A New Heuristic Approach to Solvong a Disassembly Scheduling and Process Plannning Problem

Elif Kongar [+]
Surendra M. Gupta [++]
Kenichi Nakashima [+++]

ABSTRACT

This paper deals with a disassembly scheduling and process planning problem for a given End-Of-Life (EOL) product which has recently gained enormous attention as a result of increasing environmental legislation and diminishing natural resources. Once a product reaches its EOL, there are several alternatives available for its processing, *e.g.*, reuse, remanufacture, recycle, storage and proper disposal all requiring a certain level of disassembly. Since disassembly tends to be expensive, disassembly scheduling has been of interest as of late. However, disassembly scheduling is very complex and NP complete and therefore conventional optimization methods of tackling it are unsuitable. Heuristic approaches, on the other hand, can reduce computational time and provide reasonable results. In this paper, we propose a Multiple Objective Tabu Search methodology to solve the disassembly scheduling and process planning problem for a given EOL product. A case example is presented to illustrate the methodology.

[+] Department of Industrial Engineering, Yildiz Technical University, Turkiye
[++] 334 SN, Department of MIME, Northeastern University, 360 Huntington Ave, Boston, MA 02115, USA.
[+++] Department of Industrial Management, Osaka Institute of Technology, 5-16-1 Omiya Asahiku, Osaka 535-8585, Japan.

1. INTRODUCTION

Mandatory policies, consumer awareness and decreasing number of landfills are forcing many companies to take back their products and carry out end-of-life (EOL) processing in a responsible way. The most popular options for EOL processing are reuse, remanufacture, recycle, storage or proper disposal. In majority of EOL processing, a certain level of disassembly may be necessary.

Many products are made up of a large number of components. The optimal disassembly path to retrieve the components can theoretically be obtained using exhaustive search algorithms. However, because of the combinatorial nature of the problem, heuristic methods are often employed to find near-optimal solutions. These approaches can frequently provide satisfactory results much faster and at low costs. In this paper, we present a multiple objective Tabu Search algorithm to address the disassembly problem in the presence of constraints and precedence relationships.

2. LITERATURE REVIEW

Various researchers have studied disassembly, which is one of the fundamental elements for parts and product recovery. Gupta and Taleb [9] and Taleb and Gupta [18] proposed algorithms for scheduling the disassembly of discrete and well-defined product structures. Veerakamolmal and Gupta [19] proposed a method that provides solution for component recovery planning. Recently, Lambert and Gupta [13] addressed the problem of demand driven disassembly using a tree network model. Kuo [12] analyzed the cost of disassembly in electromechanical products. Moore *et al.* [16] used Petri-nets to study products with complex AND/OR relationships.

Boon *et al.* [3] used a multi-objective approach to evaluate the viability of an established and mature recycling infrastructure.

Tabu search is a relatively new heuristic technique, which has become quite popular because of its flexibility in accommodating a wide variety of problems with multiple objectives and multiple constraints [1], [2], [4], [5], [6]. Li *et al.* [15] proposed a tabu search approach to disassembly sequence optimization for maintenance purposes. James and Buchanan [11] proposed a tabu search method for the early/tardy scheduling problem.

For more information on disassembly and product recovery see Gupta and McLean [8], Moyer and Gupta [17], Gungor and Gupta [7] and Lee *et al.* [14].

3. MULTI-OBJECTIVE TABU SEARCH ALGORITHM

Most real world problems have a tendency to be multi-objective. The traditional way of solving multi-objective problems involves prioritizing the various objectives and considering them in a pre-emptive manner or writing a single function by giving relative weights to the various objectives and considering them simultaneously. However, tabu search inherently works with more than one solution at a time (known as neighborhood solutions) providing a natural opportunity to solve multi-objective problems. Tabu search, being a popular and efficient heuristic, has been widely used in the literature because of its problem independent nature and its ability to handle any kind of objective function and any kind of constraints [1]. A typical goal-programming problem [10] can be converted into a tabu search problem thus avoiding the selection of weights before seeing the consequences of choosing them. We explain the conversion process below [2].

3.1 Goal Programming Problem

A general preemptive GP problem can be formally stated as follows [2]:

$$lexmin\{z_1, z_2, \ldots, z_t\}$$
$$\text{where } z_p = \sum_{u=1}^{r} (\eta_u + \rho_u)$$
$$\text{and } z_1 >>> z_2 >>> \ldots >>> z_t$$
$$\text{s.t.}$$
$$g_u(X) + \eta_u - \rho_u = b_u \qquad (1)$$
$$AX \leq \mathbf{0}$$
$$\eta_u, \rho_u \geq 0, X \geq \mathbf{0}$$
$$p = 1, 2, \ldots, t; u = 1, 2, \ldots, r$$

where $g_u(x)$ is the function representing goal u, ρ_u *and* η_u are the positive and negative deviation variables for goal u, b_u is the target value of goal u, r is total number of goals, **X** is an n-dimensional decision vector $(x_1, x_2, \ldots, x_n)$, **A** is the coefficient matrix for the system constraints and **0** is a zero vector. Note that $z_1 >>> z_2$ means that z_1 must be considered before z_2.

3.2 Tabu Search Problem

The goal-programming problem can be converted into a typical multiple objective optimization problem as follows. In goal-programming, the desire to overachieve (minimize η_u) or underachieve (minimize ρ_u) or satisfy the target value exactly (minimize $\rho_u + \eta_u$) is specified for each goal [10]. These desires can be converted as follows [2]. Minimization of both positive and negative deviations is expressed in the form: minimize $|g_u(\mathrm{X}) - b_u|$. Minimizing the positive deviation is expressed in the form: minimize $\langle g_u(\mathrm{X}) - b_u \rangle$. The bracket operator $\langle\ \rangle$ returns the value if it is positive, otherwise it returns zero. Minimizing the negative deviation is expressed in the form: minimize $\langle b_u - g_u(\mathrm{X}) \rangle$. Thus the formulation in the typical multiple objective optimization problem form will appear as follows:

$$\begin{aligned}
&\text{minimize } |g_u(X)-b_u| \text{ or}\\
&\text{minimize } \langle g_u(X)-b_u\rangle \text{ or}\\
&\text{minimize } \langle b_u-g_u(X)\rangle \quad \text{for all } u=1, 2,\dots, r\\
&\text{s.t.}\\
&\quad \boldsymbol{AX}\leq \boldsymbol{0}\\
&\quad \boldsymbol{X}\geq \boldsymbol{0}
\end{aligned} \tag{2}$$

Even though the above problem cannot be solved in the traditional way, tabu search can easily cope with it.

Tabu search starts with an initial solution and generates multiple neighborhood solutions from it. If not controlled, this process can be "trapped" at a local optimum, re-visiting the same solution after one or many neighborhood generations. To avoid this cycle, a so-called *Tabu list* is employed to keep track of each solution. The algorithm has two more lists in addition to the Tabu list. *Pareto list* collects the selected non-dominated solutions that are not chosen as Pareto optimal solutions found by the algorithm. *Candidate list* collects all other non-dominated solutions. If they maintain their non-dominated status, they can be selected as the seed later (see below).

Tabu search uses the number of variables (*nvar*), number of functions (*nfn*), neighborhood size (*nneigh*), lower and upper bounds for each variable (*LB*(*var*) and *UB*(*var*) respectively), objective functions (*obj_fn* ( )) and other problem specific data as input. After the initial data is established, an initial feasible solution is obtained, called the *seed* solution and denoted by s. From s, a set of neighborhood solutions (s^*) is generated using pre-determined moving strategies. The neighborhood solutions in s^* are then evaluated and the best neighbor is chosen to replace the seed solution. If no neighbor is found that dominates the seed, a corrective algorithm alters the step size and generates a new set for neighbors of size *nneigh*. A neighbor for integer variables can be generated as follows [2]:

$$x_i^* = x_i + \text{integer}\,[(2.\,\text{random}\,() - 1).\; stepi_i] \qquad (3)$$

where, x_i is the value of the i^{th} variable prior to finding the neighborhood solution, x_i^* is the value of the i^{th} variable after finding the neighborhood solution, random( ) $\in \{0, 1\}$, $stepi_i$ is the step size of the i^{th} integer variable, integer [ ] is the function to round a real number to its nearest integer value. After the generation of s^*, all current neighborhood solutions are eliminated which are dominated by any other current neighbor. Remaining solutions are then defined as candidate solutions. If there are candidate solutions, one is selected at random to become the new seed *s*. If there is no candidate solution found, the counter is incremented. Then all the dominated solutions from the Pareto and Candidate lists are eliminated. After this, *s* is added to the Pareto List and the Tabu list while the remaining candidate neighbors are added to the Candidate list. Any neighbors that are not dominated by the seed solution and the solutions in the Pareto and Candidate lists are accepted, even though they are tabu. If there is no candidate solution in the current neighborhood the oldest solution from the Candidate list is selected as the new seed *s*. The algorithm terminates if a maximum number of empty candidates, or allowable iteration number is reached. Later, Pareto list is obtained with *all* the non-dominated possible solutions.

4. PROBLEM DEFINITION AND FORMULATION

The problem under consideration is as follows. The EOL products are taken back from the last users and/or collectors and brought into the facility where the products are prepared for further processing. The EOL products are disassembled for reuse, recycling, storage and/or proper disposal. If there is no demand for an item, it is either stored or sent to disposal.

The proposed formulation consists of two independent modules, namely, disassembly processing (DP) and disassembly scheduling (DS). The output from the modules provide the feasible disassembly sequences, all the cost and revenue functions and various other performance measures such as the number of disposed items (*NDIS*) and the number of recycled items (*NRC*), in response to the desired multiple predetermined goals.

4.1 Disassembly Processing Module

This module formulates the problem as a GP. It can then be converted into a tabu search problem (as explained above) to obtain near optimal/optimal solutions (if more than one is available).

The total number of item i disassembled (TD_i) includes the number disassembled for reuse (X_i), the number disassembled for recycling (R_i), the number disassembled for storage (V_i), and the number disassembled for disposal (L_i), where $i=0, ..., n-1$, and n is the number of different items in the EOL product.

Hence, mathematically,

$$TD_i = X_i + R_i + V_i + L_i,\ \forall\, i,\ i = 0, \ldots, n-1, \text{ and} \tag{4}$$

$$TD = \sum_{i=0}^{n-1} (X_i + R_i + V_i + L_i), \tag{5}$$

where TD is the total number of all disassembled items.

The input data of this module includes the number of items demanded for reuse (D_i) and recycling (DR_i), and the reuse (PR_i) and recycling price (PRC_i) for each item. In addition, the step size for each variable

($stepi_i(var)$) is also defined. Since no backordering is allowed, the number of disassembled items for reuse (X_i) and recycling (R_i) is equal to the demand for reuse (D_i) and recycling (DR_i). Therefore,

$$D_i = X_i, \forall\, i,\ i = 0, \ldots, n-1 \tag{6}$$

$$DR_i = R_i, \forall\, i,\ i = 0, \ldots, n-1 \tag{7}$$

We consider three objective functions, as follows.

The first objective function, (Obj_fn_p (1)) is to maximize the total profit (TPR) of the system. TPR is defined as the difference between the revenues and costs of the system. The revenues consist of the sum of resale (RPS) and recycling (RMS) functions. The cost function consists of the sum of storage (CST) and disposal ($CDIS$) cost functions. Mathematically:

$$Obj\_fn\_p(1) = \max(TPR) \tag{8}$$

$$\text{where } TPR = RPS + RMS - CST - CDIS \tag{9}$$

RPS is a function of all the disassembled items for reuse (X_i) and the resale price (PR_i) of the item:

$$RPS = \sum_{i=0}^{n-1} (PR_i . X_i). \tag{10}$$

RMS is a function of all the disassembled items for recycling (R_i) and the recycling price (PRC_i) for the corresponding item:

$$RMS = \sum_{i=0}^{n-1} (PRC_i . R_i). \tag{11}$$

CST is a function of all the disassembled items for storage (V_i) and the unit holding cost (h_i) of the item:

$$CST = \sum_{i=0}^{n-1} (h_i . V_i). \tag{12}$$

$CDIS$ is a function of all the disassembled items for disposal (L_i) and their unit disposal costs ($UCDI_i$):

$$CDIS = \sum_{i=0}^{n-1} (UCDI_i . L_i). \tag{13}$$

The second objective function, (Obj_fn_p (2)), is to maximize the ratio of the sum of total number of reused ($NRES$) and recycled items (NRC) to the total number of disassembled items (TD). Mathematically,

$$Obj\_fn\_p(2) = \max DS \tag{14}$$

$$\text{where } DS = [(NRES + NRC) / TD] \tag{15}$$

$$NRES = \sum_{i=0}^{n-1} X_i, \text{ and,} \tag{16}$$

$$NRC = \sum_{i=0}^{n-1} R_i,. \tag{17}$$

The third objective function (Obj_fn_p (3)), is to minimize the ratio of disposed items ($NDIS$) to the total number of disassembled items (TD). Mathematically,

$$Obj\_fn\_p(3) = \min DL \tag{18}$$

$$\text{where } DL = [(NDIS) / TD] \tag{19}$$

$$NDIS = \sum_{i=0}^{n-1} V_i, \tag{20}$$

The GP model can now be written as follows:

Find (X_i, R_i, V_i, L_i) so as to:

$$\text{Lexicographically min. } z = \{(\eta_1), (\eta_2), (\rho_3)\} \tag{21}$$

where,

$$TPR + \eta_1 - \rho_1 = TPR^* \tag{22}$$

$$DS + \eta_2 - \rho_2 = DS^* \tag{23}$$

$$DL + \eta_3 - \rho_3 = DL^* \tag{24}$$

and TPR^*, DS^* and DL^* are the aspiration levels for TPR, DS and DL respectively.

subject to:

eq. (4−20), $\{\eta_u, \rho_u\} \geq 0$, $\forall u = 1, 2, 3$, and $X_i, R_i, V_i, L_i \geq 0\ \forall\, i, i = 0, \ldots, n-1$.

4.2 Disassembly Sequencing Module

Similar to the previous module, this module also uses tabu search to obtain near optimal/optimal solutions (if more than one is available).

In this module, the EOL product is represented by a $(6\times n)$ matrix. The rows of the matrix represent the item numbers $(i=0, 1, 2..., n-1)$, the disassembly directions $(dr_i=(+x, +y, +z, -x, -y, -z)$, the disassembly method ($mt_i=N$: non-destructive, D: destructive), the type of demand ($tp_i=0$: non-demanded, 1: demanded for reuse, 2: demanded for recycling), the required disassembly time (tm_i=time in seconds) and the due time for each component (due_i=time in seconds).

We consider three objective functions, as follows.

The first objective function, (Obj_fn_s (1)), tries to obtain the demanded items as early as possible and the non-demanded items as late as possible [11].

$$Obj\_fn\_s(1)=\min\sum_{i=0}^{n-1}(\alpha_i|d_i-c_i|^{+}+\beta_i|c_i-d_i|^{+}), \tag{25}$$

where, c_i is the disassembly time of component i, d_i is the due time of component i, α_i and β_i are the penalties per unit of time when component i is disassembled early and tardy respectively. $|\boldsymbol{x}_i|^{+}=x$ if $x>0$; 0 otherwise.

The second objective function aims to minimize the number of direction changes. Hence, each item that requires a direction change for its disassembly is penalized by one. Mathematically, the second objective function can be expressed as follows:

$$Obj\_fn\_s(2)=\min\sum_{i=0}^{n-1}(\gamma_i), \tag{26}$$

where γ_i is the penalty for direction change for item i, which takes a value of 0 if there is no direction change, 1 if the direction change is 90 degrees and 2 if the direction change is 180 degrees.

The third objective function is for minimizing the number of disassembly method changes. Hence, each item that requires a method change for its disassembly is penalized by a positive number. Mathematically, the third objective function can be expressed as in below:

$$Obj\_fn\_s(3) = \min \sum_{i=0}^{n-1} (\kappa_i), \tag{27}$$

where κ_i is the penalty for disassembly method change for item i and takes the value of 0 if there is no method change and 1 otherwise.

5. NUMERICAL EXAMPLE

Consider the ten-item EOL product in Figure 1.

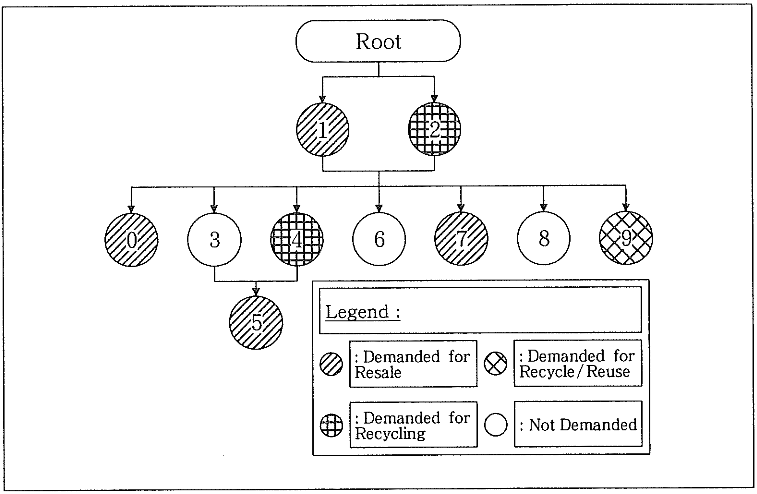

Figure 1. EOL Product Structure

In this example $nneigh=3$, $stepi_i = 5$, $\forall X_i$ and R_i, and $stepi_i=5$, $\forall V_i$ and L_i. In addition, $LB(TD_i)=\{20, 40, 30, 0, 25, 50, 0, 30, 0, 35\}$, $LB(X_i)=\{20, 0, 30, 0, 0, 50, 0, 30, 0, 35\}$, $LB(R_i)=\{0, 40, 0, 0, 25, 0, 0, 0, 0, 35\}$, $LB(V_i)=0\ \forall i$, $LB(L_i)$

$=0\ \forall i$, $LB(X_i)=\{20, 40, 30, 0, 25, 50, 0, 30, 0, 35\}$, $UB(TDi)=100\ \forall i$, $UB(X_i)=\{20, 0, 30, 0, 0, 50, 0, 30, 0, 35\}$, $UB(R_i)=\{0, 40, 0, 0, 25, 0, 0, 0, 0, 35\}$, $UB(V_i)=\{80, 60, 70, 100, 0, 50, 0, 70, 100, 0\}$, $UB(L_i)=\{0, 0, 0, 0, 75, 0, 100, 0, 0, 30\}$, $PR_i=\{10, 9, 8, 10, 7, 9, 8, 10, 8, 10\}$, $PRC_i=\{5, 6, 5, 4, 5, 4, 3, 5, 2, 1\}$, $UCDI_i=\{1, 2, 1, 1, 2, 2, 1, 1, 2, 2\}$, and $h_i=\{3, 2, 3, 2, 1, 3, 2, 1, 2, 3\}$ for, $i=\{0, ..., 9\}$. The precedence relationship data is as follows: Items 1 or 2 has to be disassembled prior to any other item and item 5 has to be disassembled after item 3 and item 4. The data for the $(6\times n)$ matrix is given as follows.

i	0	1	2	3	4	5	6	7	8	9
dr_i	$+x$	$-x$	$+x$	$-z$	$-x$	$+x$	$+z$	$+y$	$-Y$	$+Y$
mt_i	N	D	N	D	D	N	D	N	D	N
tp_i	1	1	1	0	2	1	0	1	0	1
tm_i	2	2	3	2	3	3	2	1	2	2
due_i	0	0	0	18	0	0	18	0	18	0

In order to obtain the aspiration levels for each goal, an LP model is solved for each objective function in a preemptive manner. Hence, we first solved the LP model with maximizing *TPR* (LP I model) being the objective function and obtained the corresponding results for *DS* and *DP*. Similarly the LP is solved with two more objective functions, maximizing *DS* (LP II model) and minimizing *DL* (LP III model). After the solution we obtained $TPR=\{965, 760, 965\}$, $DS=\{0.265, 0.265, 0.265\}$, $DL=\{0.205, 0, 0.205\}$ for LP I, II and III respectively. Based on the results, aspiration levels are set to $TPR^*=965$, $DS^*=0.265$, $DL^*=0$.

For the disassembly processing module, we obtain three solutions. Each solution is based on 100 EOL products to be disassembled. Here, while Pareto 1 list provides and objective function set (760, 0.265, 0.000), Pareto 2 suggests a set (840, 0.070, 0.080), which has a higher total profit function (*TPR*), but a lower reuse and recycling rate (*DS*) with a less desirable ratio of disposed items (DL). The third Pareto suggests a set (780, 0.265, 0.020) which is not dominated by any of the other solutions.

For the disassembly scheduling module, we obtain five different results. All aspiration levels are set to zero. Pareto 1 provides a sequence as (2, 5, 1, 4, 0, 6, 7, 8, 9, 3) with an objective function set $Obj\_fn\_s = (63, 11, 7)$. Pareto 2 provides a sequence as (2, 5, 1, 0, 4, 8, 6, 3, 9, 7) with an objective function set $Obj\_fn\_s = (64,11,4)$. Pareto 3 provides a sequence as (2, 9, 1, 5, 0, 4, 7, 6, 8, 3) with an objective function set $Obj\_fn\_s = (65,10,5)$. Pareto 4 provides a sequence as (2, 1, 7, 0, 5, 4, 6, 3, 8, 9) with an objective function set $Obj\_fn\_s = (53, 12, 4)$. Pareto 5 provides a sequence as (1, 2, 9, 7, 5, 6, 0, 3, 8, 4) with an objective function set $Obj\_fn\_s = (76, 9, 4)$.

6. CONCLUSIONS

A multi-objective tabu search methodology was presented in order to determine the feasible disassembly sequence and disassembly process of a given EOL product. The algorithm provided multiple solutions while achieving predetermined goals. The disassembly-sequencing module preserves the precedence relationships for disassembly. The model avoids finding the weights for the goals and hence is considered easy to use.

REFERENCES

[1] Baykasoglu, A., Owen, S. and Gindy, N., "Solution of Goal Programming Models Using a Basic Taboo Search Algorithm", *Journal of Operational Research Society*. 50, 960-973, 1999.

[2] Baykasoglu, A., "Goal Programming Using The Multiple Objective Tabu Search", *Journal of Operational Research Society*, 52, 1359-1369, 2001.

[3] Boon, J. E., Isaacs, J. A., and Gupta, S. M., "Economic Sensitivity for End Of Life Planning and Processing of Personal Computers", *Journal of Electronics Manufacturing*, 11(1), 81-93, 2002.

[4] Glover, F., "Tabu Search, Part I", *ORSA Journal of Computing*, 1(3), 190-206, 1989.

[5] Glover, F., "Tabu Search, Part II", ORSA *Journal of Computing*, 2(1), 4-32, 1990.

[6] Glover, F., "Tabu Search: A Tutorial", *Interfaces*, 20(4), 74-94, 1990.

[7] Gungor, A. and Gupta, S. M., "Issues in Environmentally Conscious Manufacturing and Product Recovery: A Survey", *Computers and Industrial Engineering*, 36(4), 811-853, 1999.

[8] Gupta, S. M. and McLean C. R., "Disassembly of Products", *Computers and Industrial Engineering*, 31, 225-228, 1996.

[9] Gupta, S. M. and Taleb, K., "Scheduling Disassembly", *International Journal of Production Research*, 32(8), 1857-1866, 1994.

[10] Ignizio, J. P., Linear Programming in Single and Multiple Objective Systems, Prentice Hall, 1982.

[11] James, R. J. W. and Buchanan, J. T., "Performance Enhancements to Tabu Search for the Early/Tardy Scheduling Problem", *European Journal of Operational Research*, 106, 254-265, 1998.

[12] Kuo, T. C., "Disassembly Sequence and Cost Analysis for Electromechanical Products", *Robotics and Computer Integrated Manufacturing*, 16(1), 43-54, 2000.

[13] Lambert, A. D. J. and Gupta, S. M., "Demand-Driven Disassembly Optimization for Electronic Products", *Journal of Electronics Manufacturing*, 11(2), 121-135, 2002.

[14] Lee, D.-H., Kang, J.-G. and Xirouchakis, P., "Disassembly Planning and Scheduling: Review and Further Research", *Journal of Engineering Manufacture*, 215 (B5), 695-709, 2001

[15] Li, J. R., Tor, S. B. and Khoo. L. P., "Integrating Tabu Search and Genetic Algorithm for Disassembly Sequence Optimization for Maintenance", *Proceedings of the 16th International Conference of Production Research*, Prague, Czech Republic, 29 July - 3 August 2001.

[16] Moore, K. E., Gungor, A. and Gupta, S. M., "Petri Net Approach to Disassembly Process Planning for Products with Complex AND/OR Precedence Relationships", *European Journal of Operational Research*, 135(2), 189-210, 2001.

[17] Moyer, L. K. and Gupta, S. M., "Environmental concerns and Recycling/ Disassembly Efforts in the electronic industry", *Journal of Electronics Manufacturing*, 7(1), 1-22, 1997.

[18] Taleb, K. and Gupta, S. M., "Disassembly of Multiple Product Structures", *Computers and Industrial Engineering*, 32(4), 949-961, 1997.

[19] Veerakamolmal, P. and Gupta, S. M., "Optimal Analysis of Lot Size Balanc-

ing for Multi-Products Selective Disassembly", *International Journal of Flexible Automation and Integrated Manufacturing*, 6(3/4), 245-269, 1998.