y, z, -z\}$. We mathematically represent the DPM by $R = [r_{ij}]$, $i, j = 1, \ldots, N$ where N is the number of parts in the product and

$$r_{ij} = \begin{Bmatrix} 1, & \text{part } i \text{ AND precedes part } j \\ d, & \text{part } i \text{ OR precedes part } j \\ 0, & \text{otherwise} \end{Bmatrix} \tag{1}$$

In order to generate the DPM, we use the algorithm developed by Gungor and Gupta.[5]

4. PROBLEM DESCRIPTION

In order to demonstrate the intricacies of a disassembly line balancing procedure, we present a simple DLBP. The problem is defined as follows: A paced disassembly line is utilized to disassemble one type of product into its constituent parts and subassemblies. We assume that there is an infinite supply of products. The configuration of each product received is identical which means that the exact quantity of the parts in each product received is known. For simplicity, the disassembly times are assumed to be deterministic and known. Every part in the product has an associated demand i.e., complete disassembly is targeted. The demand parameters are deterministic and known. The parts disassembled are accepted by the demand source in their current conditions.

5. ANALYSIS OF THE DLBP

The cycle time of a disassembly line can be written as follows (the notation not defined in the body of the paper are listed in the appendix):

$$c = \frac{\text{The duration of the planning period}}{\text{The number of products that need to be disassembled to meet the demand}} = \frac{L}{dv_{\max}}, \tag{2}$$

where dv_{max} is the demand level of a part with the highest demand, i.e.,

$$dv_{\max} = \max_{i=1,\cdots,N} \left\{ \frac{\text{Demand of part } i}{\text{Quantity of part } i \text{ in the product}} \right\} = \max_{i=1,\cdots,N} \left\{ \frac{dv_i}{q_i} \right\}. \tag{3}$$

c must satisfy the following condition:

$$t_i \le c, i = 1, \cdots, N. \tag{4}$$

If the condition given in (4) is not satisfied, the assignment of task t_{max}, where $t_{max} > c$, is not possible since tasks are assumed to be indivisible. Assume that disassembly of parts must be carried out even if their delivery is postponed. The cost associated with the backordered parts is not considered. Therefore, we can modify the cycle time such that it allows tasks to be assigned to the workstations. Increasing c is equivalent to extending the planning period. A simple modification is to set the value of c to t_{max}; i.e., $c = t_{max}$.

Once the cycle time is known, the tasks need to be assigned to the workstations in an optimized fashion. However, the number of workstations required is unknown. Minimizing the number of disassembly workstations is one of the objectives of the DLBP. The theoretical minimum number of workstations can simply be found as follows:

$$M_{\min} = \left\lceil \frac{T}{c} \right\rceil = \left\lceil \frac{\sum_{i=1}^{N} t_i}{c} \right\rceil \tag{5}$$

where $\lceil x \rceil$ is the smallest integer $\geq x$. Of course, the maximum number of workstations can be given as follows:

$$M_{\max} = N \tag{6}$$

The lower bound represents the optimistic number of workstations required for the disassembly line whereas the upper bound is the pessimistic number of the disassembly workstations. In most problems, the actual number of workstations required is less than M_{max}. The closer the actual required number of workstations M is to M_{min}, the better the disassembly line is balanced. The idle time of workstation k is the difference between the cycle time and the workstation time of k written as follows:

$$I_k = c - S_k \tag{7}$$

where:

$$S_k = \sum t_j, j \in A_k \tag{8}$$

Then, we can write the idle time of the disassembly line as follows:

$$I = \sum_{k=1}^{M} I_k \tag{9}$$

or equivalently:

$$I = (M \times c) - T \tag{10}$$

The idle time is an important measure for the disassembly line efficiency. However, in a disassembly system, there are many other important factors, which should be integrated into the disassembly line balancing procedure. For example, disassembly of highly demanded parts may be assigned to the earliest possible workstation (of course, without violating the precedence relationships), the hazardous parts in the product should be removed as early as possible to reduce the risk of contamination, the parts that are easily accessible and the parts that precede many other parts should be disassembled as early as possible to guarantee the quality of the recovered parts etc. These considerations can be captured into a priority function that can be used to select a task among the candidate tasks to be assigned to the current disassembly workstation. A task i is said to be a *candidate task*, if and only if, it satisfies the following three criteria:

1. Task i must not have already been assigned to any earlier workstation, i.e.,

$$i \notin \bigcup A_m, m = 1, \cdots, k-1 \tag{11}$$

2. Task i must not have any incomplete predecessor, i.e.,

$$\begin{cases} r_{ji} = 0, j = 1, \cdots, N; \text{or} \\ \exists r_{ji} = 1, j \in A_m; m = 1, \cdots, k-1; \text{and/or} \\ \exists r_{ji} = d, \ni j \in OG_{i,d} \subset \bigcup A_m; m = 1, \cdots, k-1. \end{cases} \tag{12}$$

3. The workstation time of k plus the operation time of task i must be less than or equal to the cycle time; i.e.,

$$S_k + t_i \leq c \tag{13}$$

Once the candidate tasks are identified, we can use a priority function that determines which one of the candidate tasks will be assigned to the current workstation, k. As previously noted we can incorporate many factors into the priority function. In this paper, we limit ourselves to the following considerations:

1. *Idle times of workstations*: Evaluation of the idle times of workstations is considered in order to achieve the minimum number of workstations required on the line.
2. *Disassembly of highly demanded parts*: Disassembly of highly demanded parts should take place at the earliest workstation(s) possible.
3. *Disassembly of easily accessible parts, which precede the largest number of parts*: Parts that are easily accessible and precede many other parts should be removed as early as possible.
4. *Disassembly of parts with hazardous contents*: Parts with hazardous material contents should be removed from the work-piece as early as possible.
5. *Minimizing disassembly direction changes*: The objective is to reduce the number of times the work-piece is re-oriented on the disassembly line.

The priority function can be written as follows:

$$F_i = f(I_{i,k}) + f(dv_i) + f(sr_i) + f(h_i) + f(md_i), i \in CA_k. \tag{14}$$

where:

$f(I_{i,k})$	=	the priority value with respect to the idle time of the workstations. It is equal to the rank of the current $I_{i,k}$ in the ascending ordered list of $I_{i,k}$.
$f(dv_i)$	=	the priority value with respect to disassembly of highly demanded parts. It is equal to the rank of the current dv_i in the descending ordered list of dv_i.
$f(sr_i)$	=	the priority value with respect to parts that are easily accessible and precede many other parts (e.g., part i where $R_{\cdot j} = \mathbf{0}$ and $R_{i\cdot}$ contains the most number of nonzero entities). It is equal to the rank of the current sr_i in the descending ordered list of sr_i.
$f(h_i)$	=	the priority value with respect to parts with hazardous materials. It is equal to the rank of the current h_i in the descending ordered list of H.
$f(md_i)$	=	the priority value with respect to disassembly movement direction changes. It is equal to the rank of the current md_i in the ascending ordered list of md_i.

The following are used for the calculation of F_i:

$$I_{i,k} = c - (S_k + t_i), i \in CA_k \tag{15}$$

$$h_i = \begin{Bmatrix} 1, \text{if task } i \text{ is associated with a part with hazardous content} \\ 0, \text{otherwise} \end{Bmatrix}, i = 1, \cdots, N \tag{16}$$

$$md_i = \begin{Bmatrix} 1, \text{ if task } i \text{ is associated with a part that requires a disassembly} \\ \text{direction change with respect to the part preceeding it} \\ 0, \text{ otherwise} \end{Bmatrix}, i = 1, \cdots, N \tag{17}$$

To find sr_i, we scan the row of each candidate task in R to find the nonzero elements. The tasks corresponding to these elements must precede i. After discounting the tasks that have already been assigned to previous workstations, we are able to generate the value of sr_i.

Tie Breaking Rule for $f(x)$: If $x_v = x_w$, where $v \neq w$ and $v, w \in CA_k$, then the ranks of x_v and x_w are assumed to be the same; i.e., $f(x_v) = f(x_w)$. For example, see Table 1.

Table 1: Example of the tie-breaking rule for $f(x)$

	CA_k			
	1	2	3	4
x	14	14	16	13
$f(x)$	**2**	**2**	3	1

Tie Breaking Rule for F_i: If $F_v = F_w = F_{min}$, where $v \neq w$ and $v, w \in CA_k$, then follow the tie breaking rule given below. The top rule has the highest priority; the bottom one has the lowest priority. If there is a tie in the highest priority rule, then break the tie by going to the next highest priority.

1. Candidate with the smallest $f(I_{j,k})$ first (Highest priority)
2. Candidate with the smallest $f(d_i)$ first
3. Candidate with the smallest $f(sr_i)$ first
4. Candidate with the smallest $f(h_i)$ first
5. Candidate with the smallest $f(md_i)$ first (Lowest Priority)

6. A HEURISTIC TO SOLVE THE DLBP

The heuristic is given below in the pseudo code format:

Steps	**Comments**
H→ (R, c, Knowledge base KB){	
k = 1;	/* Create the first workstation */
repeat {	
Determine the candidate tasks, CA_k;	
If ($\|CA_k\| = 0$)	/* There is no candidate task */
$k = k + 1$;	/* Create another workstation */
If ($\|CA_k\| \neq 0$) {	/* There are candidate tasks */
Determine F_i where $i \in CA_k$	/* Calculate the priority function value of each candidate */
$A_k = A_k \cup j$; where $F_j = F_{min}$	/* Assign task j such that j has the minimum priority function value (F_{min}) */
}	
until (all tasks are assigned to workstations)	
$M = k$;	/* the number of required disassembly workstation is k */
return (M, A_m);	
}	/ *end of the algorithm */

The complexity of the heuristic is $O(N^2 \log N)$.

7. AN APPLICATION OF THE HEURISTIC

In this section, we present an application of the heuristic. Consider the disassembly of a simple personal computer (PC). Tasks associated with the disassembly of the PC are presented in Table 2. The matrix R of the PC is given in Figure 1. In order to fulfill the demand levels given in Table 2, the number of products that need to be disassembled is calculated using (3). In Table 2, RAM has the highest part level demand (750 units). However, since disassembly of each PC yields **two**

RAM modules, i.e., r_{am} = 2, the actual requirement for the number of products to be disassembled in order to meet the demand for RAM is 375. Thus, the demand for MB defines the value of dv_{max}, which is 720. Let the planning horizon be an 8-hour shift, L = 8 x 60 x 60 = 28800 seconds. Then, the cycle time is found using (2), i.e., c = 28800/720 = 40 seconds. The condition given by (4) is satisfied which means that the tasks can be assigned to workstations without violating the indivisibility rule.

$$
\begin{array}{cc} & \begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{array} \\
R = \begin{array}{c} n \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{array} &
\begin{bmatrix}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & x & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & -x & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
\end{array}
$$

Figure 1. DPM of the PC example

Table 2: Knowledge base (KB) of the PC example

Task No. (*n*)	**Definition**	**t_i (sec.)**	**dv_i (per day)**	**Hazardous content**	**d^***
1	Removal of the top cover of the PC (TC)	14	360	No	-x
2	Removal of the floppy drive (FD)	10	500	No	x
3	Removal of the hard drive (HD)	12	620	No	-x
4	Removal of the back plane (BP)	18	480	No	x, -x, y, or -y
5	Removal of PCI cards (PCI)	23	540	No	y
6	Removal of **two** RAM modules (RAM)	16	750	No	z
7	Removal of the power unit (PU)	20	295	Yes	-x, x, or y
8	Removal of the motherboard (MB)	36	720	No	z

* Identified during the analysis of the product to generate *R*.

We can now execute the steps of heuristic since all the necessary inputs are known, i.e., *R*, *c*, and the KB. Table 3 presents the step by step execution of the heuristic for the PC example.

The number of stations, *M*, found for the PC example is 4. The tasks have been assigned to stations as follows: A_1 = {1, 3, 2}; A_2 = {5, 6}; A_3 = {8}; and A_4 = {7, 4}. The idle times of the stations are: I_1 = 4; I_2 = 1; I_3 = 4; I_4 = 2 seconds per product respectively. The overall idle time of the disassembly line, *I* = 11 seconds. If each task were assigned to one station, then *M* would be equal to 8 and the overall idle time would be 171 seconds. In addition to improvement of the utilization of the line, the above task assignment has a positive effect based on the priority considerations as discussed earlier.

8. SUMMARY AND CONCLUSIONS

Designing efficient disassembly systems is important to optimize the product recovery process. We discussed the complications that are likely to arise on a disassembly line. Later, we proposed a simple heuristic to solve a DLBP under some assumptions. In the heuristic, we demonstrated how several important factors in a disassembly line environment could be incorporated into the DLBP solution procedure. However, the heuristic is not designed to minimize the number of stations required. Future research can concentrate on this aspect. Further research can also be done when there is a limited supply of products, exact quantity of parts in each product is unknown, disassembly times are not deterministic and the parts disassembled are accepted by the demand source depending upon the type of defects in the parts.

Table 3: Step by step application of the heuristic

Iteration	1	2			3	4		5	6	7	8
W. Station (k)	1	1			1	2		2	3	4	4
$i \in CA_k$	1	2	3	5	2	5	6	6	8	7	4
t_i	14	10	12	23	10	23	16	16	36	20	18
$I_{i, k}$	26	16	14	3	4	17	24	1	4	20	2
$f(I_{i, k})$	1	3	2	1	1	1	2	1	1	1	1
dv_i	360	500	620	540	500	540	750	750	720	295	480
$f(dv_i)$	1	3	1	2	1	2	1	1	1	1	1
sr_i	7	2	2	2	0	2	1	1	1	1	0
$f(sr_i)$	1	1	1	1	1	1	2	1	1	1	1
h_i	0	0	0	0	0	0	0	0	0	1	0
$f(h_i)$	1	1	1	1	1	1	1	1	1	1	1
md_i	0	1	0	1	1	1	1	1	0	0	0
$f(md_i)$	1	2	1	2	1	1	1	1	1	1	1
F_i	**5**	10	**6**	7	**5**	**6**	7	**5**	**5**	**5**	**5**
A_k	{1}	{1, 3}			{1, 3, 2}	{5}		{5, 6}	{8}	{7}	{7, 4}
S_k	14	26			36	23		39	36	20	38
I_k	26	14			4	17		1	4	20	2

9. APPENDIX

Notation:

A_k set of tasks that have been assigned to workstation k
c cycle time
CA_k set of candidate tasks that can be assigned to workstation k
dv_i demand for part i
dv_{max} highest demand
F_i priority function value of task i, where $i \in CA_k$
H binary vector representing whether or not disassembly task i belongs to a part with hazardous content,; i.e., $H = \{h_i; i = 1, \ldots, N\}$
I cumulative idle time of all workstations (idle time of the disassembly line)
$I_{i, k}$ idle time of workstation k when task i is assigned to workstation k
I_k idle time of workstation k
k index for disassembly workstations
KB knowledge base that stores the information related to the product
L duration (or the length) of the planning period (discretely incremented)
M number of workstations; i.e., $k = 1, \ldots, M$
md_i disassembly direction change or not
M_{max} maximum number of workstations (the upper bound)
M_{min} minimum number of workstations (the lower bound)
N number of parts of the product which is equal to the number of tasks; i.e., $i = 1, \ldots, N$
$NZ(R_{i-})$ set of nonzero elements in the row i of matrix R; i.e., $j \in NZ(R_{i-})$ and $r_{ij} = 1$ or d; where $d \in D$
$OG_{i,d}$ OR group, i.e., the set of tasks that OR precedes task i in direction d
q_i number of same part i (i.e., quantity of part i) in the product
R disassembly precedence matrix (DPM); $R = \{r_{ij}; i, j = 1, \ldots, N\}$

$R_{i\text{-}}$	row i of R
$R_{\text{-}j}$	column j of R
S_k	workstation time of k (i.e., total processing time of tasks that have been assigned to workstation k)
sr_i	number of remaining tasks that task i precedes to
T	cumulative duration of all disassembly tasks
t_i	time necessary to perform task i (or operation time of i)
t_{max}	time necessary to complete the longest task
$\lceil x \rceil$	smallest integer $\geq x$

REFERENCES

1. Ansems, A. M. M., Fugger, E., and Scheidt, L.-G. (1994).“Automation of disassembly processes of consumer goods.” Proceedings of the IEEE International Symposium on Electronics and the Environment, San Francisco, CA, May 2-4.
2. de Ron, A., and Penev, K. (1995). “Disassembly and recycling of electronic consumer products: an overview.” Technovation, 15(6), 363-374.
3. Ghosh, S., and Gagnon, R. J. (1989). “A comprehensive literature review and analysis of the design, balancing and scheduling of assembly systems.” International Journal of Production Research, 27(4), 637-670.
4. Gungor, A., and Gupta, S. M. (1999). “Issues in environmentally conscious manufacturing and product recovery: a survey.” Computers and Industrial Engineering, 36, 811-853.
5. Gungor, A., and Gupta, S. M. (2000). “Disassembly sequence plan generation using a branch and bound algorithm.” International Journal of Production Research. (in press)
6. Gupta, S. M., and Taleb, K. N. (1994). “Scheduling disassembly.” *International Journal of Production Research*, **32**(8), 1857-1866.
7. Kizilkaya, E. A., and Gupta, S. M. (1998). “Material flow control and scheduling in a disassembly environment.” Computers and Industrial Engineering, 35(1-2), 93-96.
8. Kochan, A. (1995). “In search of a disassembly factory.” Assembly Automation, 15(4), 16-17.
9. Kopacek, P., and Kronreif, G. (1996).“Semi-automated robotized disassembling of personal computers.” IEEE Conference on Emerging Technologies and Factory Automation, Nov. 18-21, Kauai, Hawaii, 567-572.
10. Menrad, W., Grieger, S., and Schlogl, M. (1999).“Aspects of industrial recycling of electr(on)ic appliances.” Proceedings of the Second International Working Seminar on Re-Use, Eindhoven, The Netherlands, March 1-3, 81-90.
11. Moore, K. E., Gungor, A., and Gupta, S. M. (1998). “Disassembly Petri net generation in the presence of XOR precedence relationships.” *Proceedings of the 1998 IEEE International Conference on Systems, Man and Cybernetics*, La Jolla, California, 13-18.
12. Navin-Chandra, D. (1994). “The recovery problem in product design.” Journal of Engineering Design, 5(1), 65-86.
13. Penev, K. D., and de Ron, A. J. (1994). “Development of disassembly line for refrigerators.” Industrial Engineering, 50-53.
14. Penev, K. D., and de Ron, A. J. (1996). “Determination of a disassembly strategy.” International Journal of Production Research, 34(2), 495-506.
15. Salveson, M. E. (1955). “The assembly-line balancing problem.” Transactions of ASME, 77, 939-948.
16. Tani, K., and Güner, E. (1997). “Concept of an autonomous disassembly system using behavior-based robotics.” Advanced Robotics, 11(2), 187-198.
17. Umeda, K., and Arai, T. (1997). “Three-dimensional vision system for mechanical assembly/disassembly.” Advanced Robotics, 11(2), 147-167.
18. Wiendahl, H.-P., and Bürkner, S. (1999).“Planning and control in disassembly: the key to an increased profitability.” Proceedings of the Second International Working Seminar on Re-Use, Eindhoven, The Netherlands, March 1-3, 247-256.
19. Wiendahl, H.-P., Lorenz, B., and Bürkner, S. (1998). “The necessity for flexible and modular structures in hybrid disassembly processes.” Production Engineering, 5, 71-76.