

DETC2011-4, * ')

AN ONLINE GENETIC ALGORITHM FOR AUTOMATED DISASSEMBLY SEQUENCE GENERATION

A. ElSayed

Department of Computer Science and Engineering
University of Bridgeport
Bridgeport, CT 06604
Email : aelsayed@bridgeport.edu

E. Kongar*

(*Corresponding author)
Department of Mechanical Engineering and
Technology Management, University of Bridgeport
Bridgeport, CT 06604
Email : kongar@bridgeport.edu
Phone: 203-576-4379; Fax: 203-576-4750

S. M. Gupta

Department of Mechanical Engineering and
Industrial Engineering
Northeastern University
Boston, MA 02115
Email : gupta@neu.edu

T. Sobh

Department of Computer Science and Engineering
University of Bridgeport
Bridgeport, CT 06604
Email : sobh@bridgeport.edu

ABSTRACT

In this study, we propose an intelligent automated disassembly cell for online (real time) selective disassembly. The cell is composed of an industrial robotic manipulator, a camera, range-sensing and component segmentation visual algorithms. The cell prototype allows for robotic sensory-driven disassembly under uncertainty. An online genetic algorithm model for selective disassembly is also proposed for optimal and near/optimal disassembly sequencing.

INTRODUCTION

Advanced technology products are regularly rendered technically obsolete within a few years of commercialization due to the rapid pace of technological developments. Thus, electronic products are frequently discarded before their materials degrade. These complex End-of-Life (EOL) products contain a broad spectrum of materials including precious metals such as silver, and valuable materials such as copper. Therefore, efficient recovery of materials in the electronic EOL products is essential for economic reasons. Furthermore, the conceptual lifetime of an electronic product depends primarily on the pace of superseding technological change that makes obsolete the otherwise fully-functioning

product. Hence, the discarded product is likely to contain one or more usable components. The economically and environmentally sustainable option is to reuse these components in technically valid products. EOL processing options, e.g., reuse, recycle, and remanufacturing, are effective ways to regain the materials and the components in electronic EOL products. Regardless of the motivation, most EOL processing options necessitate a certain level of disassembly. Disassembly operations are very complex, time-consuming, and expensive. Therefore, limiting the disassembly operations to recyclable materials and reusable components in the EOL product is crucial in order to render electronic waste recovery operations economically viable. Responding to this need, generating disassembly algorithms for selective disassembly [1-2] is the first major research topic presented in this paper.

Electronic waste recovery, EOL processing options, and disassembly of electronic products are well documented [3-16]. However, related literature falls short in providing realistic methods for product recovery. Despite the ever growing research on disassembly of electronic products, the proposed methods generally assume that the part locations are known in the product structure. Furthermore, the majority of these algorithms function under the assumption of a known hierarchical disassembly path, i.e., for a personal computer (PC) disassembly, the front or side cover must be removed to

access the power supply, and the power supply must be removed to access the processor and the fan, ... etc. However, end-of-life (EOL) electronic products - specifically PCs - are unlikely to preserve their original product structure throughout their useful lives. The bill-of-materials (BOM) is likely to be altered by the user for a variety of reasons such as repair, upgrade, or personal configuration preferences. Hence, the EOL product is likely to embody missing or replaced parts. This is also true for the fastener structures; when a part is taken out or replaced with another, the type and the location of fasteners are likely to change. RAM slots and PCI slots are good examples for this type of uncertainty since these parts are more frequently replaced compared to the rest of the BOM.

In this study, we propose an intelligent automated disassembly cell utilizing a setting composed of an industrial robotic manipulator, a camera, range-sensing and component segmentation visual algorithms to design a prototype of a robotic sensory-driven disassembly cell. Figure 1 depicts the Mitsubishi Industrial Micro-Robot System Model RV-M1 and HP Pavilion 6370Z Desktop PC with an additional hard drive and three RAM modules.

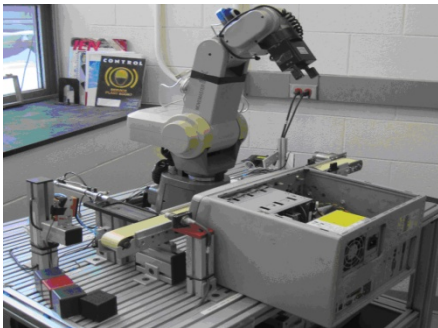


FIGURE 1. THE RISC LABORATORY MITSUBISHI MOVEMASTER INDUSTRIAL ROBOTIC MANIPULATOR, CAMERA, AND EOL PC.

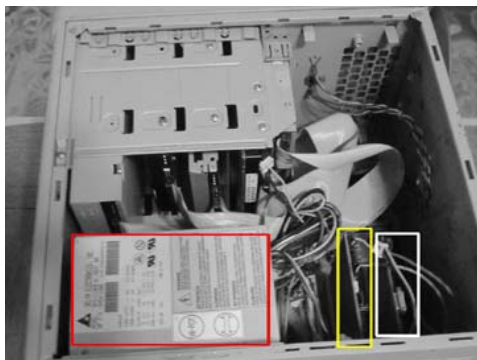


FIGURE 2. CAMERA OUTPUT FOR THE POWER SUPPLY, SOUND CARD, AND MODEM CARD DETECTION WITHIN THE EOL PC.

The proposed model aims at handling uncertainty in the EOL product structure and consists of two modules: (1) a sensory-driven visual and range acquisition and recovery system, and (2) an online genetic algorithm (GA) model. For the sensory processing system, a sensory module is designed to acquire and recognize part descriptions and coordinates via the usage of range and 2-D sensors; complemented with the appropriate pattern matching vision algorithms (Figure 2). The visual part-recognition software module output is then fed into the GA algorithm that generates online optimal and/or near-optimal disassembly sequences for the detected parts in the product structure (Figure 3).

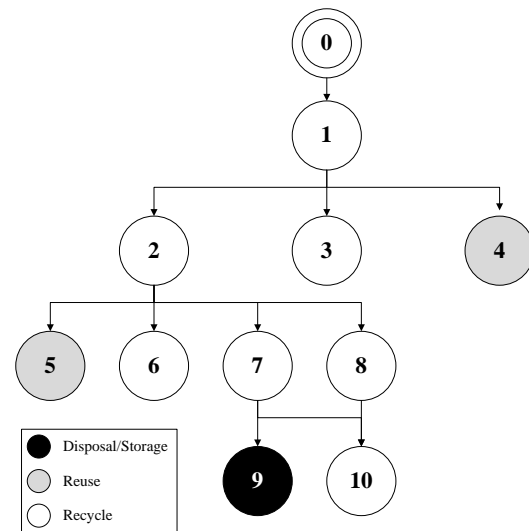


FIGURE 3. THE BILL-OF-MATERIALS OF THE PC WITH CORRESPONDING END-OF-LIFE PROCESSING OPTIONS.

In Figure 3, “0” is the robot reference point where the disassembly operation will start. Table 1 lists the remaining component information with corresponding material, disassembly time, and disassembly method information. Basically, if the component is subject to reuse, non-destructive disassemble is selected as the appropriate disassembly operation to preserve the physical structure of the component. If the component is subject to recycling, (i.e., the material value is important,) the faster disassembly option - destructive disassembly - is preferred. The components that do not contain any market value are subject to storage or proper disposal and are not disassembled unless the precedence relationships mandate their disassembly for the remaining components.

TABLE 1. COMPONENTS OF THE EOL PC WITH CORRESPONDING MATERIAL AND DISASSEMBLY OPERATION INFORMATION.

i	Description	Material	Disassembly	
			Time dt_i (sec.)	Method*
0	Robot Reference Point			
1	Side Cover	Aluminum (A)	3	D
2	Power Supply	Copper (C)	6	D
3	Sound Card	Plastic (P)	3	N
4	Modem Card	Plastic (P)	3	N
5	CPU	Plastic (P)	5	N
6	Hard Drive	Aluminum (A)	4	N
7	CD Drive	Aluminum (A)	4	N
8	Zip Drive	Aluminum (A)	4	N
9	RAM	Plastic (P)	3	N
10	Drives Slot	Aluminum (A)	2	D

*D: Destructive N: Non-destructive

BACKGROUND

Disassembly sequencing problems are combinatorial and NP-complete, prohibiting utilization of exhaustive search techniques. Therefore, genetic algorithms (GA) have been gaining popularity for such problems. One of the multi-objective optimization applications was proposed by Valenzuela-Rendón and Uresti-Charre [17]. Keung *et al.* [18] also applied a multi-objective GA approach to a tool selection model. Lazzerini and Marcelloni [19] used GA in scheduling assembly processes. In the area of disassembly, Kongar and Gupta [20] proposed a GA for disassembly sequencing problems, while McGovern and Gupta [21] applied genetic algorithm to disassembly line balancing. For a thorough review of work done in the area of product recovery, see Ilgin and Gupta [16], Gungor and Gupta [15], Lee *et al.* [22], and Lambert and Gupta [23].

Precedence relationships that must be preserved during disassembly operations constitute another factor that increases the computational complexity of disassembly sequence planning problems. Sanderson *et al.* [24] proposed a methodology considering precedence relationships in

assembly sequence planning. Seo *et al.* [25] proposed a GA for optimal disassembly sequence generation considering economical and environmental factors. Bierwirth *et al.* [26] and Bierwirth and Mattfeld [27] proposed precedence preservative crossover (PPX) technique for scheduling problems. The algorithm, which is also employed in this study, preserves the precedence relationships in the product structure. Hui *et al.* [28] utilized a genetic algorithm to determine good disassembly sequences by converting disassembly sequence planning problem into a searching problem on an information-enhanced graph. Shimizu *et al.* [29] developed a prototype system for strategic decision-making on disassembly for recycling at the design stage of the product life cycle.

In the area of automated disassembly, Torres *et al.* [30-31] presented a personal computer disassembly cell that is able to handle a certain degree of automatism for the non-destructive disassembly process. This work was then followed by Pomares *et al.* [32] who generated an object-oriented model required for developing a disassembly process. Gil *et al.* [33] proposed a flexible multi-sensorial system for automatic disassembly using cooperative robots. As a follow-up work, Torres *et al.* [34] presented a task planner for a disassembly process based on decision trees.

In this paper, we present a GA-based technique for generating online adaptive disassembly sequences for partial disassembly. Online adaptive systems proposed by Milani [35], allow modeling highly evolutionary domains where solution environment are subject to changes over time. Therefore, the model is very suitable for disassembly sequencing problems since the solution environment changes following each disassembly action due to the modifications in the bill-of-materials.

MATERIALS AND METHODS

Visual Processing via Template/Pattern Matching

Template/pattern is defined as anything fashioned, shaped, or designed to serve as a model from which something is to be made: a model, design, plan or outline, where as *matching* is the act of comparing in respect of similarity; to examine the likeness or difference [36]. Our visual processing algorithm utilizes a coarse-to-fine 3-D recovery mechanism. Range sensing and coarse scene segmentation are initially performed on the EOL PC to derive a crude 2½-D map of the major components within the PC, in order to reduce the search space for the 2-D camera-based algorithm to be used next to identify specific components in the BOM. A correlation/convolution-based 2-D template matching module is then executed, after the appropriate scaling, rotation and histogram-equalization algorithms are implemented, in order to filter visual noise and to speed up the template matching process. Components that are matched with a high confidence level are then tagged and a cross-reference - given the calibrated camera parameters - is then performed to recover the planar and depth parameters of

the recognized parts, in order to guide the manipulator end-effector (gripper) for eventual disassembly.

Genetic Algorithm

The disassembly sequence generation module is an improved version of previously published work and obtains near-optimal and/or optimal disassembly sequences in a step-wise manner. The algorithm proposed in [2] functions under the assumptions of: (1) the BOM preserves its original structure, (2) the part coordinates are available, and (3) the precedence relationships for removal are known. The algorithm then disassembles the end-of-life PC to remove the parts demanded for reuse and/or recycling. The proposed algorithm releases the above assumptions and utilizes only the demand information to launch the sensory vision module. The module generates 2 and 2½-D maps of the EOL PC and produces part matches. The recognized parts and their coordinates are then fed into the GA module that generates a near-optimal and/or optimal disassembly sequence for the detected parts. The disassembly sequence with the appropriate part coordinates is then transferred to the robot arm manipulator. After the removal of each identified part the dynamically active sensory modules detect the parts that become accessible. If no new part is detected the current sequence is continued. If a new part is detected a new sequence is generated for the newly accessible parts and the remaining parts detected in the previous sequence.

The steps of the proposed GA can be summarized as follows (Table 2).

Initial Population: The initial population consists of hundred ($n_{cr}=100$) random chromosomes that satisfy the precedence relationships and any other constraints imposed by the product structure.

Crossover: The precedence preservative crossover (PPX) methodology is utilized for the crossover operation. Here, in addition to the parents ($Parent_1$ and $Parent_2$), two additional strings ($Child_1$ and $Child_2$) pass on the precedence relationship based on the two parental permutations to two new offspring while making sure that no new precedence relationships are introduced. A vector, representing the number of operations involved in the problem is randomly filled with elements of the set. This vector defines the order in which the operations are successively drawn from $Parent_1$ and $Parent_2$.

TABLE 2. STEPS OF THE PROPOSED GENETIC ALGORITHM.

Step 1. Start with a population of (n_{cr}) random individuals each with chromosome length (chl). .

Step 2. Calculate the fitness $F(ch,gn)$ of each chromosome (ch) in the generation (gn).

Step 3. Permute the current chromosomes.

Step 4. Select the first 40 chromosomes for the next generation.

Step 5. Select 60% of the remaining chromosomes for crossover.

Step 6. Randomly calculate the crossover probability. If probability holds perform precedence preservative crossover (PPX) operation, and generate the children for the next generation.

Step 7. Randomly calculate the mutation probability. If probability holds perform the mutation operation on a randomly selected number (rnd) of chromosomes starting from the first chromosome, generate the output of the mutation as a new population; if not, define the current population as the new population.

Step 8. Return to Step 2 until the new generation contains n_{cr} chromosomes. Then replace the old population with the new generation.

Step 9. If the termination condition is satisfied, STOP, else return to Step 2.

The PPX algorithm generates an empty offspring. The leftmost operation in one of the two parents is selected in accordance with the order of parents given in the vector. After an operation is selected it is taken out from both parents and is appended to the offspring. This step is repeated until both parents are empty and the offspring contains all operations involved.

Mutation: The population is subjected to mutation operation with a given probability. If the probability threshold holds, perform the mutation operation on a randomly selected number of chromosomes (rnd) where ($0 < rnd < n-1$), and swaps any two random components in all selected chromosomes without violating the precedence relationships and the overall feasibility (Table 3).

Fitness Evaluation: The fitness value is obtained by the overall disassembly time. Disassembly time involves the time spent for the disassembly operations, the travel time for the robot arm in 3D space, the penalty for method change, and the award for recycling pairs. The first time parameter is the basic disassembly time for component i in sequence j , (dt_{ij}).

TABLE 3. SYSTEM PARAMETERS OF THE PROPOSED GENETIC ALGORITHM..

Parameter	Value/Explanation
Initial population	Random and feasible
Population size	ncr = 100
Length of the chromosome	chl= n * 5 (n changes in each level)
Max. number of generations	100
Crossover operator	Precedence preservative crossover (PPX)
Mutation operator	Applies to a random number (rnd) of chromosomes (0<rnd<n-1) and swap any two components
Crossover probability	0.60
Mutation probability	0.005
Selection procedure	Roulette selection
Regeneration procedure	Selected chromosomes are cloned to keep the population size constant
Fitness parameters	Basic disassembly time, travel time for robot arm in 3D space, penalty for method change, penalty for pairs
Assumptions	Every component is assumed to have one joint that connects the component with the rest of the product structure

The second function (tt_{ij}) is the penalty (in seconds) for each travel time to disassemble component i in sequence j , which includes a function of the distance traveled between the $(j-1)^{th}$ and j^{th} sequences and the robot arm speed factor (sf):

$$tt_{ij} = \frac{\sqrt{\left(x_{i(j-1)} - x_{ij}\right)^2 + \left(y_{i(j-1)} - y_{ij}\right)^2 + \left(z_{i(j-1)} - z_{ij}\right)^2}}{sf}$$

The proposed model assumes that the end effector speed for the robot arm is a constant value of 7 cm/sec. In addition, the time spent for the robot arm angle change (for all three angles) is assumed to be embedded in the disassembly time for each component.

The third criterion in the fitness function is the penalty for disassembly method change (mt_{ij}), for the time spent for tool change during the disassembly operations. For each disassembly method change, the sequence is penalized by 1 second:

$$mt_{ij} = \begin{cases} 0, & \text{If no method change is required, (e.g. N to N)} \\ 1, & \text{If method change is required, (e.g. N to D)} \end{cases}$$

In addition, the algorithm searches for a “recycling pair” and does not penalize the sequence if the two adjacent components are made of the same material and if they are both demanded for recycling.

Let T_j denote the cumulative disassembly time after the disassembly operation in sequence j is completed for component i . In the case where there is no recycling pair, the overall disassembly time for sequence j is calculated as:

$$T_j = T_{j-1} + dt_{ij} + tt_{ij} + mt_{ij}, \text{ for } j = 1, \dots, n-2.$$

In the case where there is a recycling pair, the travel and method change times are omitted from the equation. Hence, the overall penalty for sequence j can be calculated as:

$$T_i = T_{j-1} + dt_{ij}, \text{ for } j = n-1 \quad (1)$$

In this proposed GA model, the objective is to minimize the total fitness function (F) by minimizing (i) the traveled distance, (ii) the number of disassembly method changes, and (iii) by combining the identical-material components together, eliminating unnecessary disassembly operations. Let $F(ch, gn)$ denote the total fitness for chromosome ch in generation gn . Hence, total time to disassemble all the components can be calculated as in Eq. 2:

$$F(ch, gn) = \sum_{j=0}^{n-1} dt_{ij} + \sum_{j=0}^{n-2} ct_{ij} + \sum_{j=0}^{n-2} mt_{ij}, \quad (2)$$

$$\forall j, j = 0, \dots, n-1.$$

Selection and Regeneration Procedure: After every generation, the chromosomes obtain a certain expectation depending on their fitness values. A roulette wheel is then implemented to select the sequence of parents that will be included in the next generation (the higher the fitness value the higher the chance to be selected). This method aims at allowing the parents in the current generation to be selected for the next generation without getting trapped in the local optima. In addition, a new population is generated eliminating the weak chromosomes.

Termination: The algorithm is terminated when the maximum number of generations is exceeded or no further improvement is obtained.

RESULTS

The proposed algorithm optimizes the disassembly sequence for the above defined fitness function at every step when a new template is detected. For the provided EOL PC, the algorithm starts with the generated sequence (2 3 4), and removes component 2 (power supply). Detection is then reactivated after the removal, and components 5, 6, 7, and 8 are detected. A new sequence is then generated to include newly detected components and the components that are not yet removed from the BOM (components 3 and 4). Following

this, a new sequence is generated (7 8 6 3 5 4). After the removal of component 7, the detection is reactivated and component 9 is detected, followed by a new sequence generation to include the still intact components (8 6 3 5 4) and newly detected component (9). This is followed by a new sequence generation. The algorithm continues until all the reusable and recyclable components are taken out from the EOL product structure.

Table 4 summarizes the model results and provides detection times along with the time it takes to optimize the sequence for each template. The best sequence obtained in each step, the corresponding fitness values are also provided in the table.

TABLE 4. RESULTS OF THE PROPOSED GENETIC ALGORITHM.

Step	Detection time (sec.)	Optimization time (sec.)	Best Sequence	Fitness Value (sec.)
1	0.45	N/A	1	24.74
2	0.30	0.4680	2 3 4	49.16
3	0.12	0.1248	7 8 6 3 5 4	61.89
4	0.08	0.3276	8 6 3 4 5 9	60.99
5	0.02	0.3744	10 6 3 4 5 9	64.24

Figure 4 depicts the visual images captured by the camera for each template.

CONCLUSIONS AND FUTURE WORK

The main advantage of the proposed method is its ability to handle uncertainty, allowing flexibility in the disassembly sequence generation. Furthermore, the method releases unrealistic assumptions and constraints in the GA population allowing improvement in the optimization computations. The algorithm produces reliable and accurate disassembly sequences. The 2½-D module utilized to map the depth via a range finder increase both the accuracy and computation time for the visual segmentation and matching framework. The area of automated disassembly research is relatively new, and even though few studies demonstrate electronic disassembly cells [30-34, 37], this research project is one of very few environmentally driven and economically benign disassembly applications combining robotics research with much needed sustainability endeavors. In addition, the online selective disassembly sequencing algorithm proposed in the study further improves the existing body of research in the area of disassembly. Future work will include improving the system capability to capture 3-D models.

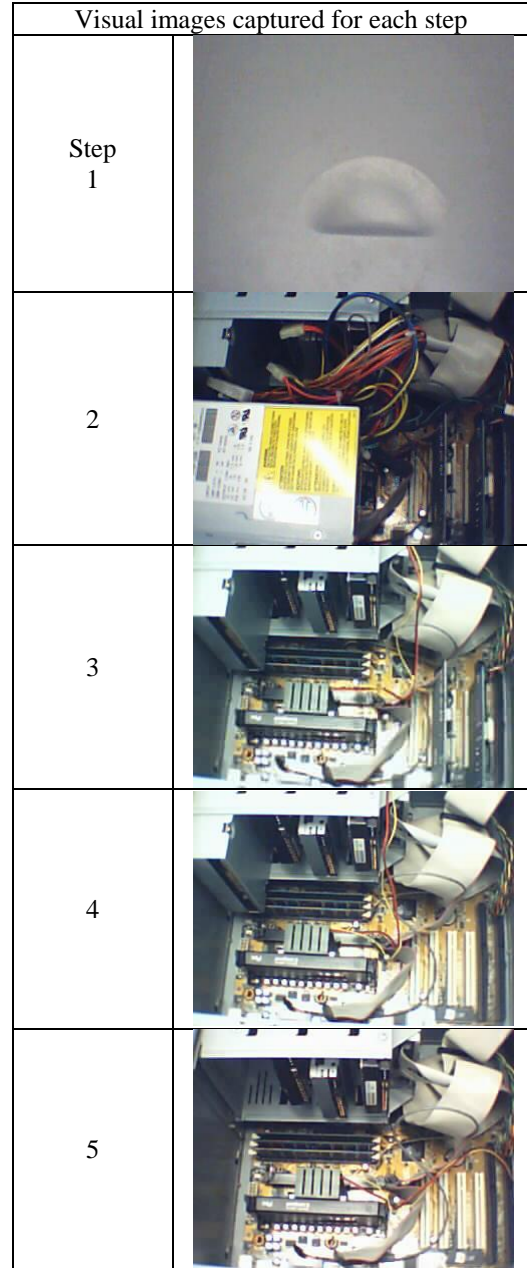


FIGURE 4. THE BILL-OF-MATERIALS OF THE PC WITH CORRESPONDING END-OF-LIFE PROCESSING OPTIONS.

REFERENCES

- [1] Kongar, E. and Gupta, S. M., 2006. "Disassembly Sequencing Using Genetic Algorithm," *International Journal of Advanced Manufacturing Technology*, vol. 30, pp. 497-506.
- [2] ElSayed, A., Kongar, E., and Gupta, S. M., 2010. "A Genetic Algorithm Approach To End-Of-Life Disassembly Sequencing for Robotic Disassembly," in

- Northeast Decision Sciences Institute Conference, NEDSI 2010*, Alexandria, VA.
- [3] Moyer, L. and Gupta, S. M., 1997. "Environmental Concerns and Recycling/ Disassembly Efforts in the Electronics Industry," *Journal of Electronics Manufacturing*, vol. 7, pp. 1-22.
- [4] Ammons, J. C., Realf, M. J., Newton, D., and Cerav, S., 1999. "Determining Reverse Production Systems Infrastructure for Electronic Assemblies," in *Proceedings of the 1999 IEEE International Symposium on Electronics and the Environment*, Danvers, Massachusetts, pp. 257-262.
- [5] Biddle, D., 2000. "End-of-Life Computer and Electronics Recovery Policy Options for the Mid-Atlantic States, 2nd Edition," Center for Solid Waste Research, Mid-Atlantic Consortium of Recycling and Economic Development Officials (MACREDO), Philadelphia, PA, March.
- [6] Lambert, A. J. D. and Gupta, S. M., 2002. "Demand-Driven Disassembly Optimization for Electronic Products," *Journal of Electronics Manufacturing*, vol. 11, pp. 121-135.
- [7] Linton, J., 1999. "Electronic Products at Their End-of-Life: Options and Obstacles," *Journal of Electronics Manufacturing*, vol. 9, pp. 41-51.
- [8] Sodhi, M. S. and Reimer, B., 2001. "Models for Recycling Electronics End-of-Life Products," *OR Spektrum*, vol. 23, pp. 97-15.
- [9] Veerakamolmal, P. and Gupta, S. M., 1999 "Analysis of Design Efficiency for the Disassembly of Modular Electronic Products," *Journal of Electronics Manufacturing*, vol. 9, pp. 79-95.
- [10] Wong, L., 2001. "Barriers to Implementation of Electronic End-of-Life Product Programs in the U.S., Investigating the Environment: Research for Environmental Management," University of California at Berkeley, Environmental Sciences Group Major, Berkeley, CA, *Compilation of Research Reports from the Fall 2000/Spring 2001 Senior Research Seminars*, May.
- [11] Zhou, M., Caudill, R. J. and Sebastian, D., 1999. "Multi-Lifecycle Product Recovery for Electronic Products," *Journal of Electronics Manufacturing*, vol. 9, pp. 1-15.
- [12] Kongar, E. and Gupta, S. M., 2009. "A Multiple Objective Tabu Search Approach for End-of-Life Product Disassembly," *International Journal of Advanced Operations Management (IJAOM)*, vol. 1, pp. 504 - 531.
- [13] Kongar, E. and Gupta, S. M., 2008. "End-of-Life Processing via Data Envelopment Analysis," in *Research Methodology on Data Envelopment Analysis (DEA)*. vol. Chapter 12, J. K. Mantri, Ed., ed Boca Raton, Florida: Universal Publishers, pp. 197-215.
- [14] Kongar, E. and Mueller, D., 2007. "A Comparative Study on Production and End-of-Life Cycles of Anthropogenic Iron Cycles," in *5th International Logistics and Supply Chain Congress 2007*, Istanbul, Turkey.
- [15] Gungor, A. and Gupta, S. M., 1999. "Issues in Environmentally Conscious Manufacturing and Product Recovery: A Survey," *Computers and Industrial Engineering*, vol. 36, pp. 811-853.
- [16] Ilgin, M. A. and Gupta, S. M., 2010. "Environmentally Conscious Manufacturing and Product Recovery (ECMPRO): A Review of the State of the Art," *Journal of Environmental Management*, vol. 91, pp. 563-591.
- [17] Valenzuela-Rendon, M. and Uresti-Charre, E., 1997. "A Non-generational Genetic Algorithm for Multiobjective Optimization," in *Seventh International Conference on Genetic Algorithms*, pp. 658-665.
- [18] Keung, K. W., Ip, W. H. and Lee, T. C., 2001. "The Solution of a Multi-Objective Tool Selection Model Using the GA Approach," *International Journal of Advanced Manufacturing Technology*, vol. 18, pp. 771-777.
- [19] Lazzarini, B. and Marcelloni, F., 2000. "A Genetic Algorithm for Generating Optimal Assembly Plans," *Artificial Intelligence in Engineering*, vol. 14, pp. 319-329.
- [20] Kongar, E. and Gupta, S. M., 2006. "Disassembly To Order System Under Uncertainty," *OMEGA*, vol. 34, pp. 550-561.
- [21] McGovern, S. M. and Gupta, S. M., 2007. "A Balancing Method and Genetic Algorithm for Disassembly Line Balancing," *European Journal of Operational Research*, vol. 179, pp. 692-708.
- [22] Lee, D.-H., Kang, J.-G. and Xirouchakis, P., 2001. "Disassembly Planning and Scheduling: Review and Further Research," *Journal of Engineering Manufacture*, vol. 215, pp. 695-709.
- [23] Lambert, A. J. D. and Gupta, S. M., 2005. *Disassembly Modeling for Assembly, Maintenance, Reuse, and Recycling*. Boca Raton, Florida: CRC Press.
- [24] Sanderson, A. C., Homem de Mello, L. S. and Zhang, H., 1990. "Assembly Sequence Planning," *AI Magazine*, vol. Spring 1990, pp. 62-81.
- [25] Seo, K.-K., Park, J.-H. and Jang, D.-S., 2001. "Optimal Disassembly Sequence Using Genetic Algorithms Considering Economic and Environmental Aspects," *International Journal of Advanced Manufacturing Technology*, vol. 18, pp. 371-380.
- [26] Bierwirth, C., Mattfeld, D. C. and Kopfer, H., 1996. "On Permutation Representations for Scheduling Problems," in *Parallel Problem Solving from Nature -- PPSN IV, Lecture Notes in Computer Science*. vol. 1141, H. M. Voigt, et al., Eds., ed Berlin, Germany: Springer-Verlag, pp. 310-318.

- [27] Bierwirth, C. and Mattfeld, D. C., 1999. "Production Scheduling and Rescheduling with Genetic Algorithms," *Evolutionary Computation*, vol. 7, pp. 1-18.
- [28] Hui, W., Dong, X. and Guanghong, D., 2008. "A Genetic Algorithm For Product Disassembly Sequence Planning," *Neurocomputing*, vol. 71, pp. 2720–2726.
- [29] Shimizu, Y., Tsuji, K. and Nomura, M., 2007. "Optimal Disassembly Sequence Generation Using a Genetic Programming," *International Journal of Production Research*, vol. 45, pp. 4537–4554, 15 September-1 October.
- [30] Torres, F., Gil, P., Puente, S. T., Pomares, J. and Aracil, R., 2004. "Automatic PC Disassembly for Component Recovery," *International Journal of Advanced Manufacturing Technology*, vol. 23, pp. 39-46, Jan.
- [31] Torres, F., Puente, S. T., and Aracil, R., 2003. "Disassembly Planning Based On Precedence Relations Among Assemblies," *International Journal of Advanced Manufacturing Technology*, vol. 21, pp. 317-327.
- [32] Pomares, J., Puente, S. T., Torres, F., Candelas, F. A. and Gil, P., 2004. "Virtual Disassembly Of Products Based On Geometric Models," *Computers in Industry*, vol. 55, pp. 1-14, Sep.
- [33] Gil, P., Pomares, J., Puente, S. V., Diaz, C., Candelas, F., and Torres, F., 2007. "Flexible Multi-Sensorial System For Automatic Disassembly Using Cooperative Robots," *International Journal of Computer Integrated Manufacturing*, vol. 20, pp. 757-772.
- [34] Torres, F., Puente, S., and Diaz, C., 2009. "Automatic Cooperative Disassembly Robotic System: Task Planner To Distribute Tasks Among Robots," *Control Engineering Practice*, vol. 17, pp. 112-121, Jan.
- [35] Milani, A., 2004. "Online genetic algorithms," *International Journal of Information Theories and Applications*, vol. 11, pp. 20-28.
- [36] Brunelli, R., 2009. *Template Matching Techniques in Computer Vision: Theory and Practice*: John Wiley & Sons Inc.
- [37] Gil, P., Torres, F., Ortiz, F. G. and Reinoso, O., 2006. "Detection Of Partial Occlusions Of Assembled Components To Simplify The Disassembly Tasks," *International Journal of Advanced Manufacturing Technology*, vol. 30, pp. 530-539, Sep.