J Intell Robot Syst (2016) 82:69–79
DOI 10.1007/s10846-015-0289-9

Disassembly Sequencing Using Tabu Search

Mohammad Alshibli · Ahmed El Sayed · Elif Kongar · Tarek M. Sobh · Surendra M. Gupta

Received: 30 September 2015 / Accepted: 1 October 2015 / Published online: 14 October 2015

Abstract End-of-life disassembly has developed into a major research area within the sustainability paradigm, resulting in the emergence of several algorithms and structures proposing heuristics techniques such as Genetic Algorithm (GA), Ant Colony Optimization (ACO) and Neural Networks (NN). The performance of the proposed methodologies heavily depends on the accuracy and the flexibility of the algorithms to accommodate several factors such as preserving the precedence relationships during disassembly while obtaining near-optimal and optimal solutions. This paper improves a previously proposed Genetic Algorithm model for disassembly sequencing by utilizing a faster metaheuristic algorithm, Tabu search, to obtain the optimal solution. The objectives of the proposed algorithm are to minimize (1) the traveled distance by the robotic arm, (2) the number of disassembly method changes, and (3) the number of robotic arm travels by combining the identical-material components together and hence eliminating unnecessary disassembly operations. In addition to improving the quality of optimum sequence generation, a comprehensive statistical analysis comparing the previous Genetic Algorithm and the proposed Tabu Search Algorithm is also included

Keywords Disassembly sequence · Electronics disassembly · End-of-life management · Heuristics · Optimization · Robotics applications · Tabu search

M. Alshibli · A. El Sayed · E. Kongar (✉) · T. M. Sobh
University of Bridgeport, Bridgeport, CT, USA
e-mail: kongar@bridgeport.edu

M. Alshibli
e-mail: malshibl@my.bridgeport.edu

A. El Sayed
e-mail: aelsayed@my.bridgeport.edu

T. M. Sobh
e-mail: sobh@bridgeport.edu

S. M. Gupta
Northeastern University, Boston, MA, USA
e-mail: gupta@neu.edu

1 Introduction

Products in today's market can be generally classified into two categories: efficient and responsive. Efficient products are considered to have a stable and constant demand, supply, pricing, and they tend to move slowly through the supply chain. However, the demand, supply, and price for responsive products fluctuate often and these products are characterized by relatively larger profit margins due to their time sensitive nature. This sensitivity requires them to move faster in the forward supply chain to ensure customer satisfaction. With similar logic, the useful lifetime of responsive products tends to be much shorter than their efficient counterparts due to macro environmental changes, viz., globalization and technological advances. Therefore, reverse distribution systems become instrumental in retrieving these products from the market for subsequent re-use, recycling, or proper disposal. Within responsive products, electrical and electronic equipment (EEE) is the largest growing waste stream requiring economically and environmentally solid and efficient reverse logistics and supply chain operations. EEE uses large quantities of natural resources including substantial amounts of precious metals such as gold, silver, and copper during their production. Furthermore, EEE is composed of several components and subassemblies that can be reused even if the whole product might not be technologically valid. Together with the precious material content, the functionality of these partial structures makes recycling and re-use activities economically valid. Re-use, recycling, or proper disposal of any product generally requires disassembly of the end-of-life product.

The efficiency of disassembly operations is a crucial factor in the success of any reverse flow. Because using human labor to disassemble these products adds more cost and time to the overall system, the need for utilizing automated solutions becomes apparent. In addition, the process of disassembly is complicated and carries various risk factors due to the hazardous substances embedded in these products. In some instances, disassembly is also required to replace or fix components that are not accessible by humans, making robotic solutions in these specific situations the only alternative.

The problem of generating an optimal sequence for disassembly operations is rather challenging due to the uncertainty of the process. EEE is subject to various changes in their original bill-of-materials due to technological advances. For instance, a component inside a personal computer may be altered over time due to an upgrade or a change, such as replacing the RAM capacity. Another, perhaps more important challenge that contributes to the complication of disassembly operations is the fact that the majority of products are not designed for disassembly; thus requiring destructive disassembly operations in some instances and prohibiting the reuse of still functioning components.

This paper aims to target the uncertainty and aforementioned challenges by introducing two modules: A sensory system and an online Tabu search algorithm. The sensory system is used to identify the depth of the product with the help of a digital camera capturing product images for processing and detecting the components. The Tabu search algorithm then generates an optimum online real time disassembly sequence using this information, hence overcoming the uncertainty in the product structure.

Figure 1 demonstrates the bill of materials (BOM) of the end-of-life product and depicts the product structure used in this paper. The proposed solution includes a robotic manipulator with a digital camera and utilizes range sensing and component segmentation algorithms (Fig. 2). Table 1 lists all of the

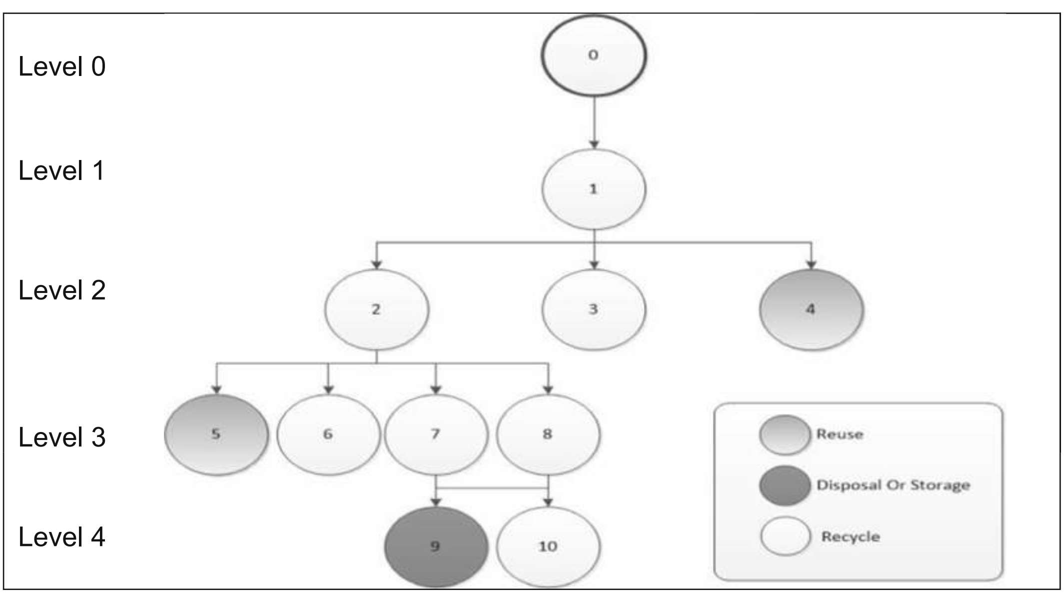


Fig. 1 Bill-Of-Materials (BOM) for the EOL product

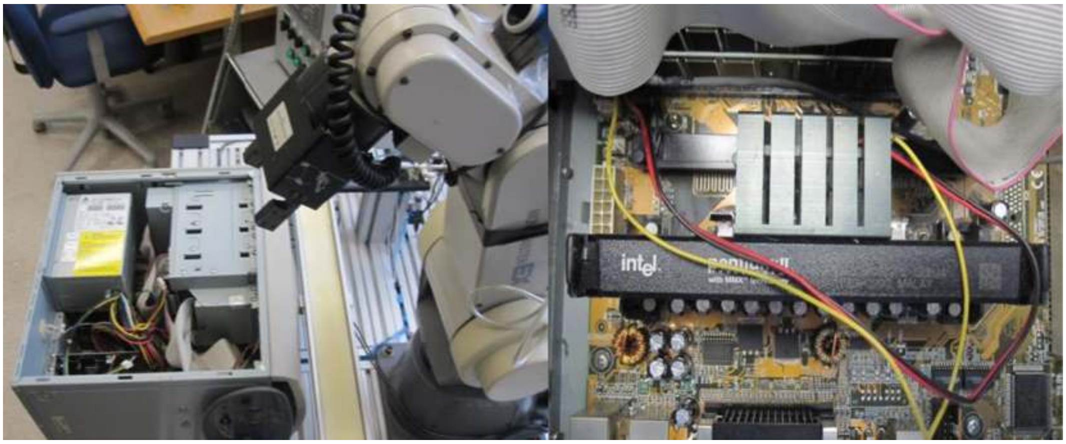


Fig. 2 Screenshot of the robot arm and the processor prior to disassembly

components in the product including their material content and the required disassembly operation (destructive (D) or non-destructive (ND)).

The Tabu Search (TS) algorithm utilized in this paper was first proposed by Fred Glover [1] in 1986 to overcome the Local Optimal Search (LS) problem and enabling global Optima search. Tabu Search generally includes two memories, namely, short and long-term memory. The short-term memory prevents the reversal of the recent moves. The long-term frequency memory reinforces attractive components, forcing the algorithm to move towards more preferable solutions. The algorithm also generates a Tabu list prohibiting returns to previously searched paths. Tabu Seach is an extension of classical LS methods. In fact, basic TS can be seen as simply the combination of LS with short-term memories. The recycle back in the moves is prevented by using the memories (Tabu Lists). Hence, the two first basic elements of any TS heuristic are the definitions of its search space and its neighborhood structure [1].

Table 1 End-of-life product components, material content and required disassembly techniques

Component number	Description	Material	Disassembly method
0	Robot reference point		
1	Side cover	Aluminum (A)	D
2	Power supply	Copper(C)	D
3	Sound card	Plastic (P)	ND
4	Modem card	Plastic (P)	ND
5	CPU	Plastic (P)	ND
6	Hard drive	Aluminum (A)	ND
7	CD drive	Aluminum (A)	ND
8	Zip drive	Aluminum (A)	ND
9	RAM	Plastic (P)	ND
10	Drives slot	Aluminum (A)	D

2 Literature Review and Background

Evolutionary algorithms have been recognized to be well-suited to multi-objective optimization since early in their development [2]. Given that the EOL disassembly embodies several objectives to ensure its efficiency, multi-objective evolutionary algorithms have been extensively used for the EOL disassembly scheduling and/or sequencing problems [3].

Kongar and Gupta [4] considered the case of complete disassembly utilizing both destructive and non-destructive methods. Their paper presented an algorithm for establishing partial and non-destructive disassembly sequences of products, where the recycling and industrial maintenance requires a non-destructive methodology for automatic disassembly. Furthermore, the authors introduced a new representation for the component included in the disassembly based on assemblies of components, not the material. Their method helps in finding the optimum disassembly sequence faster within the process of disassembling products, based on the information from the design process. Therefore, the algorithm could be used in new product design as well as for recycling and product maintenance. The code for the Tabu Search first appeared in Rizk and ElSayed [5].

McGovern and Gupta [6] focused on the disassembly line balancing problem aiming at increasing the process productivity while reducing the number of workstations used. To achieve this, their work utilized a genetic algorithm to obtain the optimal or near-optimal solution for the disassembly sequencing.

ElSayed et al. [7] used a Genetic Algorithm with precedence preservative crossover (PPX) to find the optimum or near-optimum disassembly sequence for complete disassembly The objective of the proposed GA is to minimize the total fitness function 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. Following this, a roulette wheel is employed to select the sequence of parents in the next generation. The objectives include, (1) minimizing the number of workstations and hence, minimizing the total idle time, (2) ensuring workstation idle times are similar, (3) removing hazardous parts early in the disassembly sequence, (4) removing high-demand parts before low-demand parts, and (5) minimizing the number of part removal direction changes required for disassembly. The authors also introduced a new efficiency measurement tool combining Line Efficiency (LE) and Smoothness Index (SI).

Torres et al. [8] proposed a cell with a degree of automation in non-destructive product disassembly. The authors also employed computer vision for object detection in addition to a modeling system for the products. The modeling system provides information regarding the type of products and the main components of the product architecture.

ElSayed et al. [9] proposed an online Genetics Algorithm (GA) that aims at handling uncertainty in the EOL product structure. The algorithm consists of two modules: (i) a sensory-driven visual and range acquisition recovery system, and (ii) an online genetic algorithm (GA) model. The object detection converts objects from 3D to 2D structures via a camera-based algorithm resulting in 21/2 D images. The proposed algorithm finds the optimal disassembly sequence while reducing the time required to disassemble the product.

Xing et al. [10] conducted a survey that reviews the application of soft computing to remanufacturing. The survey aimed at finding answers to various remanufacturing software questions such as the main problems within remanufacturing systems and existing remanufacturing techniques. The survey utilized the data provided by the library of the University of Johannesburg, South Africa. The results were categorized into two basic groups; disassembly and remanufacturing.

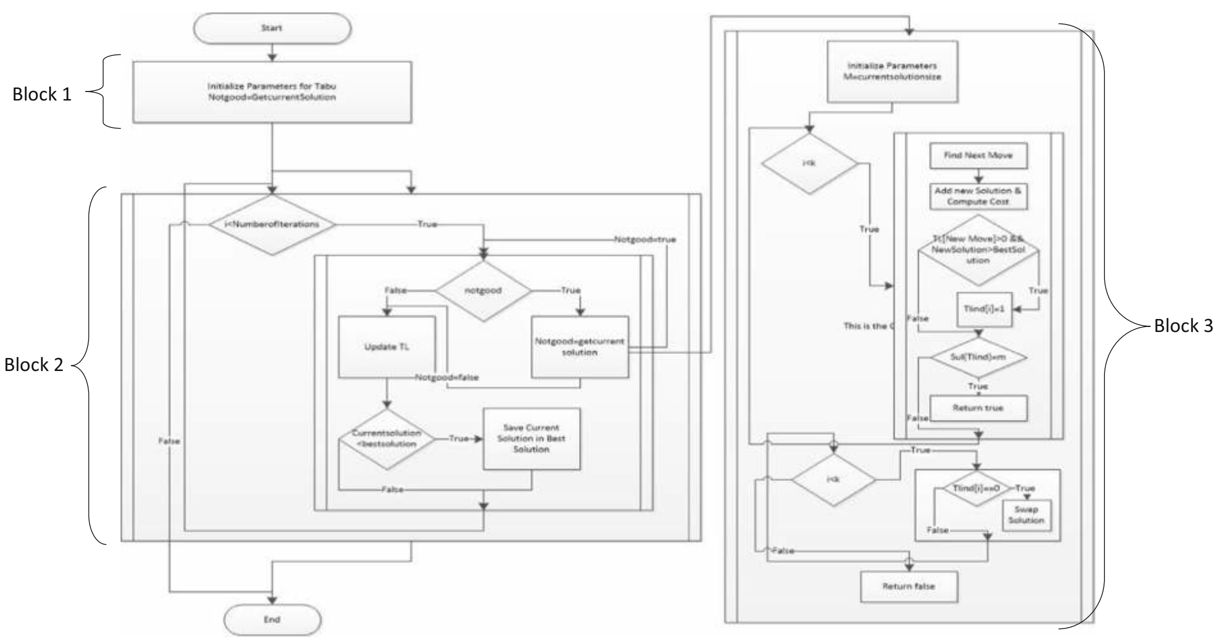


Fig. 3 Tabu Search flowchart

Table 2 Tabu Search algorithm

Step 1	Start with random initial solution
Step 2	Calculate the fitness value for the random generated solution
Step 3	Tabu search will find the next good solution
Step 4	Calculate the fitness for the next solution
Step 5	If next solution has better fitness, set the new solution as the current solution and go to step 3
Step 6	End of iterations, return best selected solution.

Kalayci and Gupta [11] introduced a Tabu Search (TS) algorithm to solve the Disassembly Line Balancing Problem (DLBP) with multiple objectives. The DLBP described in the paper consists of multiple objectives requiring the assignment of disassembly tasks to a set of ordered disassembly workstations while satisfying the disassembly precedence constraints and optimizing the effectiveness of several measures. The authors aimed at reducing the number of disassembly steps required to minimize the total idle time for all workstations. They also assigned the removal of hazardous and high demand components maximum priority.

Torres et al. [12] proposed two types of cooperation among robot arms aiming to manage the task between multiple robots. In the first cooperation, two or more robots cooperate to achieve the same task. In the second type, several tasks are achieved by different robots at the same time. The entire design was built based on a decision tree. The main goal in their follow up work [13] is to retrieve materials from the EOL product via destructive disassembly.

Kuren [14], to find an optimum disassembly path for EOL products, proposed a disassembly cell prototype and presented a case study for mobile phone disassembly. Since a destructive method was used in this paper, the need to used precedence relationships has been eliminated in the proposed solution.

This work is a follow up on the algorithms in Kongar and Gupta [4]. The proposed genetic

Table 3 Pseudocode for Tabu Search

```
BEGIN TS3M
        Set Subdistances to Detected items distances, numberofiterations = NumberOfIterations, CurrentSolution  To InitialSolution, BestSolution To InitialSolution
        CurrentSolution.Cost← ComputeCost
        BestSolution.Cost ← ComputeCost
        InitializeTL
        RunTS
END TS3M
BEGIN COMPUTECOST
        SET ft, f to zero
        IF SolutionArray count  = 0
                SET robot_speed = 7
                SUM(subdistance[:][1])
                IF SolutionArray[0] Not Equal 0
                        Return POSITIVEINFINITY
                ELSE
                        FOR i=0 TO SolutionArray.count
                        Set ct To 0, Var1 To 0, Var2 To 0, Var3 To 0
                        Var1 ← sqr(differencebetween(solutionarray[i-1][2],solutionarray[i-1][2])
                        Var2 ← sqr(differencebetween(solutionarray[i-1][3],solutionarray[i-1][3])
                        Var3 ← sqr(differencebetween(solutionarray[i-1][4],solutionarray[i-1][4])
                        END FOR
                        IF(SolutionArray[i-1][6] =0 and SolutionArray[i][6] =0 and SolutionArray[i][7] =0 and SolutionArray[i][7] =0)
                                f=f-subdistance.solutionarray[i][1]
                        ELSE
                                F=f+ct/robot_spped+abs(solutionarray[i-1][5]- solutionarray]i][5]
                        END IF
                END IF
        END IF
END COMPUTECOST
BEGIN RunTS
        Set notgood to False
        FOR i=0 To numberofiterations
                Notgood← GetCurrentSolution
                While notgood
                        Notgood← GetCurrentSolution
                END WHILE
                UPDATETL
                IF CurrentSolution.cost<BestSolution.cost
                        Swap(CurrentSolution,BestSolution)
                END IF
        END FOR
END RunTS
```

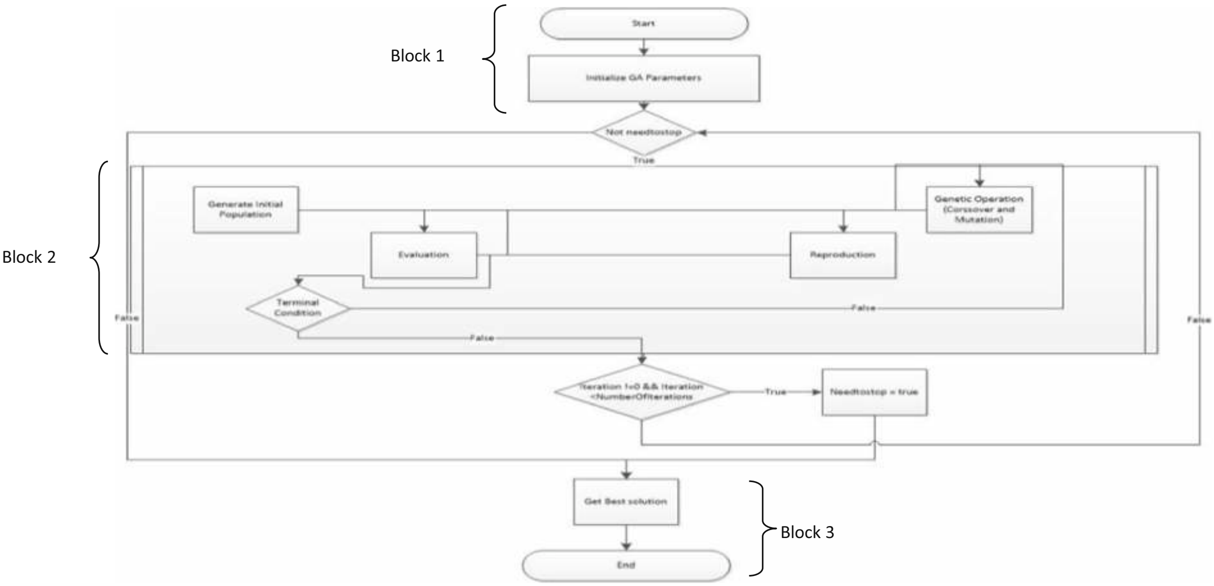


Fig. 4 Genetic Algorithm flowchart

algorithm includes PPX (Precedence Preservation Crossover) to respect the hierarchical structure of the EOL product. The main objective of the algorithm is to minimize the Makespan by minimizing the number of direction changes, disassembly method changes, and combining the identical-material components.

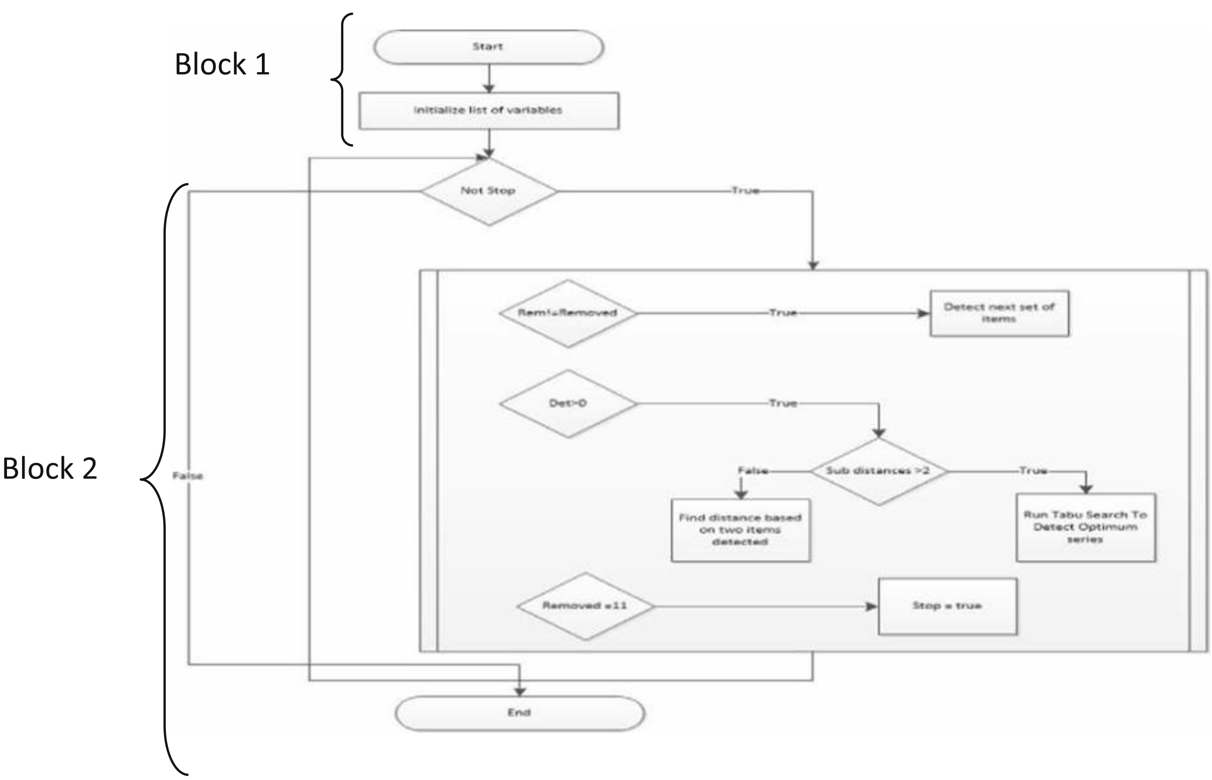


Fig. 5 Overall process for running GA and Tabu

Table 4 Summary statistics for Tabu Search (TS) and Genetic Algorithm (GA) run times in miliseconds

	Tabu Search (TS)	Genetic Algorithm (GA)
Mean	197.65625	402.9844
Standard Error	2.033077929	1.125706
Median	187.5	406.25
Mode	156.25	390.625
Standard Deviation	64.29156917	35.59795
Sample Variance	4133.405867	1267.214
Kurtosis	0.3840795	6.296531
Skewness	0.95576832	1.572811
Range	328.125	328.125
Minimum	78.125	296.875
Maximum	406.25	625
Sum	197656.25	402984.4
Count	1000	1000
Confidence Level (95.0 %)	3.989593024	2.20902

3 Proposed Methodology

The proposed algorithm aims at minimizing the uncertainty in the disassembly process via two techniques: (1) A sensory system, and (2) an online real-time Tabu Search module. The sensory system consists of a robotic manipulator, a digital camera and an image processing algorithm. The camera captures the images of components and/or subassemblies accessible at each level (Fig. 1) and identifies the depth of each available entity. The Tabu Search (TS) algorithm then uses this information to determine the optimal disassembly sequence for the current level. Since the visibility and accessibility of components are altered following each disassembly operation, the Tabu Search algorithm seeks another optimal sequence for the newly generated EOL product structure. The sensory system captures product images after every removal, providing the Tabu Search algorithm with accurate online real-time data. This loop continues until all the components demanded for recycling and reuse are removed. Unwanted components are also subject to disassembly, if and only if their removal would lead to accessibility of desired components; i.e., the components demanded for reuse or recycling. This condition prohibits unnecessary movements and hence reduces the overall Makespan.

The Tabu Search algorithm is motivated by multiple objectives while searching for the best possible sequence within each layer. The algorithm ensures that (1) the distance traveled by the robot arm, (2) the number of disassembly method changes; i.e., from *ND* to *D* or vice versa, and (3) the number of material changes are minimized. Objective (3) is achieved by grouping the components that are made out of identical materials and increases the overall Makespan via a panelizing constant if the following component to be disassembled consists of different material. A literature example is considered to demonstrate the functionality of the proposed algorithm.

The optimal disassembly path search has been conducted via Tabu Search. The following lists the equations applied in the model.

$$t_{ij} = \frac{\sqrt{\left(X_{i(i-1)} - X_{i(i)}\right)^2 + \left(Y_{i(i-1)} - Y_{i(i)}\right)^2 + \left(Z_{i(i-1)} - Z_{i(i)}\right)^2}}{sf} \tag{1}$$

Equation 1 demonstrates the fitness function used to evaluate the generated solution. This function is used every time a new solution is generated calculating the corresponding fitness value. Here, X, Y and Z

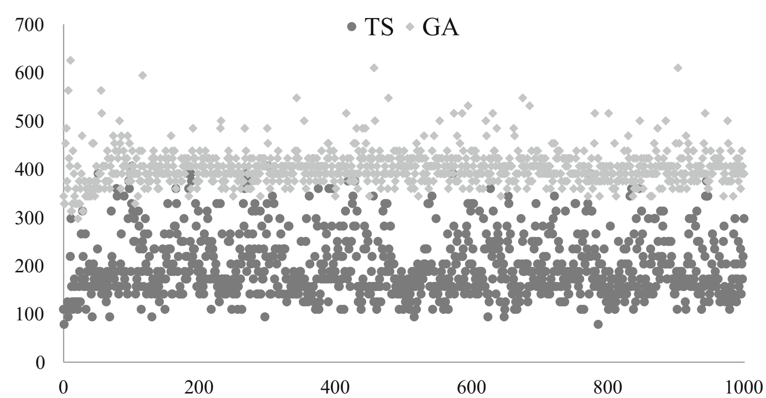


Fig. 6 Scatter plots of Tabu Search (TS) and Genetic Algorithm (GA) run times in miliseconds

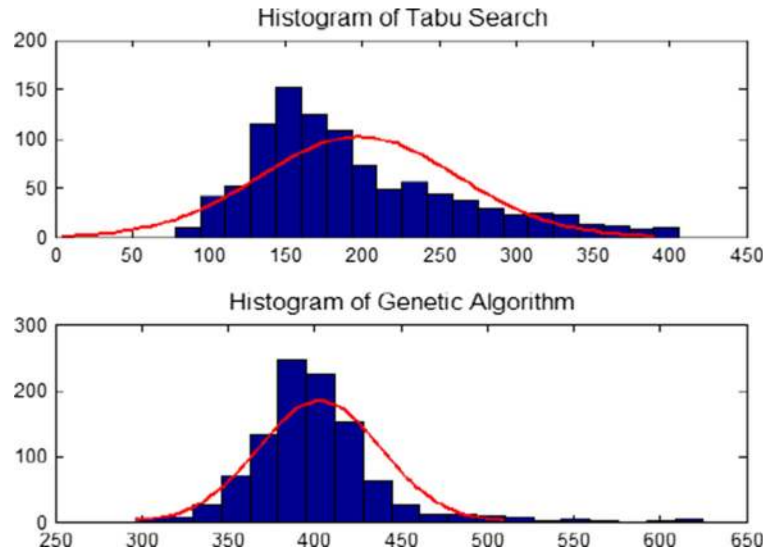


Fig. 7 Histograms of Tabu Search (TS) and Genetic Algorithm (GA) run times in milliseconds

represent the location of the detected item in 3D where X is the x-axis (width), Y is the y-axis (height) and Z is the z-axis (depth). The square root of the distance between object i and i-1 is divided by the robot speed sf (sf = 7 in the provided example).

$$m_{ij} = \begin{cases} 0 \;\; if\ method\ change\ is\ not\ required\ ND\ to\ ND \\ 1 \;\; if\ method\ change\ is\ requried\ ND\ to\ D \end{cases} \quad (2)$$

The second factor that affects the speed of disassembly is the change of disassembly method, viz., from *D* to *ND* or *ND* to *D*. This condition is represented by Eq. 2. The overall objective of the Tabu search is to minimize the fitness function. The cumulative disassembly time after the disassembly is finalized for the sequence j is represented by T_j and is provided in Eq. 3.

$$T_j = T_{j-1} + d_{ij} + t_{ij} + m_{ij} \quad (3)$$

Figure 3 represents the Tabu Search algorithm steps. In Block 1, the parameter initialization is executed to set Tabu parameters, such as short-term memory, to generate the initial solution and to calculate the fitness value of the initial solution. Block 2 is the general loop that runs every iteration during the search. Block 3 explains the internal runs. During the iteration three solutions will be generated and evaluated to find the next best solution. In the case where the current solution is not considered a good one, the same iteration will be executed until a good solution is found. This will prevent the algorithm from falling in local optima and will also serve as the short term memory for the algorithm.

The steps of the Tabu Search algorithm are provided in Table 2 and the pseudo code for the overall search is given in Table 3.

After initializing the algorithm parameters, the ComputeCost function will be executed to calculate the fitness for the first and initial solution, then RunTS will iterate to find the optimal or near optimal solution. In the case where the next best feasible solution

Table 5 Kolmogorov-Smirnov and Shapiro-Wilk tests of normality

	Kolmogorov-Smirnov			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Tabu	.165	1000	.000	.921	1000	.000
Genetic	.174	1000	.000	.879	1000	.000

Table 6 F-Test two-sample for variances results

	Tabu Search (TS)	Genetic Algorithm (GA)
Mean	197.65625	402.984375
Variance	4133.405867	1267.214236
Observations	1000	1000
Df	999	999
F	3.261805108	
P(F <= f one-tail)	4.09549E-74	
F Critical one-tail	1.109746136	

Table 7 ANOVA: Single factor results

ANOVA: Single factor

Source of Variation	SS	Df	MS	*F*	P-value	F crit
Between Groups	21079819	1	21079819	7806.444	0	3.846117028
Within Groups	5395219	1998	2700.31			
Total	26475039	1999				

is found, the new solution will be assigned as the current solution (Best Solution), and the program will continue iterating to obtain a new and better solution. If a better solution does exist, the short term memory provided by the Tabu search algorithm will prevent falling back into local optimal solution.

Figure 4 demonstrates the Genetic Algorithm Flowchart. Block 1 is used to initialize GA parameters such as population, generation size and the number of iterations. Block 2 represents the call of GA functions such as Crossover, Permutation, and Chromosome. Block 3 represents fetching the final result when the run is completed successfully. This result contains the optimal or near optimal solution generated by GA run.

Figure 5 depicts the overall process for the application. Block 1 represents the initialization of all parameters such as object distances, sub-distances, the number of items and the number of detected objects. Block 2 represents the call of Object detection functions, Tabu or GA algorithm to generate the optimal and near optimal solution in addition to the generation of sequence, action and disassembly tool. When this block is executed successfully, the optimal or near optimal solution will be ready, including the disassembly method and the tool needed to disassemble the product.

4 Numerical Example

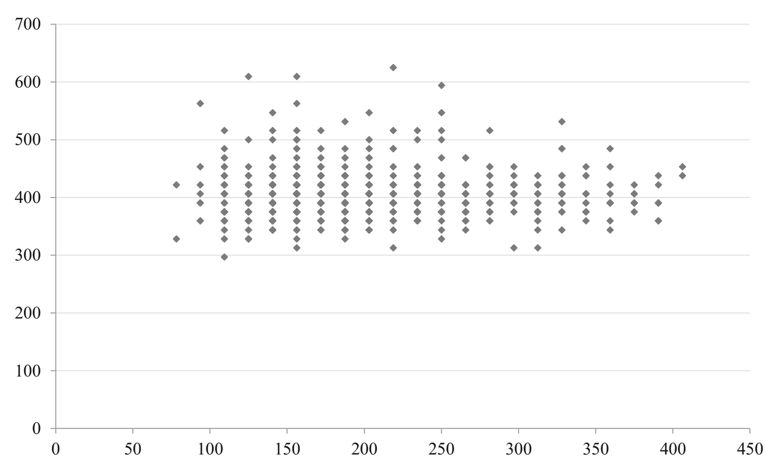
The Tabu Search algorithm is applied to the numerical example provided in Table 1 for the product provided in Fig. 1. 1,000 independent runs are completed to test the Tabu Search and to compare the solutions with the previously published Genetic Algorithm results provided in Kongar and Gupta [4]. The following details the comparison of both algorithms.

In order to validate the reliability of results various statistical analyses have been conducted in SPSS, Excel, Matlab and the Arena Simulation software. The SPSS output of the summary statistics for 1,000 random runs for Genetic Algorithm (GA) and Tabu Search (TS) are provided in Table 4. The median and mode for Tabu Search runs in milliseconds (187.5, 197.65625) are significantly less than the median and mode of the Genetic Algorithm runs (406.25, 402.9844).

Figure 6 depicts the scatter plots of Tabu Search (TS) and Genetic Algorithm (GA) Run Times in Milliseconds. Despite the fact that Genetic Algorithm (GA) runs depict a slower runtime than the Tabu Search, a hypothesis testing has been conducted to prove this suspicion.

The histograms of both runs are provided in Fig. 7. The histograms indicate that Tabu Search

Fig. 8 Scatter plot for Tabu Search (TS) versus Genetic Algorithm



($s^2 = 4133.405867$) runs are more spread compared to Genetic Algorithm ($s^2 = 1267.214$) runs.

Further distribution testing in the Arena simulation software indicated that both data sets are most likely to belong to a Gamma distribution with the parameters 78 + GAMM(35.4, 3.38) for Tabu Search and 78 + GAMM(35.4, 3.38) for the Genetic Algorithm; with test statistics being 0.085 for Kolmogorov Smirnov test and Chi Square test statistics being 559 for both data sets.

Since for a dataset smaller than 2,000 elements the Shapiro-Wilk test is considered more reliable and both Kolmogorov-Smirnoff and Shapiro-Wilk normality tests are conducted; the SPSS results of Kolmogorov-Smirnoff (.165 > .000 for Tabu Search and .174 > .000 for Genetic Algorithm) and Shapiro-Wilk tests (.921 > .000 for Tabu Search and .879 > .000 for Genetic Algorithm) for normality show that both datasets are not from a standard normal distribution (Table 5). The alternative hypothesis is rejected concluding that neither Tabu Search nor the Genetic Algorithm data set comes from a normal distribution.

F-Test Two-Sample for Variances indicates that the variances are not equal to each other (Table 6).

Due to the fact that the data sets are not normally distributed, ANOVA single factor test was also run. The results are provided in Table 7, indicating that the variation between the data sets are significantly different.

A scatter plot for Tabu Search (TS) versus Genetic Algorithm (GA) runs is plotted to illustrate the relationship between the two data sets (Fig. 8).

In order to prove the samples are independent of each other, Pearsons Correlation test has been conducted in SPSS. The test results indicate that the strength of association between the variables is very low (r = 0.011), and that the correlation coefficient is significantly close to zero (P = 0.719 > 0.001). In additon, we can say that 0.0121 % (0.011^2) of the variation in GA run times is explained by TS run times.

5 Conclusions and Future Work

In summation, it can be concluded that the data sets are statistically different from one another with unequal variances and significantly low correlation. Tabu Search runs are statistically lower than Genetic Algorithm runs, hence providing faster solutions to the disassembly sequencing problem.

Future work will include utilization of Active Shape Models (ASMs). ASMs are defined as a statistical model of the object shape which is iteratively deformed to fit an example. ASMs could be generalized for any object by training and identifying the main object curves and points that identify the object.

References

1. Glover, F.: Tabu search-part I. ORSA J. Comput. **1**, 190–206 (1989)
2. Fonseca, C.M., Fleming, P.J.: An overview of evolutionary algorithms in multiobjective optimization. Evol. Comput. **3**, 1–16 (1995)
3. Galantucci, L.M., Percoco, G., Spina, R.: Assembly and disassembly planning by using fuzzy logic & genetic algorithms. Int. J. Adv. Robot. Syst. **1**, 67–74 (2004)
4. Kongar, E., Gupta, S.M.: Disassembly sequencing using genetic algorithm. Int. J. Adv. Manuf. Technol. **30**, 497–506 (2006)
5. Rizk, M., ElSayed, A.: Learning air-to-ground integrated fuzzy guidance system using Tabu search. UbiCC Journal **6**, 843–847 (2011)
6. McGovern, S.M., Gupta, S.M.: A balancing method and genetic algorithm for disassembly line balancing. Eur. J. Oper. Res. **179**, 692–708 (2007)
7. ElSayed, A., Kongar, E., Gupta, S.M.: A genetic algorithm approach to end-of-life disassembly sequencing for robotic disassembly. Presented at the Northeast Decision Sciences Institute Conference, Hilton Alexandria Old Town, Alexandria, VA (2010)
8. Torres, F., Gil, P., Puente, S.T., Pomares, J., Aracil, R.: Automatic PC disassembly for component recovery. Int. J. Adv. Manuf. Technol. **23**, 39–46 (2004)
9. ElSayed, A., Kongar, E., Gupta, S., Sobh, T.: A robotic-driven disassembly sequence generator for end-of-life electronic products. J. Intell. Robot. Syst. **68**, 43–52 (2012)
10. Xing, B., Gao, W.-J., Nelwamondo, F.V., Battle, K., Marwala, T.: Soft computing in product recovery: A survey focusing on remanufacturing system. ICIC Express Letters. **6**, 23–27 (2012)
11. Kalayci, C.B., Gupta, S.M.: Tabu search for disassembly line balancing with multiple objectives. Presented at the 41st International Conference on Computers and Industrial Engineering (CIE41) (2011)
12. Torres, F., Puente, S., Díaz, C.: Automatic cooperative disassembly robotic system: Task planner to distribute tasks among robots. Control. Eng. Pract. **117**, 112–121 (2009)
13. Puente, S., Torres, F., Reinoso, O., Paya, L.: Disassembly planning strategies for automatic material removal. Int. J. Adv. Manuf. Technol. **46**, 339–350 (2010)
14. Kuren, B.-V.: Flexible robotic demanufacturing using real time tool path generation. Robot. Comput. Integr. Manuf. **22**, 17–24 (2006)

Mohammad Alshibli earned his Bachelor of Computer Science from Philadelphia University Jordan in 2002. He received his Master of Science degree in Computer Science in 2011 from The Al-Balq'a Applied University Jordan. In 2012 he joined the doctoral program In Computer Science at University of Bridgeport at Connecticut.

Ahmed El Sayed received his B.Sc. in Electrical Engineering in 2003 from Department of Electrical Engineering, Alexandria University, Egypt, M.Sc. in Engineering Mathematics in 2006 from the Department of Engineering Mathematics and Physics, Alexandria University, Egypt, and M.Sc in Computer Engineering in 2011 from Department of Computer Science and Engineering, University of Bridgeport. Currently he is a PhD candidate in the Department of Computer Science and Engineering, University of Bridgeport.

He is a Teaching and Research Assistant in the Department of Computer Science and Engineering, University of Bridgeport. He worked as a Teaching Assistant in the Department of Engineering Mathematics and Physics, Alexandria University, Egypt. He published papers in the fields of Fuzzy Control systems, Soft Computing methods, Computer Vision. His research interests include AI, Fuzzy Systems, Soft Computing, Machine Learning, Pattern Recognition and Computer Vision.

Elif Kongar is an Associate Professor at the University of Bridgeport. She is also serving as the Director of the Technology Management Ph.D. program while leading the development of blended and distance learning curricula. Dr. Kongar received her BS and MS degrees in industrial engineering from Yildiz Technical University, and PhD degree in industrial engineering from Northeastern University. Before joining the University of Bridgeport, Dr. Kongar was a Visiting Researcher in the Center for Industrial Ecology at Yale University. She also served as the Coordinator and Lecturer of the Logistics certificate program at Yildiz Technical University where she held an Assistant Professor position. Her main area of research is economically and environmentally sustainable waste recovery systems and operations. Dr. Kongar is the author of numerous journal and conference papers, and has presented her work at various national and international conferences. She has chaired several international meetings, conferences, and programs within the areas of supply chain and logistics, product recovery, environmentally conscious manufacturing, and engineering education.

Tarek M. Sobh received the B.Sc. in Engineering degree with honors in Computer Science and Automatic Control from the Faculty of Engineering, Alexandria University, Egypt in 1988, and M.S. and Ph.D. degrees in Computer and Information Science from the School of Engineering, University of Pennsylvania in 1989 and 1991, respectively. He is currently the Senior Vice President for Graduate Studies and Research, Dean of the School of Engineering and Distinguished Professor of Engineering and Computer Science at the University of Bridgeport (UB), Connecticut; the Founding Director of the Interdisciplinary Robotics, Intelligent Sensing, and Control (RISC) laboratory; the Founder of the High-Tech Business Incubator at UB (CTech IncUBator), and a Professor of Computer Engineering, Computer Science, Electrical Engineering and Mechanical Engineering. Professor Sobh's background is in the fields of robotics, automation, sensing, computer vision, computer science and engineering, control theory, manufacturing, AI, and signal processing. His full profile and recent activities can be viewed at http://tareksobh.org.

Surendra M. Gupta is a Professor of Mechanical and Industrial Engineering and Director of the Laboratory for Responsible Manufacturing at Northeastern University in Boston, Massachusetts, USA. He received his BE in Electronics Engineering from Birla Institute of Technology and Science, MBA from Bryant University, and MSIE and Ph.D. in Industrial Engineering from Purdue University. He has authored and co-authored over 350 technical papers published in prestigious journals, books and conference proceedings. Dr. Gupta is a recipient of the Outstanding Research Award and the Outstanding Industrial Engineering Professor Award (in recognition of Teaching Excellence) from Northeastern University. His recent activities can be viewed at <http://www1.coe.neu.edu/~smgupta/>.